

QSG180: Zigbee EmberZNet Quick-Start Guide for SDK v7.0 and Higher

This quick-start guide provides basic information on configuring, building, and installing applications for the EFR32MG family of SoCs using the Zigbee EmberZNet Software Development Kit (SDK) v7.0 and higher with Simplicity Studio 5.3 and higher.

For similar information regarding earlier SDK versions, see <u>QSG106</u>: <u>Zigbee EmberZNet Quick-Start Guide for SDK v6.10 and Earlier</u>.

This guide is designed for developers who are new to Zigbee EmberZNet and the Silicon Labs development hardware. It provides instructions to get started using the example applications provided with the Zigbee EmberZNet SDK.

KEY FEATURES

- Product overview
- Setting up the development environment
- Installing software
- Creating an example application network
- Using the Network Analyzer

1 Product Overview

Before following the procedures in this guide you must have

- Purchased your EFR32MG Mesh Networking Kit (see https://www.silabs.com/wireless/zigbee).
- Downloaded the required software components, as described below. A card included in your development hardware kit contains a link to a Getting Started page, which will direct you to links for the Silicon Labs software products.

1.1 Note on Product History

Zigbee EmberZNet SDK v8 is our latest Zigbee stack introduced along with our newest embedded software platform, the Simplicity Software Development Kit (SDK). It is designed specifically for our Series 2 and Series 3 devices. The Simplicity SDK is a significant advancement in IoT development, providing our customers with a cohesive development environment for wireless innovation. It enables wireless developers to utilize advanced features and capabilities with the most recent Silicon Labs IoT devices. See https://github.com/SiliconLabs/simplicity_sdk for more information.

This document is based on Zigbee EmberZNet SDK v7, which contains a number of changes compared to SDK v6.x. Many of these changes are due to an underlying framework redesign that results in an improved developer experience within Simplicity Studio 5 (SSv5). Projects are now built on a component architecture instead of AppBuilder. Simplicity Studio 5 includes project configuration tools that provide an enhanced level of software component discoverability, configurability, and dependency management. These include a Project Configurator, a Component Editor, and a Zigbee Cluster Configurator. See *AN1301: Transitioning from Zigbee EmberZNet SDK 6.x to SDK 7.x* for more information about the differences between SDK 6.x and SDK 7.x.

SDKs beginning with version 6.8 are only compatible with Simplicity Studio 5 (SSv5). Among many other improvements, SSv5 introduced the *Simplicity Studio 5 User's Guide*, available online at <u>https://docs.silabs.com/</u> and through SSv5's help menu. Standard information, such as how to download SSv5 and the Zigbee EmberZNet SDK, and descriptions of SSv5 features and functions, are provided in that guide, and are not repeated here.

1.2 Software

See the stack release notes for version restrictions and compatibility constraints for the stack and other software. Your Zigbee EmberZNet stack may be delivered as part of one of two SDKs:

- Zigbee EmberZNet v8 and above is included in the Simplicity SDK (SiSDK).
- Zigbee EmberZNet v7 is included in the Gecko SDK (GSDK).

This guide applies to both SDKs but will primarily reference the Gecko SDK. However, all steps are equally relevant for the Simplicity SDK.

To develop Zigbee EmberZNet applications, you will need the following.

- Simplicity Studio 5 (SSv5): Simplicity Studio is the core development environment designed to support the Silicon Labs IoT portfolio
 of system-on-chips (SoCs) and modules. It provides access to target device-specific web and SDK resources; software and hardware
 configuration tools; an integrated development environment (IDE); and advanced, value-add tools for network analysis and codecorrelated energy profiling. When you install Simplicity Studio it will walk you through installing Gecko SDK, the suite of Silicon Labs
 SDKs, including the Zigbee EmberZNet SDK. Alternatively, Gecko SDK, including Zigbee EmberZNet, may be installed manually by
 downloading or cloning the latest from GitHub. See https://github.com/SiliconLabs/gecko_sdk for more information.
- The Zigbee EmberZNet stack, an advanced implementation of a Zigbee stack, installed as part of the Gecko SDK. The stack API is
 documented in an online API reference at https://docs.silabs.com/. The stack is delivered as a collection of libraries that you can link
 to your applications. A description of each library is provided in the development environment. The release notes contain details on
 the folders installed along with their contents.
- Simplicity Commander, installed along with Simplicity Studio. A GUI with limited functionality can be accessed through Simplicity Studio's Tools menu. Most functions are accessible through a CLI invoked by opening a command prompt in the Simplicity Commander directory (\SiliconLabs\SimplicityStudio\developer\adapter_packs\commander). See UG162: Simplicity Commander Reference Guide for more information.
- Compiler toolchain (see the SDK release notes for compatible versions):
 - GCC (The GNU Compiler Collection) is provided with Simplicity Studio. GCC is used in this document. However, you must use IAR to compile projects for a part with less than 512 kB, such as the EFR32xG1

Note: Application images created with GCC are larger than those created with IAR. If you use GCC to compile the example applications in this SDK, you must use a part with at least 512 kB of flash.

• IAR Embedded Workbench for ARM (IAR-EWARM).

Note: Download the compatible version from the Silicon Labs Support Portal, as described in section 1.6 Using IAR as a Compiler. Refer to the "QuickStart Installation Information" section of the IAR installer for additional information about the installation process and how to configure your license. Once IAR-EWARM is installed, the next time Simplicity Studio starts it will automatically detect and configure the IDE to use IAR-EWARM.

While Simplicity Studio and Simplicity Commander can be run on a Mac OS or Linux machine, these instructions assume you are working with a Microsoft Windows-based PC. If you are using a non-Windows system, IAR-EWARM must be run via WINE or some other form of emulator or virtual machine.

1.3 Support

You can access the Silicon Labs support portal at <u>https://www.silabs.com/support</u> through Simplicity Studio. Use the support portal to contact Customer Support for any questions you might have during the development process. Access is through the Welcome view under Learn and Support. Note that you can return to the Welcome view at any time through the Welcome button on the toolbar.

v5.rel.Staging_39 - soc_empty_4/soc_empty_4.slcp - Simplicity Studi	57	- 🗆 ×			
Ele Edit Nongate Search Project Bun Window Help		😰 🛛 🖉 Launcher 🚯 Simplicity IDE			
Skill All Skill All <th< td=""><td>Welcome to Simplicity Studio Everything you need to develop, research, and configure devices for I</td><td>IoT applications.</td></th<>	Welcome to Simplicity Studio Everything you need to develop, research, and configure devices for I	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 All Products Connected Devices Connected Dev	✓ Start			
	Recent Projects				
🗅 My Products 🛛 🗳 🛊 🖹 🖻 🕀 🖓 🗖					
Enter product name	BECENT Projects BT_soc_empty_virtualWSTK_from_last	- Open			
Compared and the second s					
	∧ Learn and Support				
	- Ç:- Tips and Tricks Useful tips and tricks to help youplaceholder content Start	Technical Support Get technical support from Silicon Lab			
	Video Tour Discover the basics of developing for IoT devices and other types of embedded Watch systems	Silicon Labs Community Join the Silicon Lab community to learn, get help, and grow your skills Join			
		SEBM of 1543M			

1.4 Documentation

Documentation is accessed through Simplicity Studio. Simplicity Studio filters documentation based on the connected or selected part. Hardware-specific documentation can be accessed through links on the part OVERVIEW tab.

S K C R R (D-440085390)	EFR32MG12 2400/915 MHz 19	dBm RB, WSTK Mainboard (ID:	000440085390)				
	OVERVIEW EXAMPLE PROJECTS DOCUMENTATI	ON DEMOS COMPATIBLE TOOLS					
			Create New Project				
	General Information	Recommended Quick Start Guides					
	Connected Via: 🌵 J-Link Silicon Labs 🍾 Configure	Connected Via:					
	Debug Mode: Onboard Device (MCU) 🖌 Change 🔯 QSG155: Using the Silicon Labs Dynamic Multiprotocol Demonstrations						
	Adapter FW: 1v4p4b1099 Latest	st D QSG139: Bluetooth® SDK v2.x Quick-Start Guide					
	Preferred SDK:						
ly Products 😂 🕂 🗱 📄 🗃 🗖	Gecko SDK Suite v3.0.0 Manage SDKs 👻						
er product name	Gecko SDK Suite v3.0.0 Manage SDKs v						
y Products Sty Products My Products	Gecko SDK Suite v3.0.0 Manage SDKs - Board	Board	Target Part				
er product name		Board	Target Part				
er product name		Board	Target Part EFR32MG12P433F1024GL125				

SDK documentation and other references are available through the DOCUMENTATION tab. Filter with the Zigbee Technology Type checkbox to see documentation most closely related to the Zigbee EmberZNet SDK.

Simplicity Studio™ v5.master.Staging_2269 - Z3Light/Z3Light.slcp - Simplicity Studio™		- O X
Eile Edit Navigate Search Project Bun Window Help		
↑ Welcome ③ Recent III Tools 📩 Install 🌣 Preferences		😰 🚺 Simplicity IDE 🦧 Launcher
Bit Debug Adapters Image: Constraint of the state of	EFR32MG12 2.4 GHz 1	9 dBm RB, WSTK Mainboard (ID: 000440050128)
	Read documentation written for your devic	e
	Application Notes (42)	71 resources found
	Data Sheets (0) Errata (0) Quick Start Guides (2)	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.
	Reference Manuals (4) Release Notes (3) Schematic and Layout Files (0) User's Guides (20)	UG103.01: Wireless Networking Fundamentals Introduces some fundamental concepts of wireless networking. These concepts are referred to in other Fundamentals documents. If you are new to wireless networking, read this document first.
My Products My Products Fatter product name My Products	White Papers (0) Technology Type Clear Filter	UG103.02: Zigbee Fundamentals Describes the key features and characteristics of a Zigbee solution. It also includes a section on Zigbee 3.0.
	Bluetooth (28) Bluetooth Mesh (22) Bootloader (1) Operating Systems (5) Proprietary (26)	UG104: Testing and Debugging Applications for the Silicon Labs EFR32MG Platform Describes strategies for testing and debugging applications, including: hardware and application considerations, initial development testing, and lab testing. For additional information about later stages of programming and testing see application notes AN700.1: Manufacturing Test Guidelines and AN718: Manufacturing Test Overview.
	 Thread (9) Wi-SUN (7) Z-Wave (0) ✓ Zlgbee (71) 	UG491: Zigbee Application Framework Developer's Guide for SDK 7.0 and Higher The Zigbee Application Framework is a body of embedded C code that can be configured by project configuration tools to implement any Zigbee Cluster Library (ZCL) application. This guide covers the structure and usage of the Zigbee Application Framework in SDK 7.0 and higher.
Log In 🔻	¥	© 2022 Silicon Labs

1.5 Gecko Platform

The Gecko Platform is a set of drivers and other lower-layer features that interact directly with Silicon Labs chips and modules. Gecko Platform components include EMLIB, EMDRV, RAIL Library, NVM3, and mbed TLS. Application developers using Zigbee EmberZNet components, the PIN tool, or APIs may not need to interact directly with the Gecko Platform, as the code does that for you. For more information about Gecko Platform, see the Gecko Platform release notes (on the DOCUMENTATION tab, filter by Resource Type: Release Notes).

1.6 Using IAR as a Compiler

If you plan to use IAR as your compiler (required for dynamic multiprotocol and Micrium OS examples), check the compatible IAR version in the SDK release notes. To install IAR-EWARM:

- 1. Go to the Customer Support portal as described in section 1.3 Support.
- 2. If you are not already signed in, sign in.
- 3. Click the Software Releases tab. In the View list select Latest EmberZNet Software. Click Go. In the results is a link to the appropriate IAR-EWARM version.
- 4. Download the IAR package. This is a large package download time depends on connection speed but can take 1 hour or more.
- 5. Install IAR.
- 6. In the IAR License Wizard, click Register with IAR Systems to get an evaluation license.
- 7. Complete the registration and IAR will provide an evaluation license.

2 Set Up Your Development Environment

2.1 Create a Support Account

Before you install Simplicity Studio, you need to create an account on the support portal. Be sure to record your account username and password as you will use it to log in to Simplicity Studio.

Properties in your Salesforce account determine what update notifications you will receive. To review or change your subscriptions, log in to the portal, click HOME to go to the portal home page and then click the Manage Notifications tile. Make sure that Software/Security Advisory Notices & Product Change Notices (PCNs) is checked, and that you are subscribed at minimum for your platform and protocol. Click **Save** to save any changes.

2.2 Connect the Mainboard

Connect your mainboard, with radio board mounted, to your PC using a USB cable.

Note: For best performance in Simplicity Studio, be sure that the power switch is in the Advanced Energy Monitoring or "AEM" position as shown in the following figure.

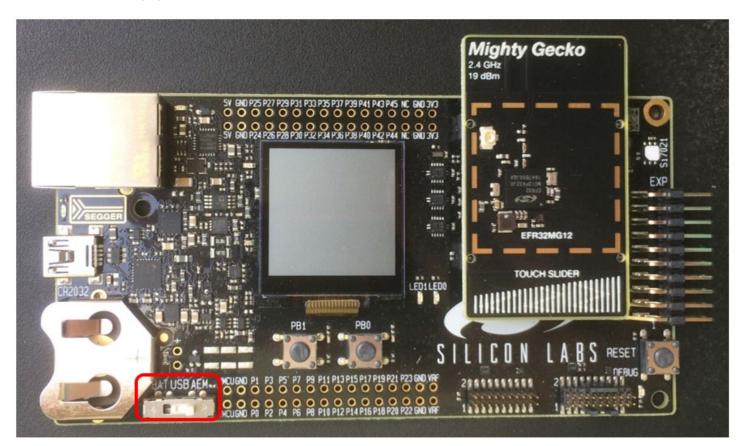


Figure 1. EFR32MG12 on a WSTK Mainboard

2.3 Install Simplicity Studio 5 (SSv5) and the Gecko SDK

General information on using SSv5 and the Gecko SDK installation can be found in the online Simplicity Studio 5 User's Guide.

3 About Demos and Examples

Because starting application development from scratch is difficult, the Zigbee EmberZNet SDK comes with a number of built-in demos and software examples covering the most frequent use cases.

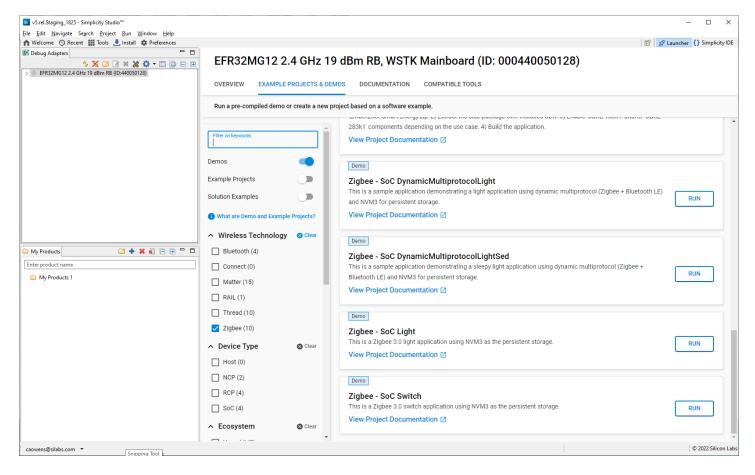
Demos are pre-built application images that you can run immediately. Software examples can be modified before building the application image. The software examples with the same names as the demos provide the demo functionality.

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, be sure to click on the part you are working with to see only those items applicable to that part.

Silicon Labs recommends that you start your own development with a use case-based example and modify it according to your needs. If none of the use case-based examples meet your needs, you can start with the **Zigbee – SoC ZigbeeMinimal** example. The examples provide default configurations needed by the stack and a basic application structure that you can build upon.

3.1 Demos

Demos are prebuilt application examples that can be directly downloaded to your device. These demos are compatible with selected devices, as noted in the demo description. You will not see the demos in Simplicity Studio unless you have one of these devices selected. If you have more than one device connected, make sure that one of these devices is selected in the Debug Adapters view. To download and run a demo on your device, select your device and click the Example Projects & Demos tab in the Launcher perspective. Turn off the Example Projects switch to see only demos, and then click **RUN** next to the demo to load. The solution examples switch is provided for future use.



The Zigbee EmberZNet SDK demos are:

Simple Zigbee 3.0 Network

Zigbee SoC SwitchWithVoice: A Zigbee 3.0 Switch application with extended voice recognition functionality.

Compatible parts: brd2601a

Zigbee – SoC Light: Zigbee 3.0 Light application. **Zigbee – SoC Switch:** Zigbee 3.0 Switch application.

Compatible parts, brd4161a, brd4162a, brd4180a

For the Light and Switch demos, press Button0 on the Switch device to initiate the network. The light device should bind to the network automatically. Once the switch has finished finding and binding, you can use Button0 as an On/Off toggle.

This may not be successful, because some timeouts are included in the button functionality. If so, use the Command Line Interface to set up the demo network.

Once the Z3Light app starts running, it makes many attempts to join a network, at the end of which it sets up its own distributed network. Once the Info command indicates that it is in a network (that is, has a PAN ID and Node ID), use the CLI command:

plugin network-creator-security open-network

You should get the response:

NWK Creator Security: Open network: 0x00 pJoin for 254 sec: 0x00 NWK Creator Security: Open network: 0x00

Now the device is ready for joining. In the Z3Switch App, instead of using the button, first make sure it has not joined any networks by issuing:

network leave

Then you can enter the command:

plugin network-steering start 0

This will kick off the joining process.

Dynamic Multiprotocol

Zigbee – SoC DynamicMultiprotocolLight and **Zigbee – SoC DynamicMultiprotocolLightSed**: Companion applications demonstrating Zigbee/Bluetooth LE dynamic multiprotocol functionality.

Compatible parts: brd4161a, brd4162a

Multiprotocol Radio Co-Processor

Multiprotocol (OpenThread+Zigbee) - RCP (SPI) and Multiprotocol (OpenThread+Zigbee) - RCP (UART): These applications support running OpenThread and Zigbee stacks simultaneously on a host processor. The host applications connect to the CPC daemon, which in turn connects to the EFR via a SPI or UART link, respectively.

Compatible parts: brd4151a, brd4161a, brd4166a, brd4168a, brd4180a, brd4304a

Multiprotocol (OpenThread+Zigbee+BLE) - RCP (SPI) and Multiprotocol (OpenThread+Zigbee+BLE) - RCP (UART): These applications support running OpenThread, Zigbee, and Bluetooth stacks simultaneously on a host processor. The host applications connect to the CPC daemon, which in turn connects to the EFR via a SPI or UART link, respectively.

Compatible parts: brd4161a, brd4166a, brd4168a, brd4180a, brd4304a

NCP Applications

Zigbee - NCP ncp-uart-hw: Network coprocessor (NCP) application that supports communication with a host application over a UART interface with hardware flow control.

Compatible parts: brd4158a, brd4161a, brd4162a, brd4163a, brd4164a, brd4169b, brd4170a, brd4180a

Zigbee - NCP ncp-spi: Network coprocessor (NCP) application that supports communication with a host application over an SPI interface.

Compatible parts: brd4158a, brd4161a, brd4162a, brd4163a, brd4164a, brd4169b, brd4170a, brd4180a

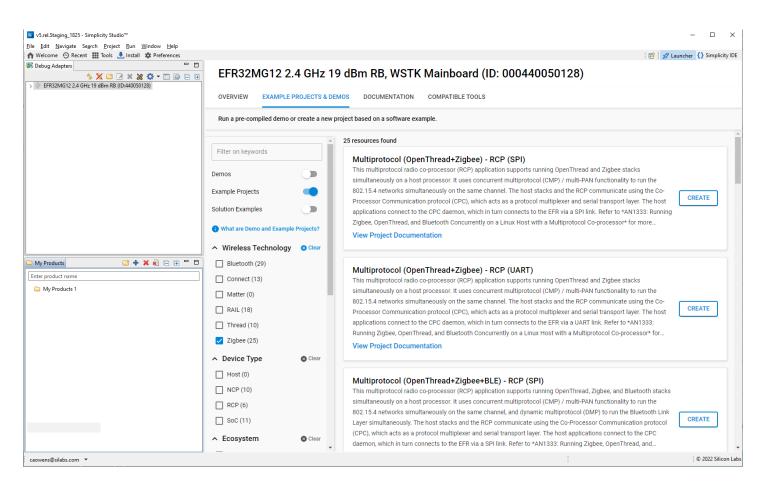
3.2 Software Examples

Note: Examples provided for the EFR32xG12 and newer parts include Silicon Labs Gecko Bootloader examples. Examples are provided for all compatible Simplicity Studio SDKs. When configuring security for a Gecko Bootloader example, you must use Simplicity Commander, not the Simplicity Studio IDE interface. For more information on using the Gecko Bootloader see UG489: Silicon Labs Gecko Bootloader User's Guide for GSDK 4.0 and Higher.

Some Zigbee EmberZNet software examples are specifically for SoC applications. If you are not familiar with the differences between System-on-Chip (SoC) and Network Coprocessor (NCP)/Radio Coprocessor (RCP) application models, see UG103.03: Application Development Fundamentals: Design Choices. For more information on Green Power Devices see UG392: Using Silicon Labs Green Power with Zigbee EmberZNet.

In the Debug Adapters view, select the device. The examples list is filtered by the selected device.

Click the EXAMPLE PROJECTS & DEMOS tab in the Launcher perspective. Under Technology Type, select Zigbee to see only Zigbee examples.



3.2.1 Zigbee 3.0 Network

The Z3Light and Z3Switch applications were developed for specific development kit hardware. Use on other devices may require some GPIO remapping using the PIN tool. See the online Simplicity Studio v5 User's Guide for more information about using the PIN tool.

Zigbee – SoC Light: Zigbee light application. Acting as a router it can form a distributed network. Acting as a touchlink target it can touchlink with the **Zigbee – SoC Switch**, which is acting as a touchlink initiator.

Zigbee – SoC Switch: Zigbee switch application, acting as an end device, can join the network.

Press Button0 on the Switch device to initiate the network. The light device should bind to the network automatically. Once the switch has finished finding and binding, you can use Button0 as an On/Off toggle.

This may not be successful, because some timeouts are included in the button functionality. If so, use the Command Line Interface to set up the demo network.

Once the Z3Light app starts running, it makes many attempts to join a network, at the end of which it sets up its own distributed network. Once the Info command indicates that it is in a network (that is, has a PAN ID and Node ID), use the CLI command:

plugin network-creator-security open-network

You should get the response:

NWK Creator Security: Open network: 0x00 pJoin for 254 sec: 0x00 NWK Creator Security: Open network: 0x00

Now the device is ready for joining. In the Z3Switch App, instead of using the button, first make sure it has not joined any networks by issuing:

network leave

Then you can enter the command:

plug network-steering start 0

This will kick off the joining process.

3.2.2 Dynamic Multiprotocol

Zigbee - SoC DynamicMultiprotocolLight and **Zigbee - SoC DynamicMultiprotocolLightMinimal:** Sample application with Zigbee 3.0 (Coordinator/Router) + Bluetooth LE. The Minimal application does not use buttons, LEDs, or the LCD.

Zigbee - SoC DynamicMultiprotocolLightSed and **Zigbee - SoC DynamicMultiprotocolLightSedMinimal:** Sample application with Zigbee 3.0 (Sleepy End Device) + Bluetooth LE. The Minimal application does not use buttons, LEDs, or the LCD.

See AN1322: Dynamic Multiprotocol Development with Bluetooth and Zigbee in SDK 7.0 and Higher for more information on using these examples.

3.2.3 Green Power

Zigbee - SoC GPD Sensor: Demonstrates a Green Power occupancy sensor device with attributes pre-configured in the component Green Power Device Application Support.

Zigbee - SoC GPD Switch: Demonstrates a Green Power switch device with attributes pre-configured in the component Green Power Device Application Support.

Zigbee - SoC LightGPCombo: Z3 Light GP Combo application demonstrates functionality of a Green Power combo device with proxy and sink instances.

3.2.4 NCP

Zigbee - NCP mp-ncp-spi: The Multi-Pan NCP SPI application supports forming two personal area networks on same channel on single radio.

Zigbee - NCP mp-ncp-uart-hw: The Multi-PAN NCP UART application supports forming two personal area networks on same channel on single radio.

Zigbee - NCP ncp-uart-hw: Network coprocessor application that communicates with a host application over a UART interface with hardware flow control.

Zigbee - NCP ncp-spi: Network coprocessor application that communicates with a host application over a SPI interface.

Zigbee - NCP ncp-uart-hw-gp-multi-rail: Network coprocessor application extends the ncp-uart-hw application with a multi-RAIL library, and a multi-RAIL demo component enabled.

Zigbee - NCP xncp-led-ncp: Network processor application that communicates with a UNIX HOST using custom EZSP commands.

3.2.5 RCP

For more information about these examples, see AN1333: Running Zigbee, OpenThread, and Bluetooth Concurrently on a Linux Host with a Multiprotocol Co-processor

Multiprotocol (OpenThread+Zigbee) - RCP (SPI) and **Multiprotocol (OpenThread+Zigbee) - RCP (UART)**: These applications support running OpenThread and Zigbee stacks simultaneously on a host processor. The host applications connect to the CPC daemon, which in turn connects to the EFR via a SPI or UART link, respectively.

Multiprotocol (OpenThread+Zigbee+BLE) - RCP (SPI) and **Multiprotocol (OpenThread+Zigbee+BLE) - RCP (UART)**: These applications support running OpenThread, Zigbee, and Bluetooth stacks simultaneously on a host processor. The host applications connect to the CPC daemon, which in turn connects to the EFR via a SPI or UART link, respectively.

Zigbee - NCP + OpenThread - RCP zigbee_ncp-ot_rcp-spi and **Zigbee – NCP + OpenThread – RCP zigbee_ncp-ot_rcp-uart**: These concurrent multiprotocol (CMP) application run the Zigbee NCP simultaneously with the OpenThread RCP. The host applications connect to the CPC daemon, which in turn connects to the EFR via a SPI or UART link, respectively.

3.2.6 Minimal Configuration

Zigbee - SoC ZigbeeMinimal: This is a Zigbee minimal network-layer application suitable as a starting point for new application development.

3.2.7 Testing

Zigbee - SoC StandardizedRfTesting: This is a pre-standardization implementation of Zigbee's RF testing standard. It utilizes the TIS (Total Isotropic Sensitivity)/ TRP (Total Radiated Power) testing interfaces and is optional for Zigbee certifications.

4 Working with Examples

In these instructions you will work with two example applications, Zigbee – SoC Light Application and Zigbee – SoC Switch Application. These steps illustrate both how to compile and load example applications and use the Simplicity Studio console interface to work with them, but also how to configure them using three configuration tools:

- · Simplicity Studio's Project Configurator allows you to add and remove functions from the applications in the form of components
- The Component Editor allows you to change the parameters of components installed in the application
- The Zigbee Cluster Configurator allows you to modify Zigbee clusters, attributes, and commands.

The following example procedures are provided. Each procedure builds on the next, so it is recommended to work through them in order.

- Create and Load a Zigbee SoC Light Application: Provides a detailed walk-through of project creation.
- Create and Load a Zigbee SoC Switch Application: Provides some shortcuts to project creation.
- Create a Distributed Network and Issue Commands: Shows how to use the Console interface, create a network, and turn a light on and off.
- Change to a Centralized Network: Describes how to use the Project Configurator and Component Editor to change an application.
- About the Zigbee Cluster Configurator: Introduces additional configuration options provided through the Zigbee Cluster Configurator.
- About Bootloaders provides instructions on uploading a bootloader image. This should not be necessary unless you erase the device.

Section 5 Using the Network Analyzer describes how to use Network Analyzer to observe traffic across the network.

These procedures are illustrated for a WSTK with an EFR32MG. **Note:** Your SDK version may be later than the version shown in the procedure illustrations.

4.1 Create and Load a Zigbee – SoC Light Application

SSv5 offers a variety of ways to begin a project using an example application. The online Simplicity Studio 5 User's Guide, available both through <u>https://docs.silabs.com/</u> and the SSv5 help menu, describes them all. This procedure uses the **File > New > Silicon Labs Project Wizard** method, because it takes you through all three of the Project Creation Dialogs.

 Open SSv5's File menu and select New > Silicon Labs Project Wizard. The Target, SDK, and Toolchain Selection dialog opens. If you want to change the toolchain from the default GCC to IAR, do so here. Click **NEXT**.

Note: If you have both IAR and GCC installed, GCC is the default.

 New Project Wizard 			- 0	×
Target, SDK, and Toolchain Sele	ction			
Select the target board, device, SDK,	and IDE/toolchain to use for the project.			
Target, SDK	Examples		Configura	tion
V High, Ohr	Examples		oomgara	
Target Boards:				
Search or Select				٣
EFR32MG12 2400/915 MHz 19 dBm E	Dual Band Radio Board (BRD4164A Rev A02) 🐼			_
Wireless Starter Kit Mainboard (BRD4)	001A Rev A01) 🔕			
Target Device:				
Search or Select				•
EFR32MG12P433F1024GL125				
SDK:				
Select SDK Gecko SDK Suite: Flex 3.0.0.0, Micrium	n OS Kernel, Platform 3.0.0.0			٣
		🗘 M	anage SD	Ks
IDE / Toolchain: Select IDE / Toolchain				_
Simplicity IDE / GNU ARM v7.2.1				٣
Simplicity IDE / GNU ARM v7.2.1				
Simplicity IDE / IAR ARM v8.30.1.114				
IAR Embedded Workbench				
Makefile IDE				
CANCEL		BACK	FINI	ISH

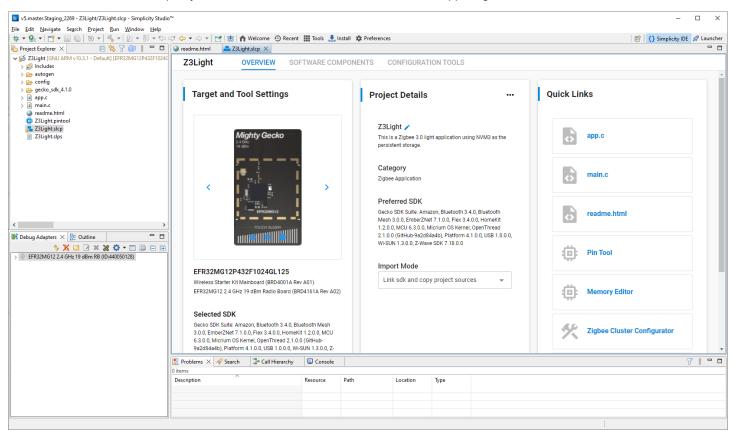
2. The Example Project Selection dialog opens. Use the Technology Type and Keyword filters to search for a specific example, in this case **Zigbee – SoC Light.** Select it and click **NEXT**.

Target, SDK		Config	uratio
		25 resources found	
Filter on keywords Example Projects Solution Examples	•	Zigbee - SoC ZigbeeMinimal This is a Zigbee minimal SoC network-layer application suitable as a starting point for new application development.	
Wireless Technology	8 Clear Filter	Zigbee - SoC Switch This is a Zigbee 3.0 switch application using NVM3 as the persistent storage.	
Connect (13) RAIL (18) Thread (10)		Zigbee - SoC StandardizedRfTesting This is a pre-standardization implementation of Zigbee's RF testing standard. It utilizes the TIS (Total Isotropic Sensitivity)/ TRP (Total	
Zigbee (25) Device Type Host (0) NCP (10)	🛿 Clear Filter	Zigbee - SoC LightGPCombo This is a Zigbee 3.0 light application with Green Power endpoint, Green Power Proxy and Sink functionality.	
RCP (6) SoC (11) Ecosystem	🔇 Clear Filter	Zigbee - SoC Light This is a Zigbee 3.0 light application using NVM3 as the persistent storage.	
Homekit (0)		Zigbee - SoC GPD Switch This is a Green Power On/Off Switch Device that pairs with a GP Combo or	

3. The Project Configuration dialog opens. Here you can rename your project, change the default project file location, and determine if you will link to or copy project files. Note that if you change any linked resource, it is changed for any other project that references it. Click **FINISH.**

lew Project Wizard			
roject Config			
elect the proje	ct name and location.		
Target, SDK	Examples	– 🕜 Configura	itio
Project name:	Z3Light		
Vse defa	ult location		
Location: C:\	Jsers\CAOWENS\SimplicityStudio\v5.master.Staging_1659\Z3Light	BROWSE	
With project file	S:		
C Link to s	urces		
Link sdk	and copy project sources		
-			
O Copy cor	tents		

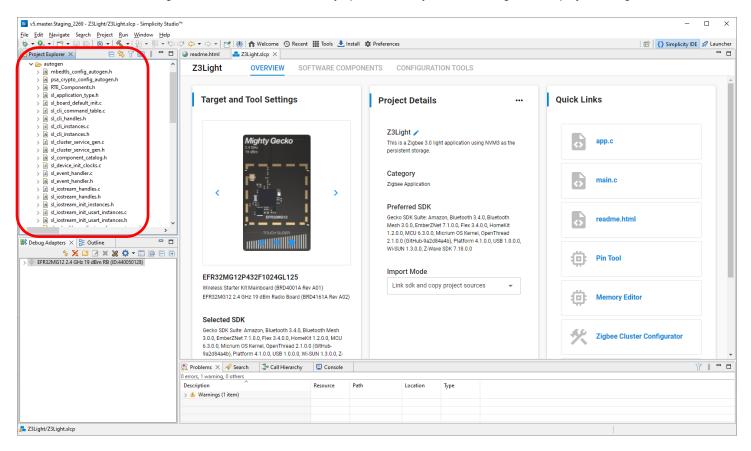
The Simplicity IDE perspective opens with the new project in Project Configurator view (abbreviated name Z3Light). The readme tab contains information about the project. See the online Simplicity Studio 5 User's Guide for details about the functionality available through the Simplicity IDE perspective and Project Configurator.



Note: You now have a Simplicity IDE button next to the Launcher button in the upper right.

Simplicity Studio automatically generates the project files needed to build the application image. If you are accustomed to using Simplicity Studio 6.x, note that this means there is no **Generate** button on the Project Configurator interface. The Force Generation button on the Project Details card is not a replacement, but rather to be used under conditions in which auto-generation fails to work.

Files are created in the autogen folder, and are automatically updated when you make changes to the project configuration.



4. Example applications are pre-configured to support the example functionality. To see the configuration, click the SOFTWARE COMPONENTS tab and search for the component of interest, for example 'device configuration'. Installed components are indicated by a check on the left. Configurable components are indicated by a gear icon on the right. Select the component to see more information about it.

Note: All EFR32 parts have a unique RSSI offset. In addition, board, antenna and enclosure design can also impact RSSI. When creating a new project, install the **RAIL Utility, RSSI** component. This feature includes the default RSSI Offset Silicon Labs has measured for each part. This offset can be modified, if necessary, after RF testing of your complete product.

📀 readme.html 🛛 🚨 Z3Light.slcp 🗙			- 0
Z3Light OVERVIEW SOFTWARE COMPONE	NTS	CONFIGURATION TOOLS	
▼ Filter components by 🏚 Configurable 🔲 🕥 Installed [_ _	Installed by you 🔲 🗈 SDK Extensions 🔲 🔍 Quality 👻 Search keywords, component's name Zigbee 🗞	•
Stack Token Interface	-	Zigbee Device Config	
Strong Random Api (PSA)			
⊘ Strong Random Api (Radio)		Description	
Throughput		Description This component is used for Zigbee device configuration.	
Watchdog periodic refresh sample implementation	n 🗢	Quality	
Xmodem Sender		PRODUCTION	
⊘ Zigbee Cluster Service			1
⊘ Zigbee Core CLI	۰	Dependencies ~	
Ø Zigbee Device Config	•	zigbee_device_config requires 0 components No Dependencies	
⊘ Zigbee Signature Decode		Dependents	
⊘ Zigbee ZCL CLI		1 components require zigbee_device_config	
▼ Zigbee 3.0		▶ Zigbee	
BLE Event handler sample implementation			
Find and Bind Initiator	٥		
⊘ Find and Bind Target	٠	× Uninstall	
HA Device Trust Center Link Key Update			

5. Click **Configure** to open the Component Editor on a new tab. The configurable parameters are shown. In this case, it indicates that the Z3Light application is configured as a router. Close the Component Editor.

🕝 readme.html 🏥 Z3Light.slcp 🚺 Zigbee Device Config 🗙				- 0
Zigbee Device Config	[Pin Tool	View Source	×
ZigBee device configuration				
Primary Network Device Type	Primary Network Security Type			
Router	ZigBee 3.0 Security 🔹			
Enable Secondary Network				

6. Once you are ready to build the application, click the **Build** (hammer) control in the top tool bar. During the build, progress is reported both in a window, which can be run in the background, and also in the lower right. The process may take over a minute.

Note: If the Build control isn't active, click the project directory or a file in it.

File Edit Navigate Search Project Run Window Help	a b leater least			
🗂 ▼ 🗐 🐚 🥸 ▼ 🔦 ▼ 🎋 ▼ 😪 ▼ 🖉 ▼ 🖗 ▼ 🤃 ♥ Project Explorer 🛛 🗧 🧐 🌾		n Welcome 🕑 Recent 🏢 Tools 👱 Install	Preferences	😰 🕼 🕼 Launcher 🚺 Simplicity IDE
v 🎘 autogen				
> h mbedtls_config_autogen.h	23Light	OVERVIEW SOFTWARE COM	PONENTS CONFIGURATION TOOLS	
b psa_crypto_config_autogen.h				
> h RTE_Components.h				Search keywords, component's name
> is board default init.c	Tilter : Confi	gurable Components 📃 🛛 Installed	Components Components Installed by You	device con 😒
> c sl_cli_command_table.c				
> h sl_cli_handles.h				
> c sl_cli_instances.c	- Zighee			
> h sl_cli_instances.h	▼ Zigbee		Device Configuration	📩 Configure
> h sl_component_catalog.h				
> 🖻 sl_device_init_clocks.c	 Stack 			
> c sl_event_handler.c	Dec Ore			
> in sl_event_handler.h	▼ Pro Cor	e		
> c sl_iostream_handles.c	O David	0	Description	
> h sl_iostream_handles.h	(S) Devi	ce Configuration		device network(s).
> h sl_iostream_init_instances.h		Si Build Project		device network(s).
> isl_iostream_init_usart_instances.c				
> h sl_iostream_init_usart_instances.h		Building project		
> isl_power_manager_handler.c				
> id sl_simple_led_instances.c				
> Is_simple_led_instances.h		Invoking Command: make -j8 all		
> 🖻 sli_cli_hooks.c		invoking Command: make ijo ali		
> 🖻 zap-cli.c	~			
<	>	Always run in background		
👫 Debug Adapters 🛛 🔠 Outline 🦳				
% 💥 😂 🗷 X 💥 🗸 🗔 📑		Run in <u>B</u> ackg	ground Cancel Details >>	
⇒ ⊕ EFR32MG12 2.4 GHz 19 dBm RB (ID:440085386)				
y ψ εποείνοτε εν στιε 19 αρτη το (τρ.440005500)				
			× Uninstall	View Dependencies
	🔊 Problems 🛛 🔗 S	earch 🎲 Call Hierarchy 📮 Console 🖄		ê ê 😵 📰 📲 = 🐘 🛃 = 🖬 🖓 - 🗂 - 🗆
	CDT Build Console [Z3Li			
	Finished building:	C:/Users/CAOWENS/ZUC/zigbee_on_uc_alp	ha_3_b382/zigbee_on_uc_alpha_3_b382/protocol/zigbee/a	pp/framework/plugin/green-power-client/green-power-client '
			_b382/zigbee_on_uc_alpha_3_b382/protocol/zigbee/app/f ha_3_b382/zigbee_on_uc_alpha_3_b382/protocol/zigbee/a	ramework/plugin/counters/counters-cli.c pp/framework/plugin/find-and-bind-target/find-and-bind-ta
	Invoking: GNU ARM arm-none-eabi-gcc		-std=c99 '-DUSE_ZAP_CONFIG=1' '-DSL_RAIL_LIB_MULTIPRO	TOCOL_SUPPORT=0' '-DEFR32MG12P432F1024GL125=1' '-DCONFIGU
			_b382/zigbee_on_uc_alpha_3_b382/protocol/zigbee/app/f	ramework/plugin/counters/counters-ota.c
	Invoking: GNU ARM	C Compiler	b382/zigbee on uc alpha 3 b382/protocol/zigbee/app/f	
	building file: C:/	users/chowens/20c/21gbee_on_uc_aipna_5	_0302/21guee_on_uc_aipna_3_0302/protocol/21guee/app/f	railework/progrit/counters/counters-soc.c
	<			>
			1	945M of 1876M

Build completion is reported in the Build Console. The build should complete without errors. If any errors occur, they are highlighted in red in the console. Contact technical support for assistance.



7. In the Project Explorer view, the binaries are shown in a new folder, Binaries.

SI v5_workspace - Z3Light/Z3Light.slcp - Simplicity Studio™	
<u> Eile Edit Navigate Search Project Run W</u> indow <u>H</u> elp	
* + S + B = B = B + S + S + B + B + C + + +	👻 📑 💼 🕈 Welcome 🕑 Recent 🏢 Tools 👤 Install 🌣 Preferenc
🎦 Project Explorer 🛛 📄 🛱 🏹 🖇 🖓 🗖	Lange State
✓ S Z3Light [GNU ARM v10.2.1 - Default] [EFR32MG12P432F1024GL125 - Ge ✓ S Binaries	Z3Light OVERVIEW SOFTWARE COMPO
 > S Z3Light.axf - [arm/le] > Z3Light.bin - [unknown/le] > Z3Light.hex - [unknown/le] > Z3Light.s37 - [unknown/le] 	▼ Filter : Configurable Components
> 🔊 Includes > 👝 autogen > 👝 config	► Advanced Configurators
> 🎥 gecko_sdk_3.2.0 > 👝 GNU ARM v10.2.1 - Default > 🖻 main.c	► Application
Z3Light_callbacks.c Z3Light_pintool Z3Light.pintool Z3Light.slcp	► Bluetooth
Z3Light.slps	► LwIP
	► OpenThread
< >	► Platform
💕 Debug Adapters 🕺 📴 Outline 🛛 🗖	► RTOS
 % X ≅ I = E E ♦ Z3Light (ID:440085390) 	► Services

8. Right-click the <project>.s37 file, and select **Flash to Device**. If you have more than one device connected, you are prompted to select the target. Then the following dialog is displayed.

🛿 Flash Programmer	— 🗆 X
Change Device	
Device Board Name: Wireless Starter Kit Mainboard Board Name: EFR32MG12 2.4 GHz 19 dBm Radii MCU Name: EFR32MG12P432F1024GL125	9 Board
Adapter Name: Z3Light	
Flash Part	
File Type hex O bin Base address	/ 0x0
File	
rs\CAOWENS\SimplicityStudio\v5_workspac	e\Z3Light\GNU ARM v10.2.1 - Default\Z3Light.s37 🗸 Browse
Advanced Settings	
	Erase Program
Flash Erase/Write Protection	
 Select flash range 	▼ 0x0
O Select default sections	🗹 Lock Main Flash 🛛 🗹 Lock User Page
	Protect Remove Protection
Debug Lock Tools	
The unlock function only works using Silicon L Unlocking the chip will erase all data on flash a	
	Unlock Debug Access Lock Debug Access
	Close

- 9. Click **Program**. The binary is flashed to the target device.
- 10. In the Debug Adapters view, right-click the Z3Light device and select Launch Console. In the console window, you will see four tabs: Serial 0 (the Virtual UART interface), Serial 1 (the physical UART interface), Admin (where you can configure the debug adapter such as the WSTK for EFR32), and Debug (where you see raw binary data over the debug interface). Click the Serial 1 tab. Type 'info' and press enter.

No translation	Line terminator: CR-LF (DOS, OS/2, MS Windows) V Buffer size: 100000 Serial 1, 8 bytes out, 1529 bytes in.	
Serial 0 🚔 Serial 1 🚔 Admin 🚔	Debug	
info do		
G String: AppBuilder MFG Co		
de [(>)000B57FFFE491BC2] ch		
rentID [0xC188] nodeID [0xBED6 rentID [0xFFFF] parentRssi] xpan [0x(>)5A9CFFFACCD75F1B] [0]	
ack ver. [7.0.0 GA build 38		
deType [0x02] curity level [5]		
ecurity level [5] etwork state [2] Buffs: 75 /	75	
ocnt: 3		
) 1 [endpoint enabled, devic in (server) cluster: 0x00	s enabled] nwk [0] profile [0x0104] devId [0x0000] ver [0x1]	
in (server) cluster: 0x00	3 (Identify)	
in (server) cluster: 0x00	04 (Groups)	
in (server) cluster: 0x00 in (server) cluster: 0x00		
in (server) cluster: 0x00	08 (Level Control)	
	19 (Over the Air Bootloading)	
out(client) cluster: 0x04 in (server) cluster: 0x10	16 (Occupancy Sensing)	
2 [endpoint enabled, devic	e enabled] nwk [0] profile [0x0104] devId [0x0000] ver [0x1]	
in (server) cluster: 0x00		
in (server) cluster: 0x00 in (server) cluster: 0x00		
in (server) cluster: 0x00	05 (Scenes)	
in (server) cluster: 0x00		
in (server) cluster: 0x00 out(client) cluster: 0x00	/// (Level Control) 9 (Over the Air Bootloading)	
in (server) cluster: 0x03	00 (Color Control)	
in (server) cluster: 0x10	00 (ZLL Commissioning) ice enabled] nwk [0] profile [0xA1E0] devId [0x0000] ver [0x1]	
out(client) cluster: 0x00		
k cnt 1		
vk 0 [Primary (pro)] nodeType [0x2] securityPro		
 	fowely.	
2		

The application has created a distributed network with three endpoints, the last being the Green Power endpoint.

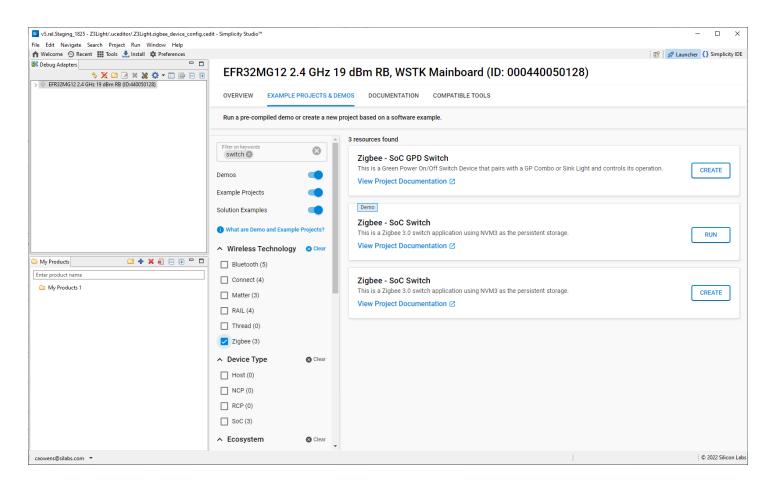
Note: During development you may make changes, work with the console, and then make other changes. While you can load a fresh application image with the device connected, you may have to repair the connection. Alternatively, disconnect from the console by right-clicking the device and selecting **Disconnect**.

If you have multiple devices connected, the device context menu also contains a Rename selection, which you can use to better identify the device roles.

4.2 Create and Load a Zigbee – SoC Switch Application

The **Zigbee – SoC Switch** example procedure follows a slightly different path than that illustrated for Z3Light. Use this shorter path if you do not need to modify devices or compilers.

- 1. Click Launcher in the upper right to return to the Launcher perspective.
- 2. In the Debug Adapters view, select the Switch device.
- 3. Go to the EXAMPLE PROJECTS & DEMOS tab and click CREATE next to the Zigbee SoC Switch application.



4. This opens the third of the three project creation dialogs illustrated in the Z3Light process. Change any values you wish to change, and click **FINISH**.

New Project Wizard	- 🗆 X
Project Configuration	
Select the project name and location.	
C Target, SDK C Examples	Configuration
Project name: Z3Switch_2]	
Use default location	
Location: C:\Users\CAOWENS\SimplicityStudio\v5.master.Staging_1659\Z3Switch_2	BROWSE
With project files: Link to sources Link sdk and copy project sources Copy contents 	
CANCEL	BACK NEXT FINISH

The project opens (abbreviated name Z3Switch). All files are autogenerated. The Z3Switch application is configured as an end device.

- 5. Click the **Build** (hammer) control to build the application.
- 6. When the build is complete, right-click the <project>.s37 file and select Flash to Device. Click Program.

7. When upload is complete, right-click the Switch device and select **Launch Console**. Select the Serial 1 tab, and enter 'info'. The output should resemble the following. The switch has not yet joined a network.

v5.master.Staging_1659 - Adapter: J-Link Silicon Labs (44005012	7)-Switch - Simplicity Studio™	- 🗆 X
File Edit Navigate Search Project Run Window Help	7 Switch Simplicity Stadio	
	🖙 🚽 📷 🖞 🕮 🔓 🏫 Welcome 📀 Recent 🏢 Tools 📩 Install 🌣 Preferences	😭 💋 Launcher { Simplicity IDE
Project Explorer 🛛 📄 🐴 🍸 🕴 🗆	Z3Light.slcp & Z3Switch.slcp & J-Link Silicon Labs (440050127)-Switch 🛛	
Z3Light [GNU ARM v10.2.1 - Debug] [EFR32MG12P432F1024GL125 Z3Switch [GNU ARM v10.2.1 - Debug] [EFR32MG12P433F1024GL125	Image: Second	Serial 1, 8 bytes out, 1489 bytes in.
> 🎼 Binaries	🚄 Serial 0 🚔 Serial 1 🚔 Admin 🚔 Debug	
> 🔊 Includes		^
> 👝 autogen > 👝 config	>	
> 🚌 gecko_sdk_3.2.0	> info info	
> 👝 GNU ARM v10.2.1 - Debug	MFG String: AppBuilder MFG Code: 0x1002 node [())000B57FFFE25F9EF] chan [0] prr [0]	
> <a> main.c Z3Switch_callbacks.c		
23Switch.pintool	parent Di OkfFFE] parent Basi [0] stack ver. [7.0.0 GA build 382]	
Z3Switch.slcp	nodeType [unknown] Security level [5] network state [0] Buffs: 75 / 75	
Z3Switch.slps	Ep cnt: 2	
	<pre>ep 1 [endpoint enabled, device enabled] nvk [0] profile [0x0104] devId [0x0000] ver [0x1] in (server) cluster: 0x0000 (Basic)</pre>	
	out(client) cluster: 0x0003 (Identify) in (server) cluster: 0x0003 (Identify)	
	out(client) cluster: 0x0004 (Groups) out(client) cluster: 0x0005 (Scenes)	
	out(client) cluster: 0x0006 (On/off)	
	in (server) cluster: 0x0007 (On/off Switch Configuration)	
	out(client) cluster: 0x0019 (Over the Air Bootloading) ep 2 [endpoint enabled, device enabled] nwk [0] profile [0x0104] devId [0x0000] ver [0x1]	
	in (server) cluster: 0x0000 (Basic) out(client) cluster: 0x0003 (Identify)	
< >	in (server) cluster: 0x0003 (Identify) out(client) cluster: 0x0004 (Groups)	
🖉 Debug Adapters: 2 🛛 🔚 Outline 🛛 🗖	out(client) cluster: 0x0005 (Scenes)	
Solution (1997) Solution (out(client) cluster: 0x0006 (On/off) out(client) cluster: 0x0008 (Level Control)	
J-Link Silicon Labs (440050127)-Switch (ID:440050127)	out(client) cluster: 0x0019 (Over the Air Bootloading) out(client) cluster: 0x0300 (Color Control)	
> 🜵 J-Link Silicon Labs (440085386)-Light (ID:440085386)	out(client) cluster: 0x1000 (ZLL Commissioning) in (server) cluster: 0x1000 (ZLL Commissioning)	
	Nwk cnt: 1 nwk 0 [Primary (pro)] nodeType [0x3] securityProfile [0x5]>	
	nodeType [0x3] securityProfile [0x5]>	
	>	
		×
		~
	😨 Problems 🔗 Search 🎇 Call Hierarchy 📮 Console 🛛	-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0-0
	CDT Build Console [Z3Switch]	
	.debug_line 1148946 0	
	.debug_str 2416095 0 .debug_frame 50056 0	
	Total 5817318	
	08:50:53 Build Finished. 0 errors, 7 warnings. (took 1m:46s.593ms)	
	¢	>
<u>ا</u> ب		1111M of 1436M
		· · · · · · · · · · · · · · · · · · ·

4.3 Create a Distributed Network and Issue Commands

Once you have downloaded both the light and switch applications to different radio boards, you can create a network. Make sure that the switch device is physically close to the light device.

To see the process from beginning to end, close the distributed network created automatically on the Z3Light device. In the Z3Light console, enter:

> network leave

Then form the network:

> plugin network-creator start 0

Open the network for joining:

> plugin network-creator-security open-network

You now have 55 seconds to join the switch to the open network. On the Z3Switch console, enter:

> plugin network-steering start 0

Enter 'info' on both devices and you will see that both the light and switch are on the same PAN ID.

Project Explorer 🛛 📄 🕏 🍸 🕴 🗖	🚑 Z3Light.slcp 🚔 Z3Switch.slcp 🔐 J-Link Silicon Labs (440050127)-Switch 🖂 🔐 J-Link Silicon Labs (44005386)-Light	- 0
 Z3Switch [GNU ARM v10.2.1 - Debug] [EFR32MG12P433F1024GI K Binaries Includes 	Wo translation ✓ Line terminator: CR-LF (DOS, OS/2, MS Windows) ✓ Buffer size: 100000 Serial 1, 47 bytes out, 4455 bytes in. Serial 0 Serial 1 Serial 1	
 > in Autogen > is autogen > is getes, sidt, 32.0 > is getes, sidt, 32.0 > is autogen > is getes, sidt, 32.0 > is main.o - [arm/le] > is 32Switch.aid - [arm/le] > is 23Switch.aid - [arm/le] 	Svc Disc: Waiting 2 see for discovery to complete Svc Disc: complete. Svc Disc: complete. Svc Disc: complete. Find and bind initiator complete: 0x00 info MFGS String: AppBuilder MFG Code: 0x1002 MFGS String: AppBuilder MFG Code: 0x1002 mode (r):0008/FFFL25/FFF [cham [15] pwr [3] panDD [0x1C44] nodeD [0x3502] % ypan [0x(>)\$CCED66B01342B21] stack ver: [7.0.0 GA build 382]	^
陷 Project Explorer 🛛 📄 🕏 🍞 🕴 🗖	🚢 Z3Light.slep 🛛 🏭 Z3Switch.slep 🛛 👸 J-Link Silicon Labs (440050127)-Switch 🛛 🍪 J-Link Silicon Labs (440085386)-Light 🖂	- 0
 ✓ \$\$ 23Switch (GNU ARM v10.2.1 - Debug) [EFR32MG12P433F1024GL ∧ > \$\$ Binaries > \$\$ config > \$\$ config > \$\$ config > \$\$ GNU ARM v10.2.1 - Debug > \$\$ geck_sdk_32.0 > \$\$ GNU ARM v10.2.1 - Debug > \$\$ autogen > \$\$ geck_sdk_32.0 > \$\$ min.o - [arm/le] > \$\$ geck_sdk_act 	Image: Serial 0 Serial 1 Admin 2 Debug Info MFG String: AppBuilder MFG Code: 0x1002 Impo Info Debug Debug Info MFG String: AppBuilder MFG Code: 0x1002 Insort (CAPDODS) FFERSTING: (Cade: 0x1002 Debug (CADDODS) Debug (CADDODS) Insort (CADDODS) FFERSTING: (Cade: 0x1002 Debug (CADDODS) Debug (CADDODS) Insort (CADDODS) FFERSTING: (Cade: 0x1002 Debug (CADDODS) Debug (CADDODS) Insort (CADDODS) FFERSTING: (CADDODS) (CADDODS) Debug (CADDODS) Insort (CADDODS) FFERSTING: (CADDODS) (CADDODS) Debug (CADDODS) Insort (CADDODS) Debug (CADDODS) Debug (CADDOS) Insort (CADDODS) Debug (CADDOS) Debug (CADDOS) Insort (CADDOS) Debug (CADDOS) Debug (CADDOS)	

To send a command from the switch to the light, first build the toggle command and load it into a TX buffer:

> zcl on-off toggle

Then send the command to the light, using the device's Node ID (also known as the short ID), which is 0x0738. To toggle the device's LED on and off, on the Z3Switch device enter:

> send 0x0738 1 1

The light console should show:

T00000000:RX len 3, ep 01, clus 0x0006 (On/off) FC 01 seq 01 cmd 02 payload[] On/Off set value: 01 02 Toggle on/off from 00 to 01

On the WSTK board, LED1 should turn on.

4.4 Change to a Centralized Network

In a distributed network, any router can issue network security keys. In a centralized network, a coordinator has the functionality of a Trust Center. Each node that joins is authenticated by the Trust Center before it can join the network. The coordinator is always node ID 0x0000. To change Z3Light to a Coordinator:

- 1. Go to the Z3Light.slcp tab.
- 2. On the SOFTWARE COMPONENTS tab, search for 'device config'.
- 3. Select the Zigbee Device Config component and click Configure.

4. Change the Primary Network Device Type to 'Coordinator or Router'. The change is saved automatically.

Bee device configuration Primary Network Device Type Coordinator or Router Router End Device	e.html 🛛 🚨 Z3Light.slcp 🚺 Zigbee Device Config 🗙		
Primary Network Device Type Coordinator or Router Coordinator or Router Router End Device	gbee Device Config	Pin Tool View Sour	ce >
Primary Network Device Type Coordinator or Router Coordinator or Router Router End Device	igBee device configuration		
Coordinator or Router Router End Device	Primary Network Device Type	Primary Network Security Type	
Router End Device	Coordinator or Router 👻	ZigBee 3.0 Security 👻	
End Device	Coordinator or Router		
Sleenv End Device	work		
stopy in beneo	Sleepy End Device		

- 5. All coordinators must also have Concentrator functionality. Close the Component Editor and, back on the Software Components tab, search for 'concentrator'.
- 6. Select the Concentrator Support component and click Install. Changes are saved automatically.

readme.html	🚢 Z3Light.slcp 🗙 🚺	Zigbee Device Config	readme.html	🚢 Z3Switch.slcp				- 0
Z3Light	OVERVIEW	SOFTWARE COM	PONENTS	CONFIGURATION	TOOLS			
T Filter com	ponents by 🏚 Configu	irable 🗌 🔮 Insta	illed 🗌 💄 In	nstalled by you 🗌	SDK Extensions	Quality -	Search keywords, component's name concentrat 😒	8
▼ Zigbee				Concentrator	Support		Ins	tall
▼ Utility								
	sentrator Support		•	will act as a con defined as part of Quality PRODUCTION Depende Zigbee_concentra > Zigbee Depende	ncentrator (sink) in the of the source routing lib encies ator requires 1 components ents quire zigbee_concentrator	network. Note NCP/SoC rary.	-request (MTORR) so that the de source route table size can now	

- 7. In the Project Explorer view, select Z3Light and click the Build (hammer) icon.
- 8. When build is complete, right-click the Z3Light device and click Disconnect.
- 9. When the build is complete, right-click the <project>.s37 file and select Flash to Device. Click Program.
- 10. Launch the console for the Z3Light device.

Next, take down the distributed network and start the centralized network.

> network leave

Then form the network:

> plugin network-creator form 1 0xfeed 20 11

Enter 'info'. The PAN ID is that specified, and the Node ID is 0x000

🛓 Z3Light.slcp 🛛 🚑 Z3Switch.slcp 🛛 🔗 Z3Light 🖾 🦃 Z3Switch	- 6
No translation V Line terminator: CR-LF (DOS, OS/2, MS Windows) V Buffer size: 100000 Serial 1, 174 bytes out, 6453 bytes in.	
🚖 Serial 1 🚔 Admin 🚖 Debug	
NWK Steering stack status 0x90 EMEER_NETWORK_UP 0x0000 NWK Creator: Form. Channel: 11. Status: 0x00 Form distributed network complete: 0x00 NWK Creator: Form: 0x00 Z3Light>Find and Bind Target: Start target: 0x00 Find and bind target start: 0x00	^
Z3Light>info info MFG String: AppBuilder MFG Code: 0x1002 mode [(>)000B57FFFE491BC2] chan [11] pur [20] parmetTD: f0xeFFE1 marmetPsei 10 stack ver. [7.0.0 GA build 9] nodeType [0x01] Security level [5] metwork state [2] Buffs: 75 / 75	
<pre>Ep ont: 3 p1 [endpoint enabled, device enabled] nvk [0] profile [0x0104] devId [0x0000] ver [0x1] in (server) cluster: 0x0000 (Basic) in (server) cluster: 0x0003 (Identify) in (server) cluster: 0x0004 (Groups) in (server) cluster: 0x0006 (On/off) in (server) cluster: 0x0006 (On/off) in (server) cluster: 0x0006 (On/off) in (server) cluster: 0x0019 (Over the Air Bootloading) out(client) cluster: 0x0019 (Over the Air Bootloading) out(client) cluster: 0x0000 (ZLL Commissioning)</pre>	
<pre>pp 2 [endpoint enabled, device enabled] nvk [0] profile [0x0104] devId [0x0000] ver [0x1] in (server) cluster: 0x0000 (Basic) in (server) cluster: 0x0004 (Groups) in (server) cluster: 0x0005 (Gcenes) in (server) cluster: 0x0006 (Level Control) out(client) cluster: 0x0008 (Level Control) out(client) cluster: 0x0008 (Level Control) in (server) cluster: 0x0008 (Level Control) in (server) cluster: 0x0008 (Level Control) in (server) cluster: 0x0008 (Level Control) en (server) cluster: 0x1000 (CLIC Control) in (server) cluster: 0x1000 (CLIC Control) en (server) cluster: 0x1000 (CLIC Control) in (server) cluster: 0x1000 (CLIC Control) in (server) cluster: 0x1000 (DLIC Control) in (server) cluster: 0x100 (DLIC Control) in (server) cluster: 0x100 (DLIC Control) in (server) cluster: 0x1000 (DLIC Control) in (server) cluster: 0x100 (DLIC Control) in (server) cluster: 0x1000 (DLIC Control) in (server) cluster: 0x1000 (DLIC Control) in (server) cluster: 0x1000 (DLIC Control) in (server) cluster: 0x100 (DLIC Control) in (</pre>	
out(client) cluster: 0x0021 (Green Power) Nwk cnt: 1 node[ype [0x1] securityProfile [0x5]Z3Light> Z3Light}	

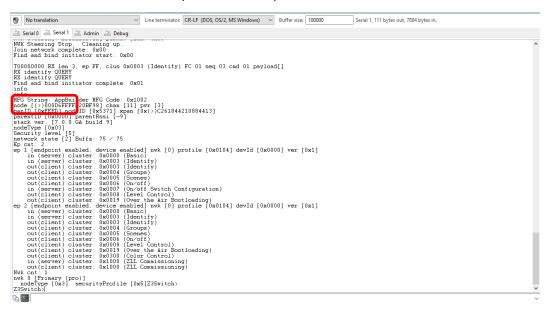
Open the network for joining:

> plugin network-creator-security open-network

You now have 55 seconds to join the switch to the open network. On the switch device:

```
> network leave
> plugin network-steering start 0
```

Enter 'info'. The PAN ID is that specified, and the parent ID is 0x0000.



To send a command from the switch to the light, first build the toggle command and load it into a TX buffer:

> zcl on-off toggle

Then send the command to the light, using the device's Node ID 0x0000. To toggle the device's LED on and off, on the Z3Switch device enter:

> send 0 1 1

The light console should reflect the command and, on the WSTK board, LED1 should toggle from its previous state.

4.5 About the Zigbee Cluster Configurator

Additional customizations can be made through the Zigbee Cluster Configurator. Using this tool is described in detail in AN1325: Zigbee Cluster Configurator User's Guide.

4.6 About Bootloaders

All Silicon Labs examples require that a bootloader be installed. A bootloader is a program stored in reserved flash memory that can initialize a device, update firmware images, and possibly perform some integrity checks. Silicon Labs networking devices use bootloaders that perform firmware updates in two different modes: standalone (also called standalone bootloaders) and application (also called application bootloader performs a firmware image update by reprogramming the flash with an update image stored in internal or external memory. For more information about bootloaders see *UG103.6: Application Development Fundamentals: Bootloading*.

In March of 2017, Silicon Labs introduced the Gecko Bootloader, a code library configurable through Simplicity Studio's IDE to generate bootloaders that can be used with a variety of Silicon Labs protocol stacks. The Gecko Bootloader is used with all EFR32xG parts.

The Gecko Bootloader works with a specialized firmware update image format, ending in the extension .gbl (the GBL file). To create a GBL file from an .s37 or binary, follow the instructions in <u>UG162</u>: <u>Simplicity Commander Reference Guide</u>, section 6.7.1, GBL File Creation. The exact format of the GBL file depends on the hardware you selected.

Note: When working with the Gecko Bootloader, you must use Simplicity Commander to enable some configuration option such as security features. See UG489: Silicon Labs Gecko Bootloader User's Guide for GSDK 3.3 and Higher.

By default, a new device is factory-programmed with a bootloader. If you have a new device, haven't cleared the bootloader region for your part or have a supported bootloader image already flashed on your device, you do not need to flash a bootloader. Once you have installed a bootloader image, it remains installed until you erase the device.

If you need to load a bootloader, select an example, such as **SPI Flash Storage Bootloader (single image)**, and build it and flash it as described above. For more information about the Gecko Bootloader, see UG489: Silicon Labs Gecko Bootloader User's Guide for GSDK 4.0 and Higher.

5 Using the Network Analyzer

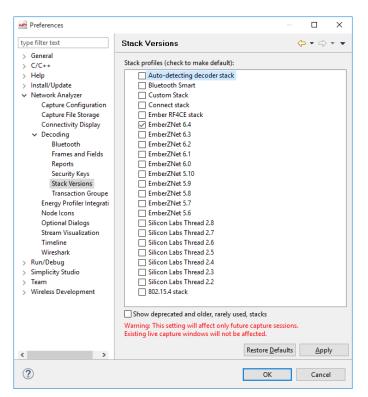
Now that your network is set up, you can evaluate the data being transmitted using the Network Analyzer tool. Network Analyzer helps you debug network connectivity by displaying radio packets and certain debug interface events in a format that is easy to visualize and analyze. See the online <u>Simplicity Studio User's Guide</u> for more information.

1. Click the Launcher button in the upper right and select Network Analyzer from the Tools menu. The Network Analyzer opens with your console window(s) still displaying data.

<pre> Control is in the interval is interval in the interval is interval interva</pre>				
<pre> Control is in the interval is interval in the interval is interval interva</pre>	🕶 Network Analyzer - Adapter: J-Link Silicon Labs (440085390) -	r - Simplicity Studio ™ –	- 🗆	\times
<pre>> Deckes 2 %</pre>	<u>File E</u> dit F <u>i</u> lters <u>W</u> indow <u>H</u> elp			
<pre> Link Silcon Lab: (4005383</pre>	i 🗁 L1 Ng M_ 2) 🖅 👺 i 🛷 🌾 i 🐶 🕜 i 🗂 🕶 🗐 N	🐘 ഈ 🖏 ▾ 🏷 🗢 マ→ 📲 💼 👘 🚼 👔 👘 👔	Network Ar	alyzer
<pre>v Lets Sicon Lets (4400338) v EFF220X012932F1024G1135 V EFF220X01295</pre>	• Devices: 2 🔀 🗖 🗖	😰 Z3SwitchSoc.isc 🛛 🗞 J-Link Silicon Labs (440085388) 😨 Z3LightSoc_3.isc 🛛 🎉 J-Link Silicon Labs (440085390) 🔀		
P Not Logged In 👘 👘 🖉 2017 Silicon Lat	Image: Second	Operation Construction Serial Admin Debug 231:ghtSoc_3:231:ghtSoc_3:231:ghtSoc_3:251:ghtSoc_3:ghtSoc_3:251:ghtSoc_3:251:ghtSoc_3:ghtSoc_3:251:ghtSoc_3:ghtSoc_3:251:ghtSoc_3:ghtSoc_3:251:ghtSoc_3:ghtSoc_3:251:ghtSoc_3:ghtSoc_3:251:ghtSoc_3:ghtSoc_	31 FE FF	
	📲 Not Logged In		© 2017 Silic	on Lab

Make sure that Network Analyzer is set to decode the correct protocol. Select Window > Preferences > Network Analyzer > Decoding > Stack Versions, and verify it is set correctly. If you need to change it, click the correct stack, click Apply, and then OK.

Note: If you are working with a Zigbee+Bluetooth Dynamic Multiprotocol application, Auto-detecting decoder stack must be selected.



3. To make sure that packets decode correctly, manually enter the NWK key. In either the Switch or Light console window, type the following, being sure to include the 's' in 'keys':

Keys print

4. In the information returned, find the network key and copy it:

NWK Key: EF DE OC 69 5B 72 6E C4 41 27 C6 E6 F1 36 26 26

5. In Window > Preferences, open Network Analyzer > Decoding > Security Keys, click **New**, name the new entry, and paste the copied key into it. Click **Apply**. Click **OK** to leave.

ype filter text	Socuri	ty Keys			⊳ - ⇔ -
General	Securi	ty Keys			
C/C++	Check t	o activate. Enter	16 char ASCII or 32 char Hex		
Help	AT	Name	Key	Touched	New
Install/Update		App Bootloa	65 6D 62 65 72 20 45 4D 32 35 30 20 61 62 6C 4C	Sep 7	
 Network Analyzer 		App Bootloa	65 6D 62 65 72 20 45 4D 32 35 30 20 61 62 6C 4C	Sep 7	Import
Capture Configuration		ConnectSa	AA	Sep 7	Clone
Capture File Storage		Default Fra	05 05 05 05 05 05 05 05 05 05 05 05 05 0		Cione
Connectivity Display				Sep 7	Delete
✓ Decoding		Default Fra	04 04 04 04 04 04 04 04 04 04 04 04 04 0	Sep 7	
Bluetooth		Default IP Te		Sep 7	Invert
Frames and Fields	✓ N	Test 090918	C2 50 65 03 8B 75 52 C1 63 DE EB F3 7B 01 F6 70	Today	Clear All
Reports		Sensor/Sink	5A 69 67 62 65 65 20 53 65 63 75 72 69 74 79 21	Sep 7	Clear All
Security Keys		Sensor/Sink	65 6D 62 65 72 20 45 4D 32 35 30 20 63 68 69 70	Sep 7	Run HMAG
Stack Versions		Smart Energ	12 33 33 33 33 33 33 33 33 33 33 33 33 33	Sep 7	
Transaction Groupe		Smart Energ	56 77 77 77 77 77 77 77 77 77 77 77 77 77	Sep 7	ASCII edit
Energy Profiler Integrati	V C	Standalone	65 6D 62 65 72 20 45 4D 32 35 30 20 6C 69 6E 6B	Sep 7	
Node Icons	V C	Standalone	65 6D 62 65 72 20 45 4D 32 35 30 20 6E 77 6B 20	Sep 7	
Optional Dialogs	V C	Super Paren	53 75 70 65 72 50 61 72 65 6E 74 4C 69 6E 6B 4B	Sep 7	
Stream Visualization	V C	Super Paren	53 75 70 65 72 50 61 72 65 6E 74 4E 65 74 77 4B	Sep 7	
Timeline	V C	ZigBee 3.0 d	43 5F 91 1D B5 8C AC B4 12 EC 7E 8B 6B 58 40 B2	Sep 7	
Wireshark		ZigBee 3.0 d	B9 5F 9E 42 5C 48 87 8B FE 74 3C F0 FA CE 25 91	Sep 7	
> Run/Debug	V C	ZigBeeAllian	5A 69 67 42 65 65 41 6C 6C 69 61 6E 63 65 30 39	Sep 7	
Simplicity Studio					
> Team					
Wireless Development					
	<			>	
				[] .	
	✓ Save	decryption keys	in ISD files 🛛 Disable keys when not used for	365 days	
				Restore Defaults	Apply
>					

- 6. Right-click on the light or the switch device, and select **Start Capture**. Do the same for the other device.
- 7. If you are in an environment with a number of wireless devices, you may have a very noisy Network Analyzer environment, as reflected both in the event traffic and in the map. To show additional information in the map, click on the map.

On the toolbar, click the PAN ID button, circled in the following image.

on: Ember IP st	tack, Thread 0.99 - Simplicity	y Studio ™					- 0 ×
3 - 🕼	ᡚ ▼ ᢓ ▼ ∜ � ♥ ⊂	⇒ + 🗑 🔗	Գ ա∣ Գ ե <mark>ւ Մն</mark> ∮ՈՈձ Թ Ղ Զ ։	q 🖹 +≣ , 8 + 8 ≖ T 🕖 ▶ 💌 🕸) 🖷		
rk Analyzer			_				
- 8	G server.isc G a	client.isc	😵 J-Link Silicon Labs (440050127)	🔗 J-Link Silicon Labs (440050128)	🔺 *Live 🛛		
🌣 🗝 🔳 📴	0 saved filters AND						• 🔅 🛇 🚔
RD4153A) (Rev A01)	34.00 p/s 0.003s						12.672s
RD4153A)	Time:not set Real time:	:N/A Nodes:0	Event:none			Event Detail	
(Rev A01)	•						

8. Right-click on the representation of your Switch device (the dot that has the same ISA3 adapter name or WSTK name or J-Link serial number as the device) and select **Show only this PAN**.

To filter transactions:

1. Press button 0 on the switch to get a complete transaction (in blue).

 Network Analyzer - Live capture streamStack version: Emb <u>E</u>dit F<u>i</u>lters <u>W</u>indow <u>H</u>elp 		-											
> 🗟 🕼 🔍 🌮 🛸 🛷 💯 🐶 🔂 🗂 🕶 📓	1		⇔ - □	🛇 📲 🔮 🗇 🔟 🛛 🦕 կթ	Pip 🖉 🖬	1 🚱 🗸	Q Q 🗎	÷≣ " 8	3 +₿ т	Т 🥖 🕹	▶ 🔤 🔆		
😭 📊 Launcher { Simplicity IDE 🔔 Network Analyz													
Devices: 2 🕱		3SwitchSoc.isc	<i>5</i> 0 1	-Link Silicon Labs (440085388)	Z3LightS	oc 3 isc	🦃 J-Link S	ilicon La	be (4/100	85300)	*Live 🔀		
				-Enrik Shircon Eabs (440005500)	23Lights	oc_sase	I WE PERINCO		DS (4400	65550)		→I.	
⊘ □ 2 ∞ ∞ √ 1 □ □ Output: O	0 sa	aved filters A										×	
 EFR32MG12 2.4GHz 10 dBm (BRD4162A Rev A01) 	611		111s	to ask to constra t		Y	k		in.	had as		A data bi ka ka ma	in the state
EFR32MG12P332F1024GL125		001s 	<u>N A N</u>		a di da		- Midelum III	1 518 11	عياللمه	an dahilik dala	بالسبام الأراد	A DARMA AND A A A A A A	648.612
 Wireless Starter Kit Mainboard (BRD4001A Rev A01) J-Link Silicon Labs (440085390) [capture] 	Tim	ne:50.111093s	Real time:	N/A Nodes:7 Event:ZigBee AP	S transaction	PAN filter:	6D3F					Event Detail	
 EFR32MG12 2.4GHz 10 dBm (BRD4162A Rev A01) 												NWK crypto: ROOT, 3E	
EFR32MG12P332F1024GL125		•			•							 IEEE 802.15.4 [10 bytes] PHY Header: 0x30 	
> 🌠 Wireless Starter Kit Mainboard (BRD4001A Rev A01)		0004400853	90		000	44008538	18					Packet Length: 4	8
		0002			000	1						Frame Control: 0x88	
		PAN: 6D3F			PAN	6D3F						Frame Type: Data	a (1)
												Security Enabled	
												Frame Pending: 1	
												Ack Required: tru Intra Pan: true	ue
												Frame Version: 2	003 (0)
												Reserved: 0x00	
												Destination Add	
	Tra	nsactions tota	al:2 shown	n:2								Source Address M	Mode: Short
		Time	Durati	Summary	NWK Src	NWK	Dest P#	M#	E#	Status		Sequence: 0x72 Destination PAN ID:	0x6D3E
		50.111093	0.079	ZCL: Toggle	0001	0002	3)	Short Destination A	
	Ð	50.158888	0.042	ZCL: DefaultResponse	0002	0001	4						,
												Hex Dump [54 bytes]	▽
												FC 30 61 88 72	.0a.r
												3F 6D 02 00 01 00 48 22 02 00	? <u></u>
												01 00 1E DB 28	
												OE 00 00 00 89 5B 31 FE FF 57	jiv
	Eve	nts total:6,24	1 shown:6	,241 Decoders: EmberZNet 5.10	ZigbeePro						8	0B 00 00 61 1D 2B 87 9C E0 BC	a. +
		Time	Туре	Summary		MAC Src	MAC Dest	Status			^	92 F6 1C 8C 14 04 07 03 E4 92	
		648.148323 I	Packet	802.15.4 Ack				Droppe	ed			FD 04 05 00	
			Packet	Data		0000	560C	Droppe					
			Packet	Packet				Droppe					
			Packet	Data		0000	560C	Droppe					
			Packet Packet	802.15.4 Ack Packet				Droppe					
			Packet Packet	Data		0000	FFFF	Droppe				_	
		0.0000000000000000000000000000000000000	achet	0.000				5.opp			~	Raw captured data (1 more)

2. Click on one of the blue transactions.

- 3. In the Event Detail, expand IEEE802.15.4 and scroll down until you see the Destination PAN ID
- 4. Right-click on it, and select **Add to filter**.
- 5. Apply the filter by clicking the icon next to the green filter expression field, circled in the following image.

Vet 5.10, ZigbeePro - Si	mplicity Studio ™				-		×
1111 • 121 • 14 4	> - ⇔ - i 🕲 🕫 ኛ 🔟 <mark>So</mark> 4o	Ro & 11 & @ V	l 🗛 🔍 🖹 +E +8 +5 🔻 T 🌽	🕹 🕨 🔤 🍀 👯	bi		
Z3SwitchSoc.isc	🔗 J-Link Silicon Labs (440085388)	Z3LightSoc_3.isc	🔗 J-Link Silicon Labs (440085390)	▲ *Live ⊠		_	
0 saved filters AND	fifteenFour.destPan == 0x6d3f					30	0 🖻
54,00,p 50,111 0.001s 4444	المستحد الدائلة ومعرفان الله والمالية	he is he bear	da anti i di en di teli adere de de.	مر المانية سالي الم	A sould a soul as the	739	.910s
Time:50.111093s Rea	al time:N/A Nodes:7 Event:ZigBee APS	transaction PAN filter:	6D3F		Event Detail	, 3E 66 A0	36 15 个

Now when you press button 0 you can clearly see each event associated with the transaction.

Network Analyzer - Live capture streamStack version: Ember	ZNet 5.10, ZigbeePro -	Simplicity	y Studio ™								- 🗆
<u>E</u> dit Filters <u>W</u> indow <u>H</u> elp											
📙 🕼 🖳 🌮 👺 🥓 🌮 🐶 🕜 🖬 🕶 🔚 🛛	🖹 🖢 🕶 🖗 🕶 🏷	⇔ -	ି 📲 🧐 🤣 💝 🔟 🛼 🗛	Rid 🖉 🖬 🖧	or 11 🖗	<u> </u>	÷≣ ₊₿		Т 🥖 坐 🕨	₩ ‡	<u>.</u>
😚 📊 🖍 Launcher 🛛 👌 Simplicity IDE 🔍 Network Analyzer											
Devices: 2 🛛 🗖 🗖	C Z3SwitchSoc.isc	🤗 J-	-Link Silicon Labs (440085388)	Z3LightSoc_	3.isc 🔗	J-Link Si	licon Lal	bs (4400	85390) 📃 📥 *	Live 🛛	-
🍫 😂 🗷 🗙 🌣 🔻 🔲 📴	0 saved filters A	ND fiftee	nFour.destPan == 0x6d3f								 > 🔅 🛇
	58.00 58s11	~	V								
EFR32MG12 2.4GHZ 10 dBm (BRD4102A Rev A01) EFR32MG12P332F1024GL125	0.001:	سالهتا	it has been been and been	man she was	مهميناه بالحد	أوالمانس	Will	يقدأ أن	بالدي أبله يعاليه	بغيانة أعداده	1 . 29 . 1978 . 29
> 🌠 Wireless Starter Kit Mainboard (BRD4001A Rev A01)	Time:50.111093s	Real time:	N/A Nodes:7 Event:ZigBee APS	Stransaction PAI	N filter: 6D3F					۵	🛬 Event Detail
했 J-Link Silicon Labs (440085390) [capture] > [2] EFR32MG12 2.4GHz 10 dBm (BRD4162A Rev A01)											NWK crypto: ROOT, 3E 66 A0 36 1
EFR32MG12P332F1024GL125 Wireless Starter Kit Mainboard (BRD4001A Rev A01)				•		•	•		• •	•	PHY Header: 0x30
wireless statter kit Mainboard (bKD4001A KeV A01)	0004400853	90		000440 Switch	085388				(0000) (0C3 1998-10): (311-1998-10):		Packet Length: 48 Frame Control: 0x8861
	PAN: 6D3F			0001 PAN: 6			,			()	Frame Control: 0x0001 Frame Type: Data (1)
				THR. C	DJF						Security Enabled: false
											Frame Pending: false
											Ack Required: true
											Intra Pan: true
											Frame Version: 2003 (0) Reserved: 0x00
	<									>	Destination Address Mode
	Transactions tota	al:4 shown	:4							⊽ 🗆	Source Address Mode: She
	Time	Durati	Summary	NWK Src	NWK Dest	P#	M#	E#	Status		Sequence: 0x72
	50.111093	0.079	ZCL: Toggle	0001	0002	3	IVI-	2.0	Status		Destination PAN ID: 0x6D3F
	50.158888	0.042	ZCL: DefaultResponse	0002	0002	4					Short Destination Address: 0x
	966.655824	0.048	ZCL: Toggle	0001	0002	3					<
	966.697888	0.034	ZCL: DefaultResponse	0002	0001	3					Hex Dump [54 bytes]
											FC 30 61 88 72 .0a.r 3F 6D 02 00 01 ?=
											00 48 22 02 00 .H [*]
											01 00 1E DB 28(0E 00 00 00 89
	Events total:9,50	1 shown:7	3 Decoders: EmberZNet 5.10, Zi	gbeePro							5B 31 FE FF 57 [1 0B 00 00 61 1Da.
	Time	Туре	Summary	MA	C Src MA	C Dest	Status			^	2B 87 9C E0 BC + 92 F6 1C 8C 14
	943.026938	Packet	Link Status	000	2 FFF	F					04 07 03 E4 92 FD 04 05 00
	959.391960	Packet	Link Status	000		F					
		Packet	ZCL: Toggle	000							
		Packet	ZCL: DefaultResponse	000							
		Packet	APS Ack	000							
		Packet Packet	APS Ack Link Status	000		-					
	2/4.210202	racket	LITIK DURUS	000	- rrr	r i				~	Raw captured data (1 more)

When analyzing more complex networks, you can drag and reposition the items shown in the map. By right-clicking on a device, you can also show connectivity and add labels. Labelling is useful not only in map, but also in the log. To label the full log, click **From beginning**.

🗢 Node label			×
Enter label:			
Switch			
From beginning			
◯ Starting at time:	50.111093		
		ОК	Cancel

6 Next Steps

6.1 Example Applications

Explore configuring the example application to meet your needs. Much of the software configuration can be done through components and the Zigbee Cluster Configurator.

6.2 PIN Tool

Simplicity Studio offers a PIN tool that allows you to easily configure new peripherals or change the properties of existing ones. See the online Simplicity Studio v5 User's Guide: Pin Tool for more information.

6.3 Energy Profiler

You may also want to explore using the Energy Profiler tool. Energy Profiler provides energy debugging capability by displaying graphical real-time energy consumption information. This can be particularly useful for developing a low-power application. See the online <u>Simplicity</u> <u>Studio 5 User's Guide: Energy Profiler</u> for more information.

Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!



www.silabs.com/IoT



www.silabs.com/simplicity



www.silabs.com/quality



Support & Community www.silabs.com/community

Disclaimer

Silicon Labs intends to provide customers with the latest, accurate, and in-depth documentation of all peripherals and modules available for system and software implementers using or intending to use the Silicon Labs products. Characterization data, available modules and peripherals, memory sizes and memory addresses refer to each specific device, and "Typical" parameters provided can and do vary in different applications. Application examples described herein are for illustrative purposes only. Silicon Labs reserves the right to make changes without further notice to the product information, specifications, and descriptions herein, and does not give warranties as to the accuracy or completeness of the included information. Without prior notification, Silicon Labs may update product firmware during the manufacturing process for security or reliability reasons. Such changes will not alter the specifications or the performance of the product. Silicon Labs shall have no liability for the consequences of use of the information supplied in this document. This document does not imply or expressly grant any license to design or fabricate any integrated circuits. The products are not designed or authorized to be used within any FDA Class III devices, applications for which FDA premarket approval is required or Life Support Systems without the specific written consent of Silicon Labs. A "Life Support System" is any product or system intended to support or sustain life and/or health, which, if it fails, can be reasonably expected to result in significant personal injury or death. Silicon Labs products are not designed or authorized for military applications. Silicon Labs product shall under no circumstances be used in weapons of mass destruction including (but not limited to) nuclear, biological or chemical weapons, or missiles capable of delivering such weapons. Silicon Labs disclaims all express and implied warranties and shall not be responsible or liable for any injuries or damages related to use of a Silicon Lab

Trademark Information

Silicon Laboratories Inc.[®], Silicon Laboratories[®], Silicon Labs[®], SiLabs[®] and the Silicon Labs logo[®], Bluegiga[®], Bluegiga Logo[®], EFM[®], EFM32[®], EFR, Ember[®], Energy Micro, Energy Micro logo and combinations thereof, "the world's most energy friendly microcontrollers", Redpine Signals[®], WiSeConnect, n-Link, EZLink[®], EZRadio[®], EZRadio[®], Gecko[®], Gecko OS, Gecko OS Studio, Precision32[®], Simplicity Studio[®], Telegesis, the Telegesis Logo[®], USBXpress[®], Zentri, the Zentri logo and Zentri DMS, Z-Wave[®], and others are trademarks or registered trademarks of Silicon Labs. ARM, CORTEX, Cortex-M3 and THUMB are trademarks or registered trademarks of ARM Holdings. Keil is a registered trademark of ARM Limited. Wi-Fi is a registered trademark of the Wi-Fi Alliance. All other products or brand names mentioned herein are trademarks of their respective holders.



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

www.silabs.com