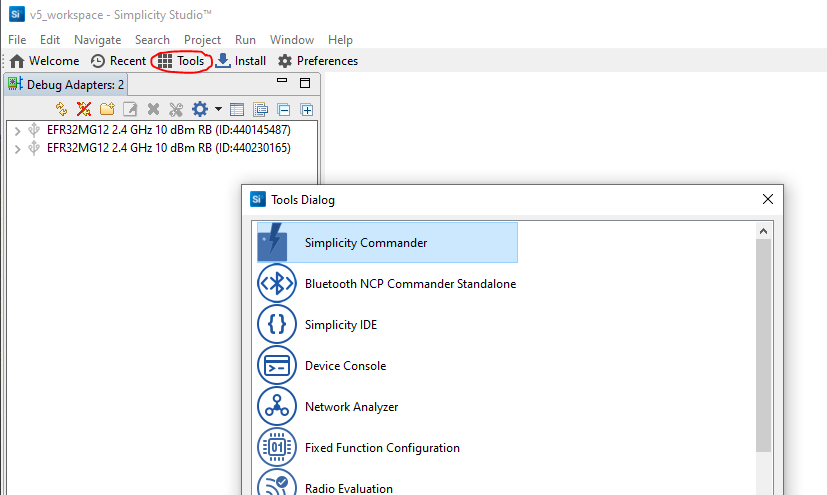
Setup of Simplicity Studio:

* Install gecko SDK … to be finished.

Setup Flash-Tool for the Gecko

* Make sure the **Simplicity Commander** is installed.



Original Source:

https://github.com/SiliconLabs/zigbee\_applications/blob/f7059cdff81e6c57b711aaa2f0bd3634e2300b6f/zigbee\_bootcamp/Zigbee-Hands-on-Forming-and-Joining/Zigbee-Hands-on-Forming-Joining.md

**1. Introduction**

In this worksheet we provide a step-by-step guide to create, build and run ZigBee 3.0 applications based on EmberZNet Stack 7.1.0. If you use a later release in the future, most of the instructions should still be applied, although there could be minor differences not foreseen at the time of this document. These exercises help you get familiar with ZigBee 3.0 in the EmberZNet Stack, Simplicity Studio v5 development environment, and the Wireless Start Kit (WSTK) with EFR32MG12 SoC. We assume that you have two WSTK with SoC Radio boards and the following software (Simplicity Studio and EmberZnet SDK; this is part of the Gecko SDK).

**1.1. Application features**

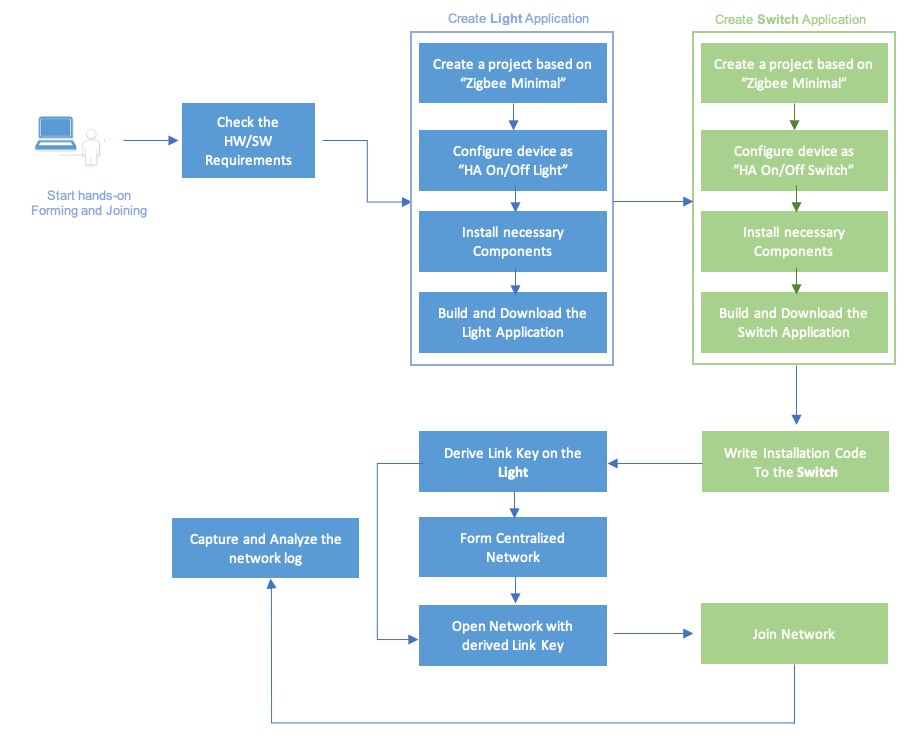
The boot camp series hands-on workshop will cover the four functionalities below, and the application development is split into four steps respectively to show how an application should be built up from scratch. The exercise in this documentation is the first exercise in the "Zigbee Boot Camp" series.

* **The 1st phase, forming a basic network (Light), and a joining process (Switch) will be done using install code.**
* The 2nd phase, will prepare the devices to transmit, receive, and process the On-Off commands using APIs.
* The 3rd phase, the Switch will have a periodic event to execute custom code, in this case a blinking LED.
* The 4th phase, the Switch will be able to store any custom data in its flash using Non-volatile memory.

**1.2. Purpose**

This tutorial will give an overall knowledge about how to build a Light and Switch device from scratch. By the end of this Lab, the user will be familiar with Simplicity Studio and Zigbee fundamentals. The network will consist of two devices using the EFR32MG12 SoCs (BRD4162A).

* One of the devices is the Light. Since the created network is centralized, the Light will work as the Coordinator and Trust Center of the network. This device forms and opens the network, permits other devices to join, and manages security keys.
* The other device is the Switch. It joins the network opened by the Light and sends On-Off commands to the Light. The figure below illustrates the workflow of this hands-on.

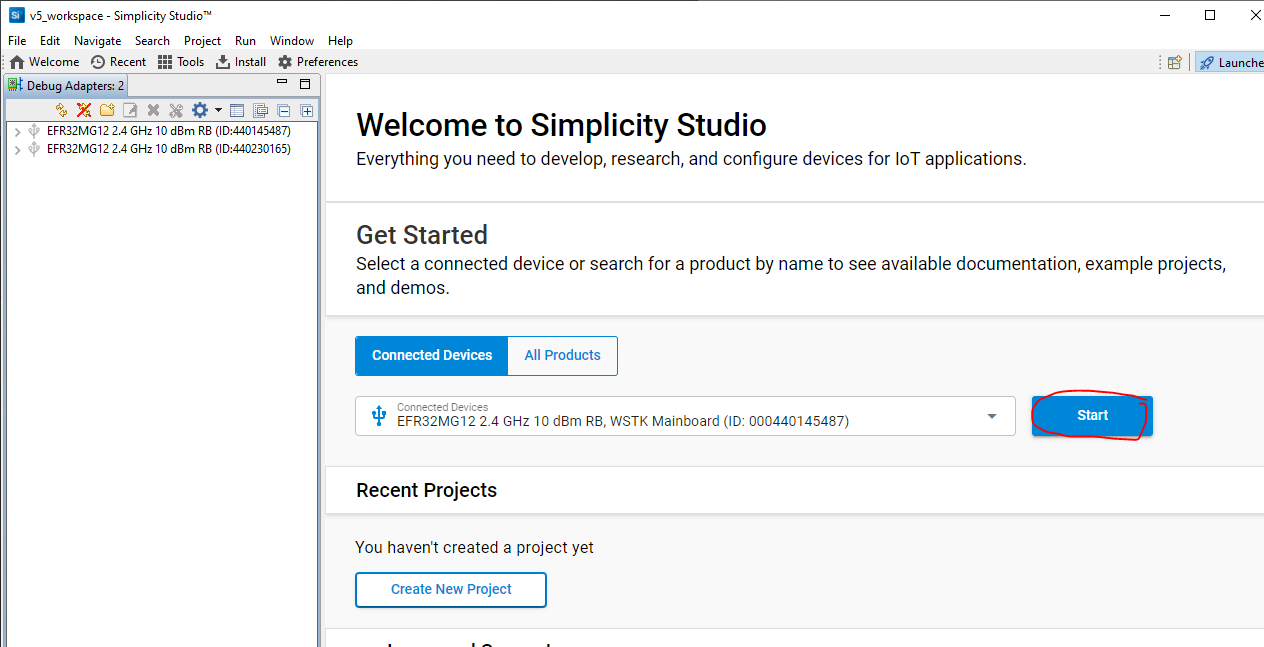


**Figure 1-1 Light and Switch Application Workflow**

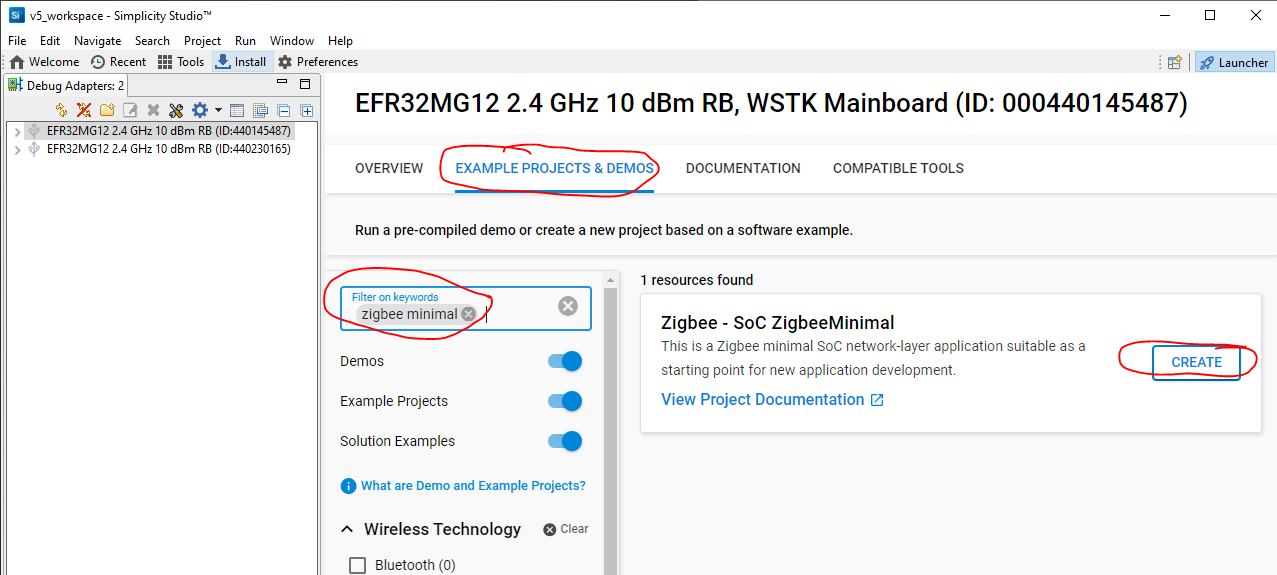
**2. Create Light Application**

As presented in Section 1.2, the purpose of the Light is to form and open the network. The Silicon Labs Project Wizard will be used for creating the application. Silicon Labs Project Wizard is an interactive GUI tool that allows developers to create and configure most of their Zigbee applications.

Start Simplicity Studio go to Welcome-View and click on Start

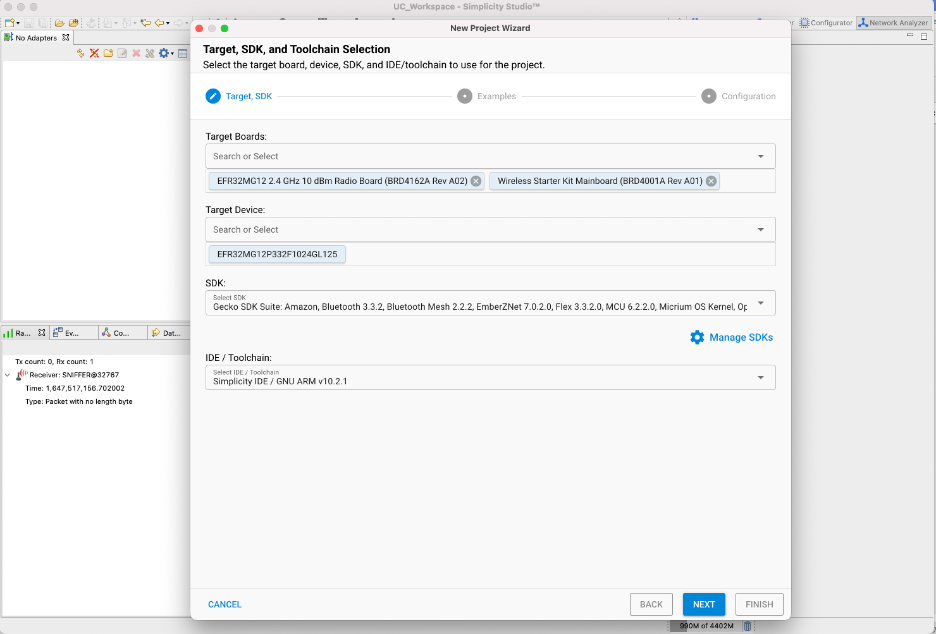


Create a new project from an existing minimal example:



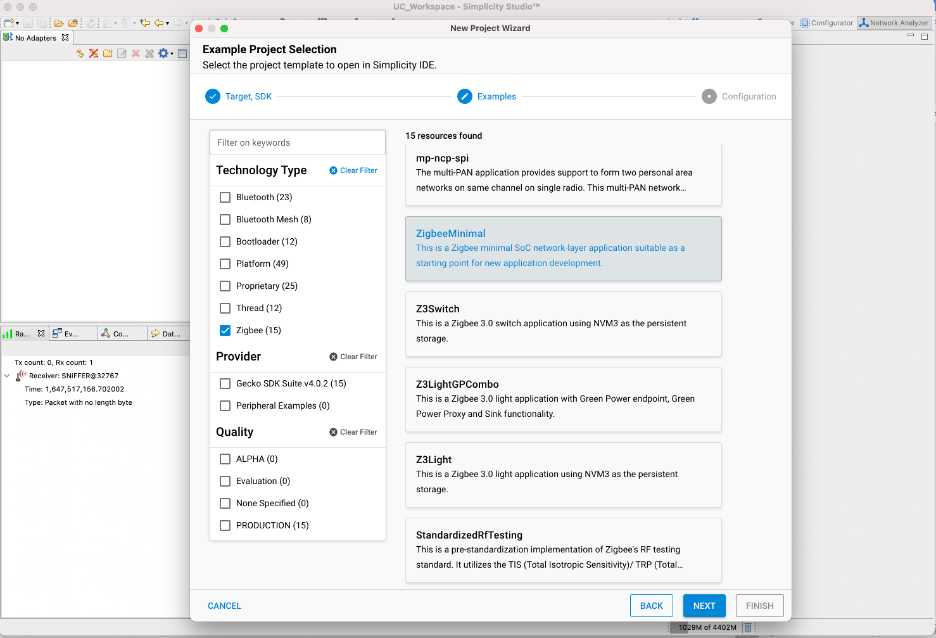
Optionally you can follow the subsequent steps 1. to 3.

1. Go to File > New > Silicon Labs Project Wizard. This will bring up the New Project Wizard. See Figure 2-1 below.



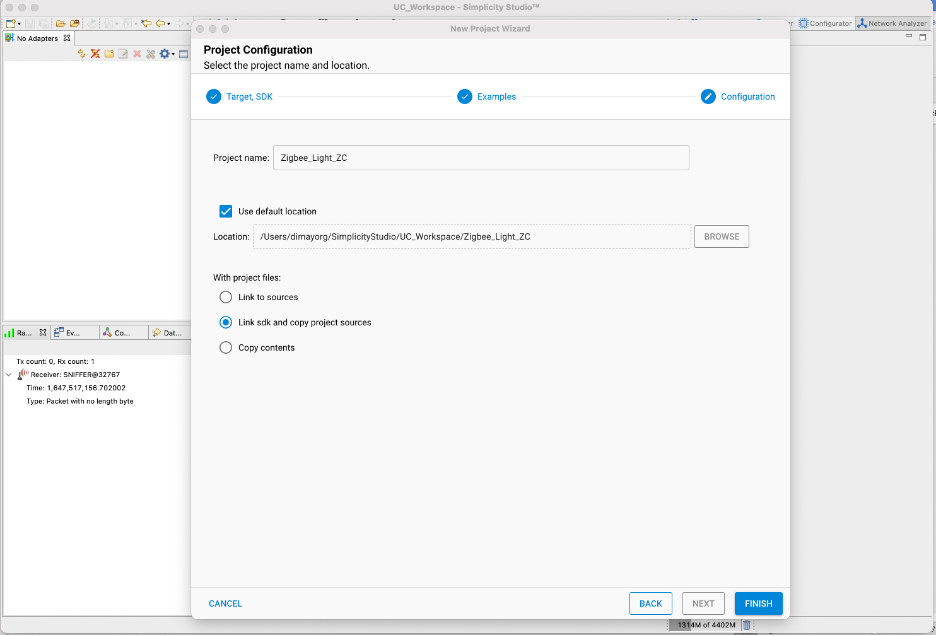
**Figure 2-1 New Project Wizard**

1. Select your Target Board (BRD4162A), Target Device (EFR32MG12P332F1024GL125), and ensure that you have selected GSDK 4.1.0 and GNU ARM v10.3.1 click Next. See Figure 2-2.



**Figure 2-2 Example Project Selection.**

1. Under Technology Type check “Zigbee”, search and select “ZigbeeMinimal”. This is a Zigbee minimal application suitable as a starting point for new application development. Click Next. See Figure 2-3.

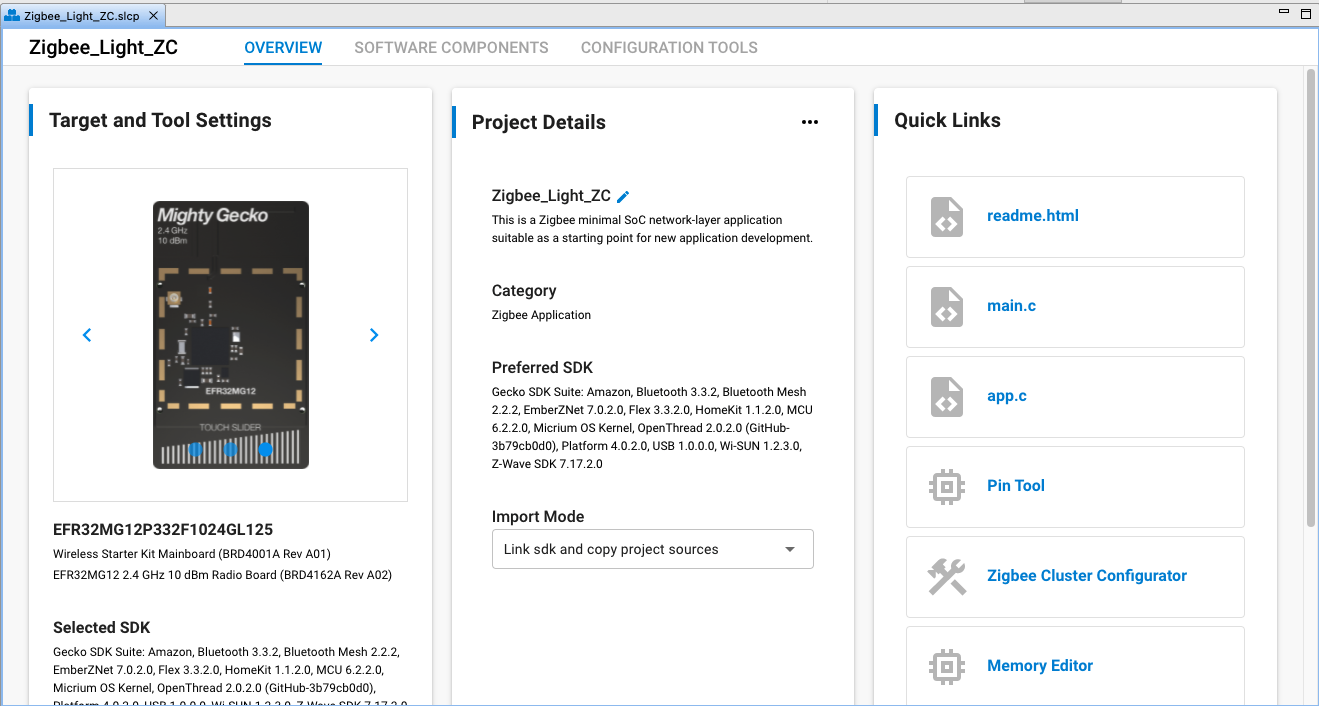


**Figure 2-3 Project Configuration.**

1. Name your project to "Zigbee\_Light\_ZC" and then Click Finish to create the project.

**2.1. Configure the "Zigbee\_Light\_ZC" project.**

Once the project has been created Simplicity Studio should automatically open the Project Configurator (.slcp) which shows an overview of the project as depicted in Fig 2-4. In the overview you can see the Target and SDK Selection, Project Details and the Project Generators.



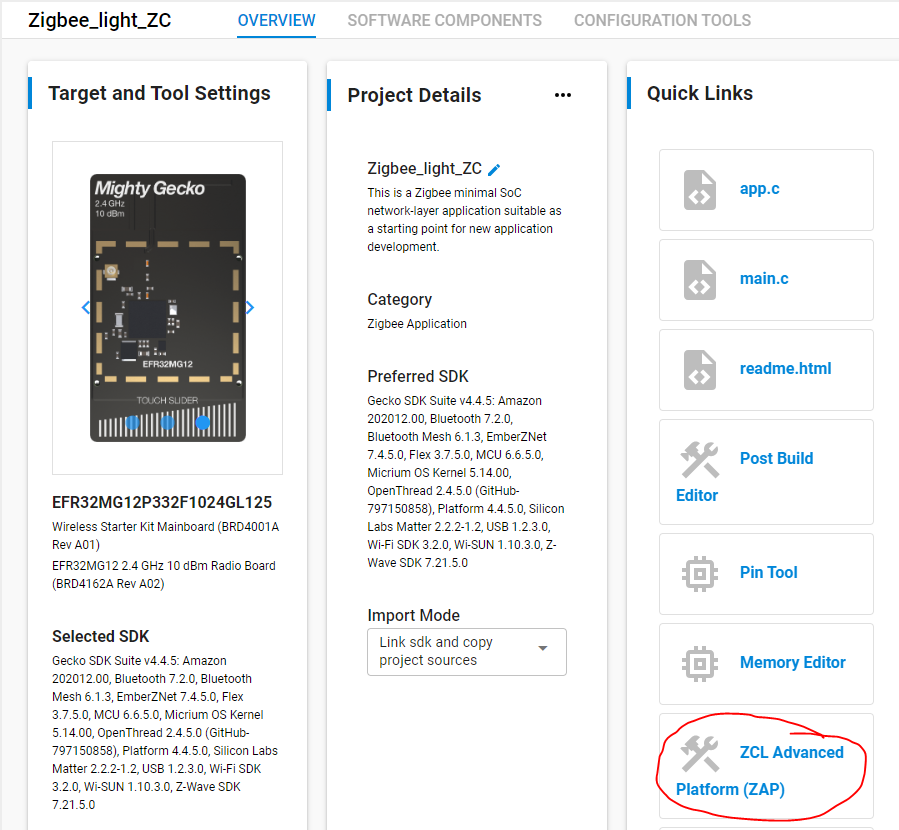
**Figure 2-4 Project Overview**

**Zigbee Cluster Configurator.**

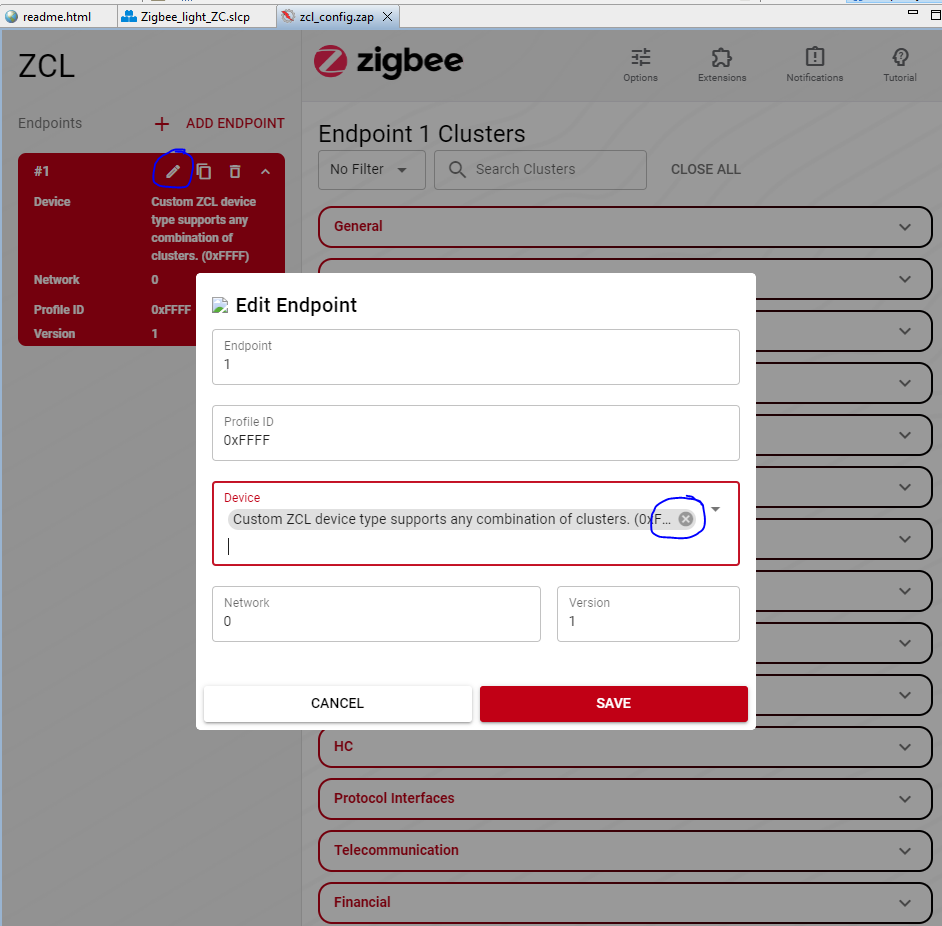
For this tutorial, you must define the device type which will provide the proper clusters and attributes for the Light application. It is important to set the ZCL (Zigbee Cluster Library) configuration for the Light Application as it assists in configuration of the cluster functionality incorporated . In order to change the ZCL configuration, you must open the Zigbee Cluster Configurator in the Configuration Tools tab of the .slcp.

For this purpose

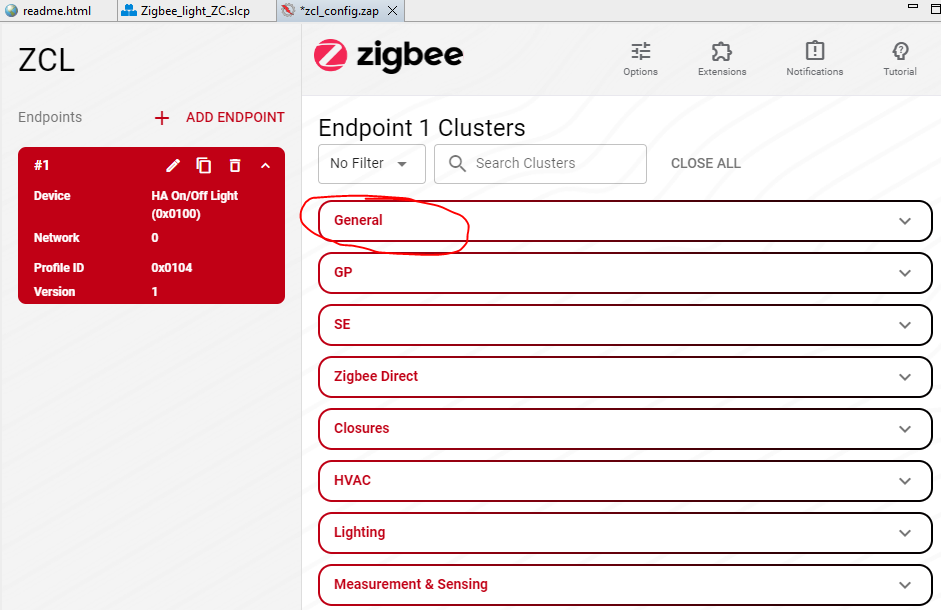
* Click on Overview (see fig. 2-4)
* Choose ZCL Advanced



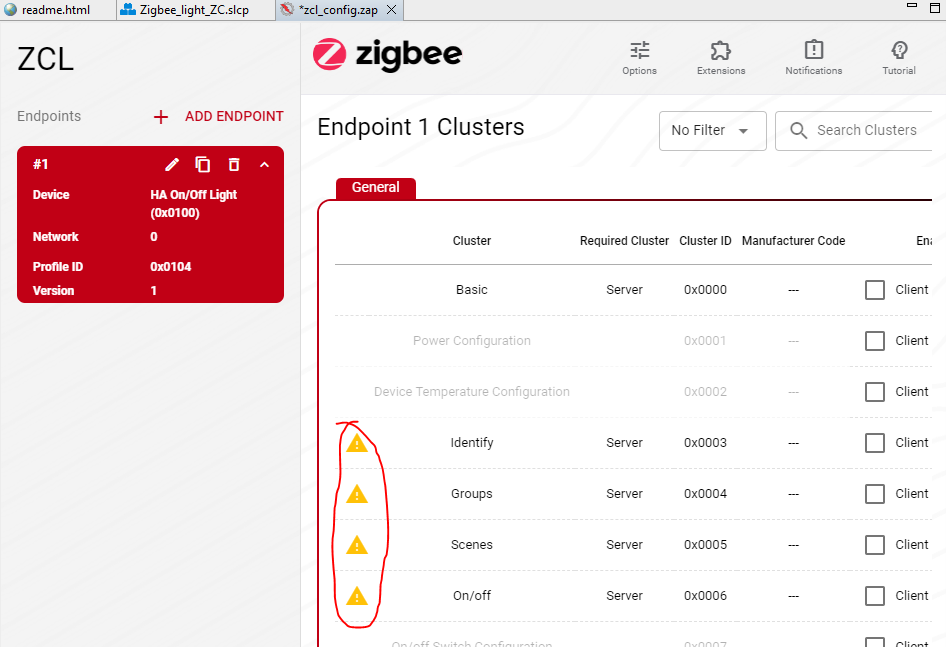
If an endpoint is not created, click on ADD NEW ENDPOINT. Once an endpoint has been added click on edit and select Device: “HA On/Off Light (0x0100)”. By selecting a device and pressing “SAVE” type endpoint is configured according to the Zigbee Specifications.



**Note:** Verify that all software components are installed properly by choosing “General”.



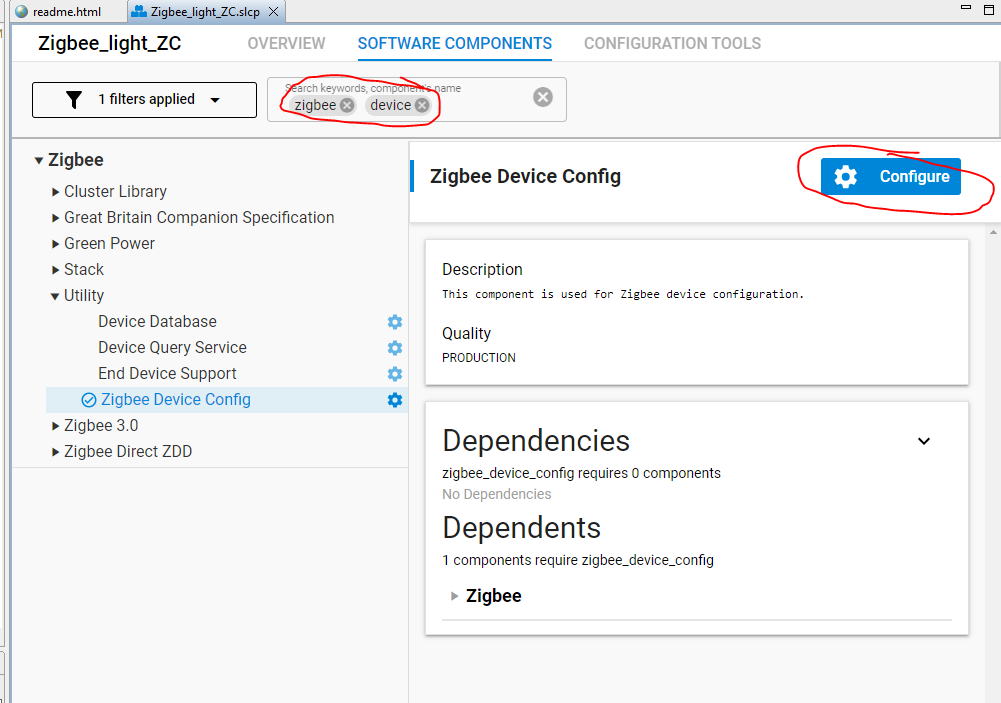
The endpoint clusters with a yellow warning sign implies that the component corresponding to that cluster is not installed in the Software Components tab see Figure 2-5. Click on the warning sign and it will automatically install the component. The Enabled Clusters filter can be used to identify the required clusters easily. Afterwards save configuration by clicking ***Ctrl+S***.



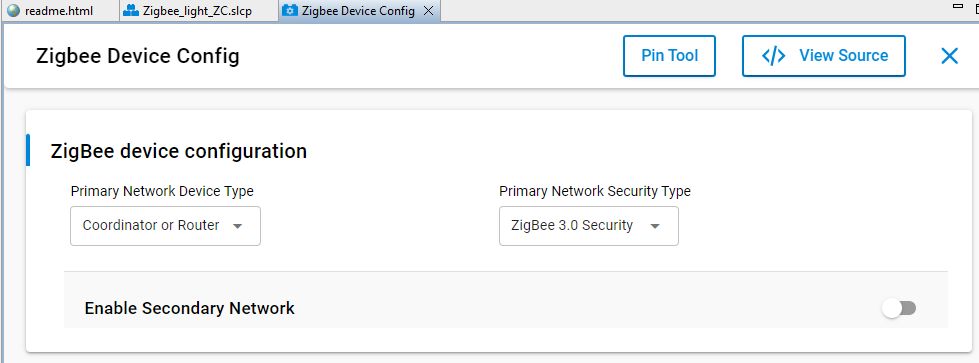
**Figure 2-5 Warning to install missing Components**

Configure next the Zigbee Light as a coordinator:

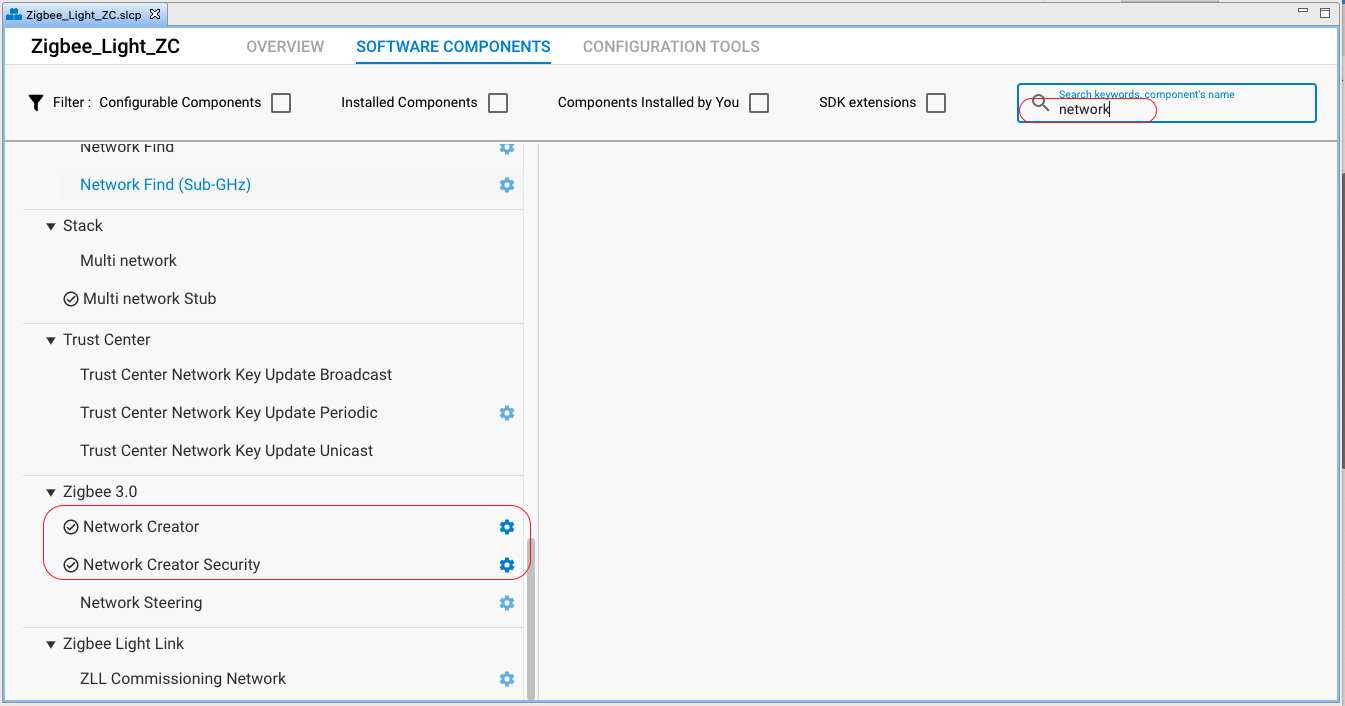
* In the Project Configurator (doubleclick to the .slcp file) click on “Software Components”. This will show the components that can be installed/uninstalled in the project.
* Go to Zigbee > Utility > Zigbee Device Config or search for “zigbee device” and click Configure.



* Change the Primary Network Device Type to “Coordinator or Router”.



* The components mentioned below must be installed or removed to get a device that can operate as a Coordinator. See Figure 2-6 on how to install components in Project Configurator. Please note that the components mentioned below are the minimal requirements to finish the Forming and Joining hands-on, however, it’s not enough for making the “Coordinator/Router” and “Router” device to pass the Z3 certification. For Z3 certification, please refer to the Z3LightSoc and Z3SwitchSoc examples for the necessary Components.



**Figure 2-6 Components**

​ •The **Network Creator** and **Network Creator Security** components implement the network forming and opening functionality, therefore these are required to have for the Coordinator.

​ •The **Network Steering** and **Update TC Link Key** should be removed, since the device doesn't intend to join any network.

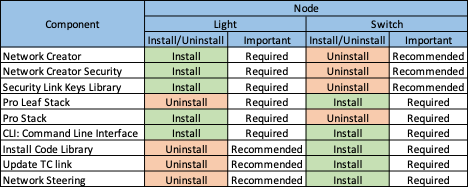
​ •The **Pro Stack** component under Zigbee > Stack > Pro Core includes one of the most complex stack libraries. It contains routing, networking, scanning, neighbor, child-handler and other functionalities. It's mandatory for Coordinator and Router. The sample application uses this component by default.

​ •The **Security Link Keys** component under Zigbee > Stack provides management of APS link keys in the key table. It is used by a trust center (coordinator) to manage link keys of devices in the network, or by non-trust center devices wishing to manage partner link keys. Therefore, it is required to have.

​ •It must be installed the **CLI Global Configuration**

**Note:** Ensure the Zigbee > **Zigbee Core CLI** is also installed in the project. This component should be preinstalled by the Zigbee Minimal Project Configuration.

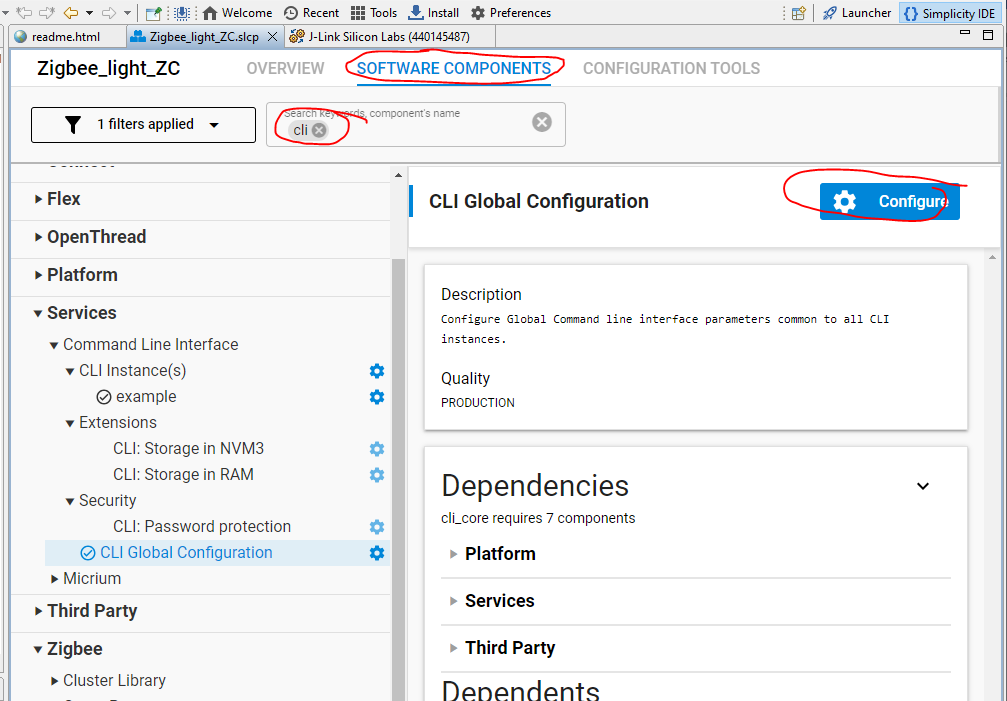
Table 2.1 depicts the affected components on the Light (Coordinator) node:



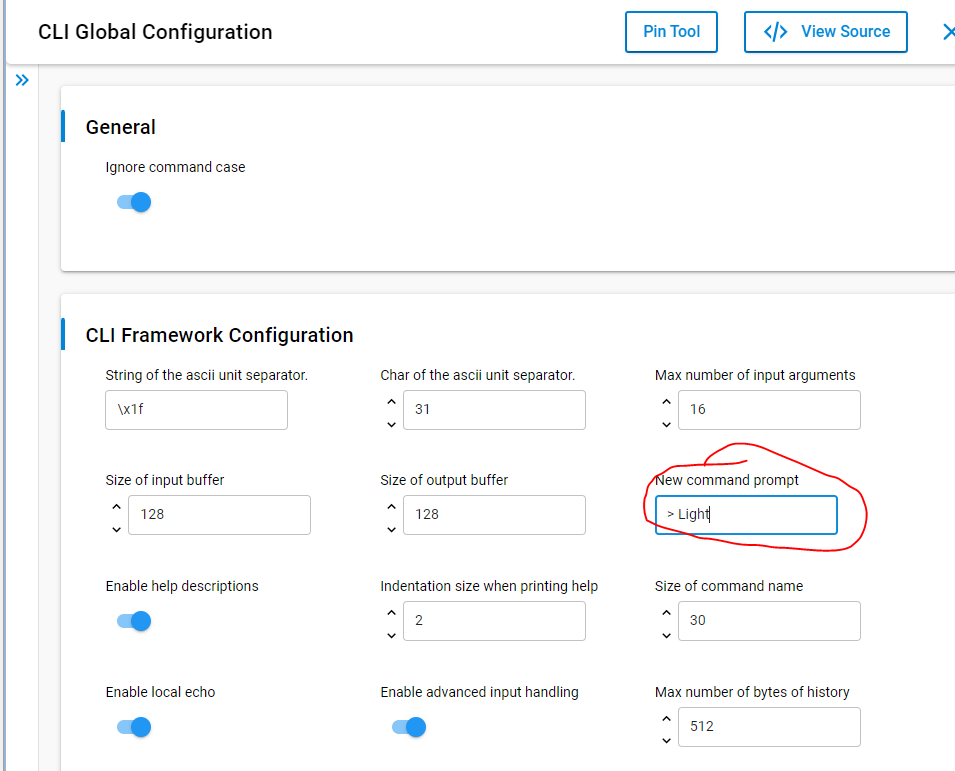
**Table 2-1 Components to Install/Uninstall**

Now we are configuring the prompt oft he serial console:

* Go to “Software Components” and search for **CLI global configuration** and choose “Configure”-Button:

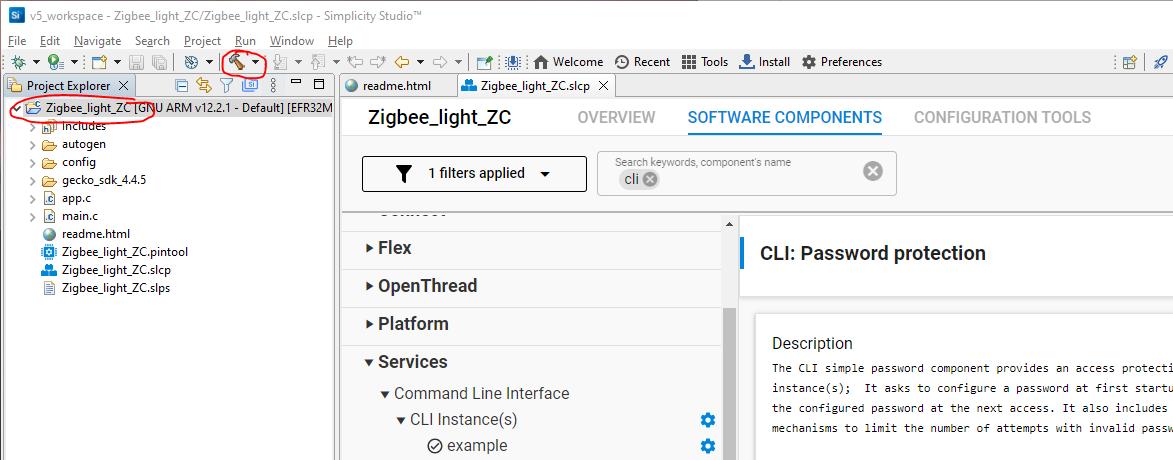


Put the new **command prompt name**



**Figure 3-5 New Command Prompt CLI Core**

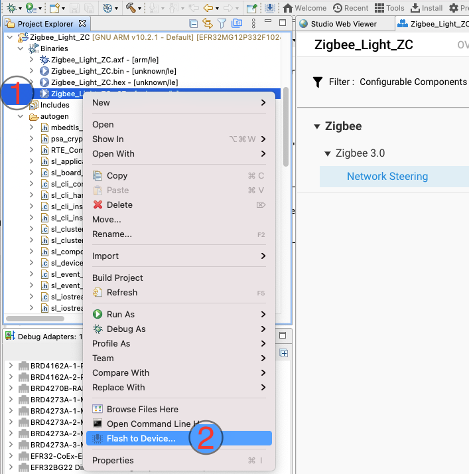
Now press the **Build button**. Upon a successful build, the binary files should appear in the “Binaries” directory.



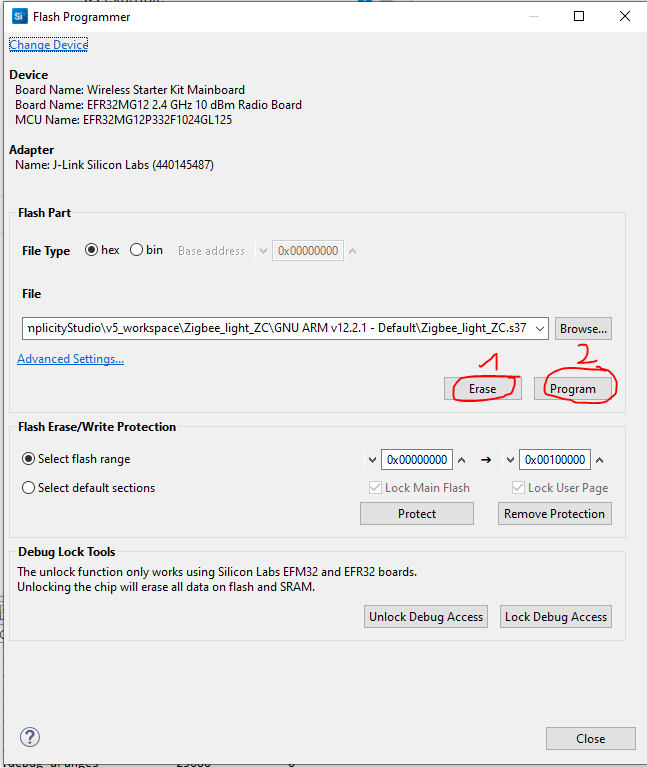
**3. Download and test the Light application**

Let's download the **Zigbee\_Light\_ZC.s37** file to the development kit as shown below. See Figure 3-1 and Figure 3-2. The highlighted "Advanced Settings.." in Figure 3-3 allows the user to decide how to flash the chip. Here, the flash can be merged with a new image (Merge Content), partially (Page Erase) or completely (Full Erase) erase before downloading the file. Keep in mind that neither erase type cleans the bootloader section in EFR32MG12 part, but the Full erase deletes the token region. After the image has been downloaded, it's possible to communicate with the device. For this purpose, open the Launch console, which is a built-in serial port terminal in the Studio.

**Note:** Please execute an "Erase" process before the following steps to avoid any unintended effect by the existing network settings in the device.

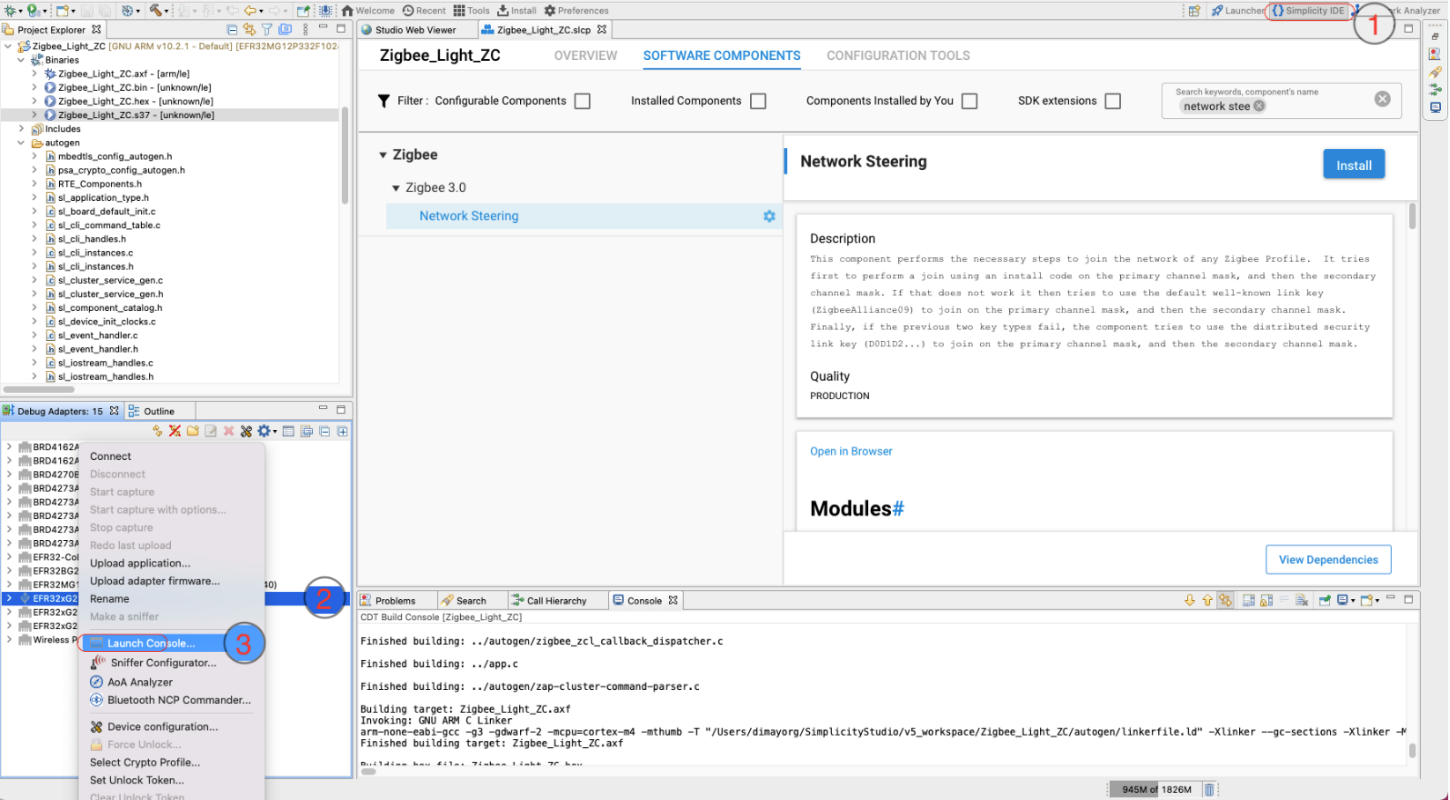


**Figure 3-1 Open Flash Programmer**



**Figure 3-2 Download the Image.**

Go back and open **Serial Console 1**:



**Figure 3-3 Open Serial Console.**

If the serial console is opened, switch to "Serial 1" and press "Enter". See Figure 3-4.



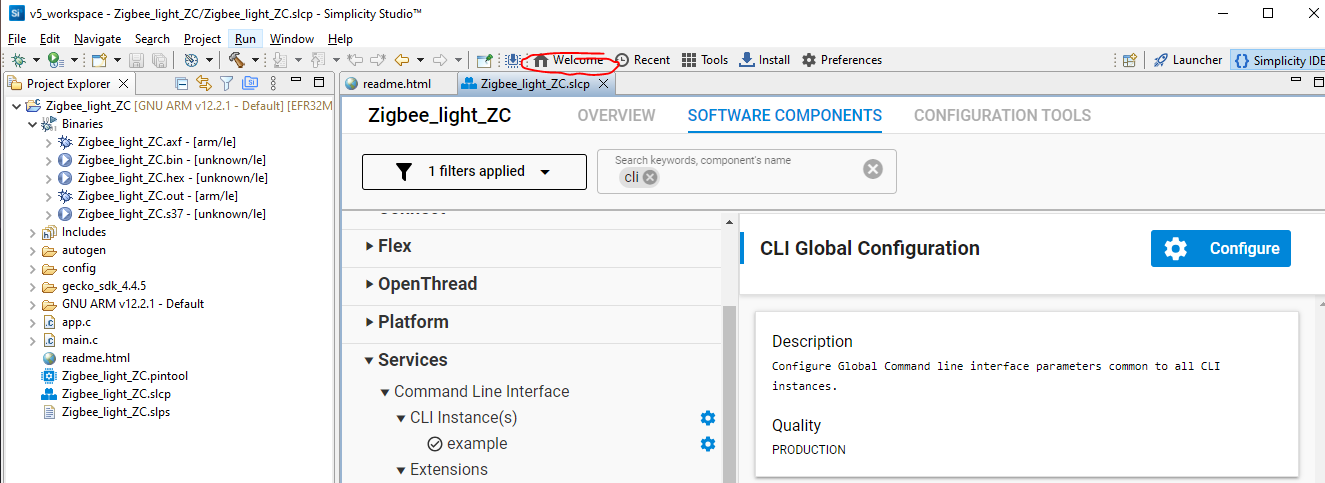
**Figure 3-4 Select Serial 1 tab.**

The "\n\r" characters triggers the project name printing. This basic test shows that the RX and TX of the CLI is working correctly. If the same text is printed, put away the Light application and start creating the Switch. Note: If the project name does not appear in console you can add this in the Services > CLI: CLI Core Component > New Command prompt option as depicted in Figure 3-5.

**4. Create Switch Application**

In this hands-on, the Switch is the device that will join the network created and opened by the Light. However, the procedure to create and form the network is described in the next module. Creating the project and configuration is similar to the Light application. The Switch application is also based on the "ZigBeeMinimal" sample application, therefore:

1. Repeat the step 1-4 of chapter Create Light application, except name the project to "Zigbee\_Switch\_ZR". For this purpose go back to “Welcome” Page:



1. Open the .slcp file of the project.

* Go to Overview-tab, open the Zigbee Cluster Configurator (ZCL Advanced Platform) and choose HA On/Off Switch(0x0000) device template. Install the clusters that have a yellow warning sign. Click on **General** Link. The Enabled Clusters filter can be used to identify the required clusters easily.
* Go to Software Components tab and search for Zigbee Device Config select the Router device type from the dropdown menu.
* Go to Software Components tab and double check the below components are installed:
  + Network Creator is uninstalled
  + Network Creator Security is uninstalled
  + Security Link Keys Libraries are uninstalled
  + Pro Leaf Stack is uninstalled
  + Pro stack (basic and common) are installed
  + CLI Global Configuration is installed (configure the prompt!)
  + ZigBee Core CLI is installed
  + Network Steering is installed.
  + Update TC Link Key is installed.

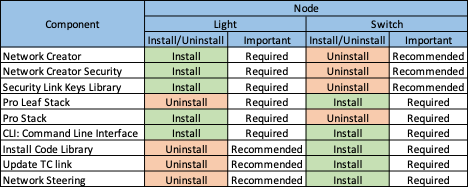
The major difference between the Light application and Switch application is the selection of the network related components. Let's have a closer look at the enabled components.

​ •The **Network Steering** component serves to discover the existing networks on the enabled channels. The device issues a Beacon Request message and listens to the responses. If the Beacon response (from ZC or ZR) is received with "permit association" flag set, the device starts the joining process for the network, otherwise continue the scanning. Please see the Table 5.1 below for the recommended and required components.

​ •The **Update TC Link Key** is used to request a new APS Link Key from the Trust Center. It should be installed since the Light (Trust Center) has the Security Link Keys Library.

​ •The **Install Code Library** provides an initial link key based upon an install code manufacturing token in the device. The key is hashed according to the ZigBee spec. Summarized the above, the following table presents the affected plugins on the Switch (Router) node.

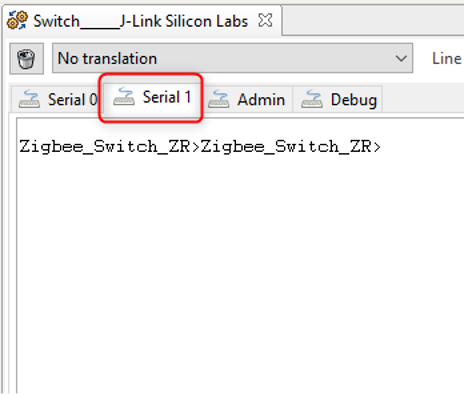
Table 4-1 presents the affected plugins on the Switch (Router) node.



**Table 4-1 Components to check**

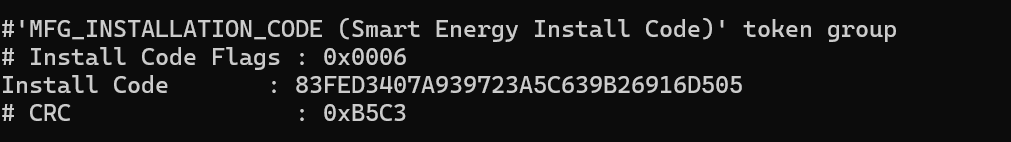
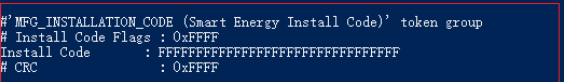
**5. Download and test the Switch application**

Please repeat the steps from chapter **3. Download and test the Light application** and test if the Switch application is working by pressing enter on the Serial 1 tab of the Launch console. See Figure 5-1.



**Figure 5-1 CLI Testing**

Before going to next step check if Install code is already derived on the device by running this command by only connecting switch device on command prompt :

C:\SiliconLabs\SimplicityStudio\v5\developer\adapter\_packs\commander\commander.exe tokendump --tokengroup znet  
  
If install code is already derived we can see the output as shown below:  
  
  
Now we can directly skip to 6.2 Form centralized network on Light (Coordinator) device section and follow the steps.  
  
If Install Code is not derived we get output as shown below:  
  
  
  
Then we have to follow the next steps given and run the batch file and check whether the install code is derived after running the batch file.

**6. Establish Connection between Light and Switch with an install code-derived link key**

This chapter presents how to form and join a network. The communication between devices will be captured by the Network Analyzer tool. The install code will be programmed in this section. An install code is used to create a preconfigured link key. The install code is transformed into a link key by using the AES-MMO hash algorithm, and the derived Zigbee link key will be known only by the Trust Center (Light) and the joining device (Switch). So, the Trust Center can use that key to securely transport the Zigbee network key to the device. Once the device has the network key, it can communicate at the network layer to the Zigbee network.

**6.1 Programming the Install Code to Switch (Router) Device**

For programming the install code into the Switch device, you need to create a text file with the value of the install code, and then write the install code into the manufacturing area of the Switch node by using the Simplicity Commander. To save your time on this hands-on, we have prepared a batch file as below that can finish the install code programming automatically. Create a batch file, as seen below, open it with any text editor, copy and paste the content below to it, save and execute it for programming the install code. Please ensure only the switch device is connected to the PC, before running the script.

Create a BAT-file in the project directory (e.g. install\_code.bat) and run it from the command line.

@echo off

:: THIS FILE IS USED FOR PROGRAMMING INSTALLATION CODE AUTOMATICALLY.

:: use PATH\_SCMD env var to override default path for Simplicity Commander

if "%PATH\_SCMD%"=="" (

set COMMANDER="C:\SiliconLabs\SimplicityStudio\v5\developer\adapter\_packs\commander\commander.exe"

) else (

set COMMANDER=%PATH\_SCMD%\commander.exe

)

:: default file extension of GCC and IAR

set DEFAULT\_INSTALL\_CODE="83FED3407A939723A5C639B26916D505"

:: change the working dir to the dir of the batch file, which should be in the project root

cd %~dp0

if not exist "%COMMANDER%" (

echo Error: Simplicity Commander not found at '%COMMANDER%'

echo Use PATH\_SCMD env var to override default path for Simplicity Commander.

pause

goto:eof

)

echo \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

echo Program the default installation code to the specified device

echo 1. Erase the Installation Code if existing

echo 2. Program the Installation Code into the Manufacturing Area of the specified Device

echo 3. Check the Stored Installation Code

echo \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

echo.

%COMMANDER% flash --tokengroup znet --token "Install Code: !ERASE!"

echo.

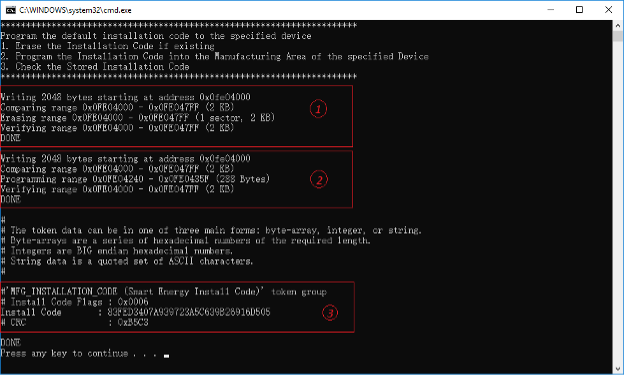
%COMMANDER% flash --tokengroup znet --token "Install Code:%DEFAULT\_INSTALL\_CODE%"

echo.

%COMMANDER% tokendump --tokengroup znet --token TOKEN\_MFG\_INSTALLATION\_CODE

pause

Below is the result of executing the batch file:



**Figure 6-1 Programming the Install Code**

**NOTE**: Every time we flash into the board try leaving the network using network leave command and please make sure you start running commands with option install-code <link key table index> {<Joining Node's EUI64>} {<16-byte install code + 2-byte CRC>} to form a new network irrespective of with and without security.

**6.2 Form Centralized Network without Security**

To form the centralized network without security run these commands   
  
Make sure you run the command :   
  
option install-code <link key table index> {<Joining Node's EUI64>} {<16-byte install code + 2-byte CRC>}  
  
On Light(Coordinator):

1. plugin network-creator form 1 0x1234 3 12
2. plugin network-creator-security open-network

On Switch(Router):

1. plugin network-steering start 0

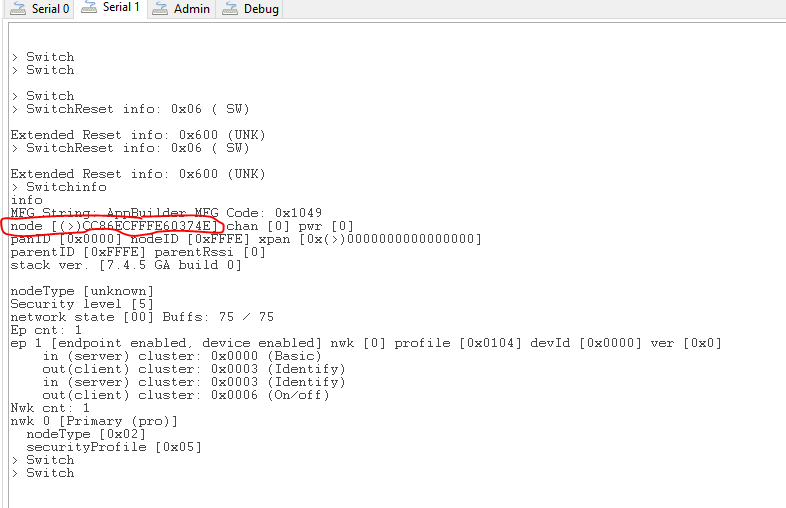
Source: https://community.silabs.com/s/article/how-to-form-zigbee-3-0-network-with-emberznet-stack-x

**6.3 Form centralized network on Light (Coordinator) device**

**6.2.1 Derive a link key from the install code**

Connect the Light(Coordinator) to USB again. Now we derive the link code for the coordinator from the client:

* Go to serial console(serial1) of the switch(router)
* Enter command *info*
* Copy the physical address



To derive a link key from the install code and store that into the link key table on the Light, which acts as the Trust Center for the centralized network, enter the command below in the console-window of the coordinator: option install-code {<Joining Node's EUI64>} {<16-byte install code + 2-byte CRC>} For example (replace red text in the example with the physical address of the client from above):

option install-code <link key table index> {<Joining Node's EUI64>} {<16-byte install code + 2-byte CRC>}

For example:

option install-code 0 { 90 FD 9F FF FE 19 B1 FC } {83 FE D3 40 7A 93 97 23 A5 C6 39 B2 69 16 D5 05 C3 B5}

* The first argument is the link key table index.
* The next argument is the EUI64 of the joining node (in this example, it's the Switch node).
* The last argument is the install code with the 2-byte CRC appended at the end. You can calculate the CRC yourself, or you can simply find out from the output of the batch file execution which has the command $ commander tokendump --tokengroup znet inside:



**Figure 6-5 Get the CRC Install Code**

The CRC is displayed just below the install code and is printed in little endian format. Reverse the bytes to big endian before using as an argument with the option install-code CLI.

To see if the link key is added successfully to the Transient Key Table, enter the keys print command on the CLI of the Light node. The printout shows the Transient Key table that contains the derived key. Additionally, once the Switch application is connected to the network it shows the Trust Center Link key table and the Network key.



**Figure 6-6 Check the Link Key**

As shown above, the derived link key is:

''' 66 B6 90 09 81 E1 EE 3C A4 20 6B 6B 86 1C 02 BB '''

**6.2.2 Form Centralized Network with Security**

On Light node, use the command below to form a centralized network with Zigbee 3.0 security.

plugin network-creator start 1

After that, please check the Pan ID of the network, it will be used to identify the network.

network id

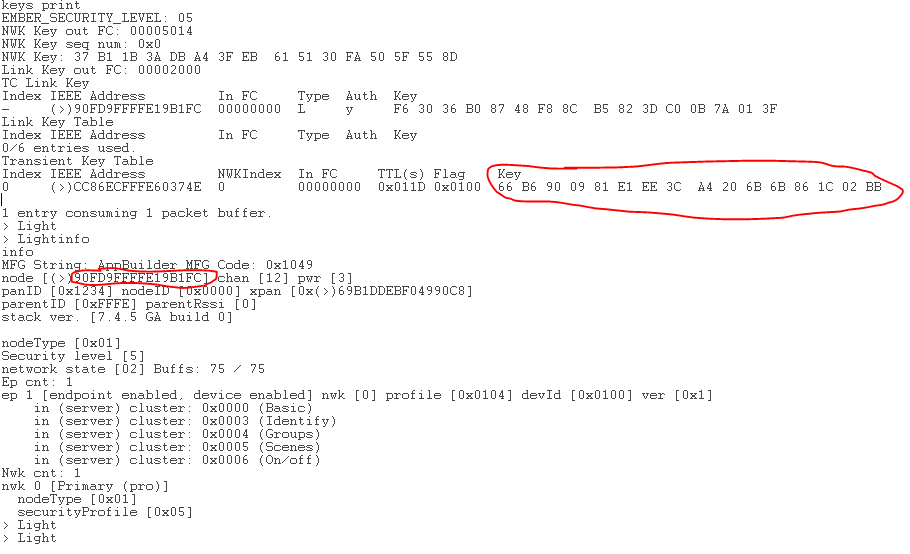
Now set the transient link key (the same link key that you derived from the install code) on the Trust Center and open the network on the Light Application to join with the joining device's EUI64:

plugin network-creator-security open-with-key {eui64} {linkkey}

For Our Example:

plugin network-creator-security open-with-key {CC 86 EC FF FE 60 37 4E} { 66 B6 90 09 81 E1 EE 3C A4 20 6B 6B 86 1C 02 BB}

Note: Red Text in line 3 is the physical address of the Light which we get this by running network id command as shown below.



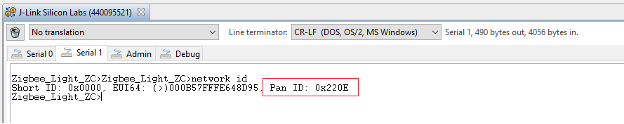
**6.4 Join the network on Switch (Router) device**

On the Switch node, enter this CLI to use the Network Steering plugin to join the network:

plugin network-steering start 0

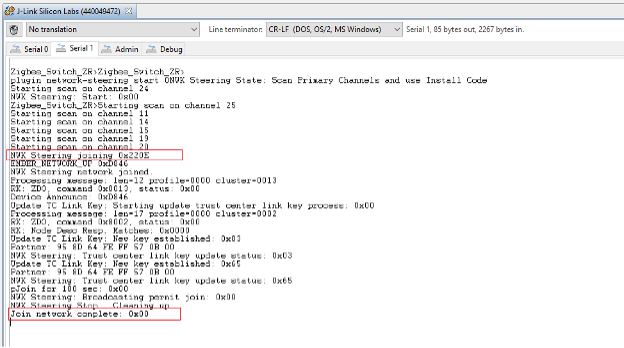
After that, please check the Pan ID of the network, it will be used to identify the network.

network id



**Figure 6-7 Check the Pan ID**

And the serial console will output similar as below to indicate that the Switch node has joined the network 0x220E successfully.



**Figure 6-8 Join the Network**

**Conclusion**

In this hands-on, you learned how to create your Zigbee application projects starting with the ZigbeeMinimal example. And how to configure your application as a different type of Zigbee node (coordinator, Router, etc.), how to enable/disable different Components for different functionality to meet your needs, and how to form a centralized network and join this network. This hands-on also demonstrates how to evaluate data being transmitted in the Zigbee network using the Network Analyzer tool.