Instructions for Zigbee Light & Switch Tutorial
https://www.silabs.com/community/wireless/zigbee-and-thread/knowledge-base.entry.html/2018/11/08/zigbee_3_0_tutorial-FUTT

1. Connect your WSTKs to your machine and connect your Thunderboard Sense 2s
    a. Set your WSTKs to Debug Mode OUT
2. Build your bootloader
    a. From Launcher or the Simplicity IDE view, select File > New > Project.
    b. Select **Silicon Labs AppBuilder Project** and press Next.
    c. Select **Gecko Bootloader** and press Next.
    d. Select "**SPI Flash Storage Bootloader (single image)**" and press Next.
    e. Give your Project a name, such as "TB-app-BL" and hit Next.
    f. Specify the board as "**Thunderboard Sense 2 (BRD4166A)**" and select your toolchain (GNU ARM 7.2.1 or IAR ARM 8.30.1) then press Finish.
    g. Once your project comes up, hit **Generate**.
    h. Once your project has generated compile with the **Hammer icon**.
    i. Use commander to flash your bootloaders to your Thunderboards
3. Create your projects
    a. From Launcher or the Simplicity IDE view, select File > New > Project.
    b. Select **Silicon Labs AppBuilder Project** and press Next.
    c. Select "**ZCL Application Framework V2**" and press Next.
    d. Pick **EmberZNet 6.4.0.0 GA SOC** and press Next.
    e. Check the box that says, "**Start with a blank application**" and press Next.
    f. Give your project a name, such as "MyLight" or "MySwitch", and press Next.
    g. Specify the board as z**Thunderboard Sense 2 (BRD4166A)**" and select your toolchain (GNU ARM 7.2.1 or IAR ARM 8.30.1) then press Finish.

    Do this for both the light and switch.
4. Configure your light
    a. On the ZCL Clusters tab set your device for a **single endpoint** and the ZCL device type as **LO On/Off Light**.
    b. On the Znet Stack tab set your ZigBee Device Type as **Coordinator or Router** and your Security Type as **Zigbee 3.0 Security**.
    c. On the Printing and CLI tab, check both boxes for **On/Off Cluster debugging** printing (Compiled in and Enabled at Startup)
    d. On the Plugins tab add these plugins to your project: **Network Creator**, **Network Creator Security**, **Find and Bind Target**, and **Security Link Keys Library**
    e. In the Serial Plugin make sure both SERIAL and USART0 are checked as enabled and that USART0's flow control is set to Xon-Xoff.
    f. Save, generate and build your project.
5. Configure your switch
    a. On the ZCL Clusters tab set your device for a **single endpoint** and the ZCL device type as **LO On/Off Light Switch**.
    b. On the Znet Stack tab set your ZigBee Device Type as **End Device** and your Security Type as **Zigbee 3.0 Security**.
    c. On the Printing and CLI tab, check both boxes for **On/Off Cluster debugging** printing (Compiled in and Enabled at Startup)

    d. On the Plugins tab add these plugins to your project: **Find and Bind Initiator** and **Button Interface**.
    e. Switch the Zigbee PRO Stack Library Plugin to the Zigbee Pro Leaf Library
    f. In the Binding Table Library plugin expand the Binding Table to 6 entries
    g. In the Serial Plugin make sure both SERIAL and USART0 are checked as enabled and that USART0's flow control is set to Xon-Xoff.
    h. Save, generate and build your project.
6. Flash your light and switch apps to their respective boards.
7. Right click on your two boards and select "**Launch Console**." Use the serial 1 console to verify that both applications are running.
8. Add your network key to Simplicity Studio
    a. Open Studio Preferences
    b. Go to Network Analyzer > Decoding
    c. Create a New Security Key
    d. Enter the key: **FE 2A 47 BE 40 CC AE B2 DC 54 0E 0A EA DB 82 A1**
    e. Press Apply then press Save
9. Right click on your boards and start a capture, this will let you see the network traffic for the rest of this tutorial.
10. Start your network on the light
    a. plugin network-creator start 1
11. Flash your install code to the switch
    a. commander flash --tokengroup znet --tokenfile install.txt --device efr32mg12p
12. Open the light for joining
    a. plugin network-creator-security open-with-key {00 0B 57 FF FE DE A4 C3} {FE 2A 47 BE 40 CC AE B2 DC 54 0E 0A EA DB 82 A1}
13. Join the switch to the light
    a. Plugin network-steering start 0
14. Using the CLI verify that you can send and receive commands from the swtich to the light
    a. On the switch:
    zcl on-off toggle
    send 0 1 1
    b. Verify that your light receives this message.
    c. Watch the traffic via Network Analyzer
15. Bind your light and switch
    a. On the light:
    plugin find-and-bind target 1
    b. On the swtich
    plugin find-and-bind initiator 1
    c. Retest your toggle command with the binding:
    zcl on-off toggle
    bsend 1
16. Configure your Light to respond to on-off commands
    a. On the Callbacks tab, enable these callbacks
        i. Main Start
        ii. Server Attribute Changed
    b. Check the box for "Generate project-specific callbacks file" at the bottom of the Callbacks tab is check.
    c. Launch Hardware Configurator and setup the following pins:

i. PJ14 – RGB_LED_ENABLE
ii. PD11 (M10) – LEDS_RED
iii. PD12 (N11) – LEDS_GREEN
iv. PD13 (M11) – LEDS_BLUE
v. PI0 (G13) – LED0
vi. PI1 (G12) – LED1
vii. PI2 (G11) – LED2
viii. PI3 (F13) – LED3

d. Save your HWConf File
e. Regenerate
f. Edit your MyLight_callbacks.c file (see the included code samples)
    i. Edit emberAfMainStartCallback() to configure your lights at startup
    ii. Edit emberAfOnOffClusterServerAttributeChangedCallback() to turn your lights on and off as your attribute changes
g. Recompile your project and flash it to your light Thunderboard.
h. Use the switch CLI to test that the Light LEDs turn on and off.

17. Configure your switch to respond to button presses
    a. On the Callbacks tab enable the Button0 Pressed Short callback
    b. Check the box for “Generate project-specific callbacks file” at the bottom of the Callbacks tab is check.
    c. Regenerate the switch
    d. Edit your MySwitch_callbacks.c file (see the included code samples)
        i. Edit emberAfPluginButtonInterfaceButton0PressedShortCallback() to send a ZCL toggle message when pressed.
    e. Recompile your project and flash it to your switch Thunderboard.
    f. Test that the button presses work.