



# OTA Upgrading Worksheet

---

In this worksheet we provide a step-by-step guide to create, build and run Z3GatewayHost sample and ZigbeeMinimal applications based on EmberZNet Stack 6.6.4. If you use a later release in the future, most of the instructions should still apply, 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 v4 development environment, and the Wireless Start Kit (WSTK) with EFR32MG modules. We assume that you have a WSTK and the following software requirements:

- Simplicity Studio 4
- EmberZNet 6.6.4
- GCC 7.2

## KEY FEATURES

- Step-by-step guide to creating, building and running ZigBee 3.0 applications based on EmberZNet 6.6.4
- Use Simplicity Studio v4 as the development tool
- ZigBee end device polling
- Zigbee end device keepalive and aging
- Zigbee end device rejoin

## 1 Pre-requisites

Make sure you have installed the EmberZnet 6.6.4 SDK and GCC toolchain on your PC.

### 1.1 Check EmberZnet SDK

1. Launch Simplicity Studio v4.
2. “Windows”→”Preference”→”Simplicity Studio”→”SDKs”, make sure “EmberZnet 6.6.4” is installed

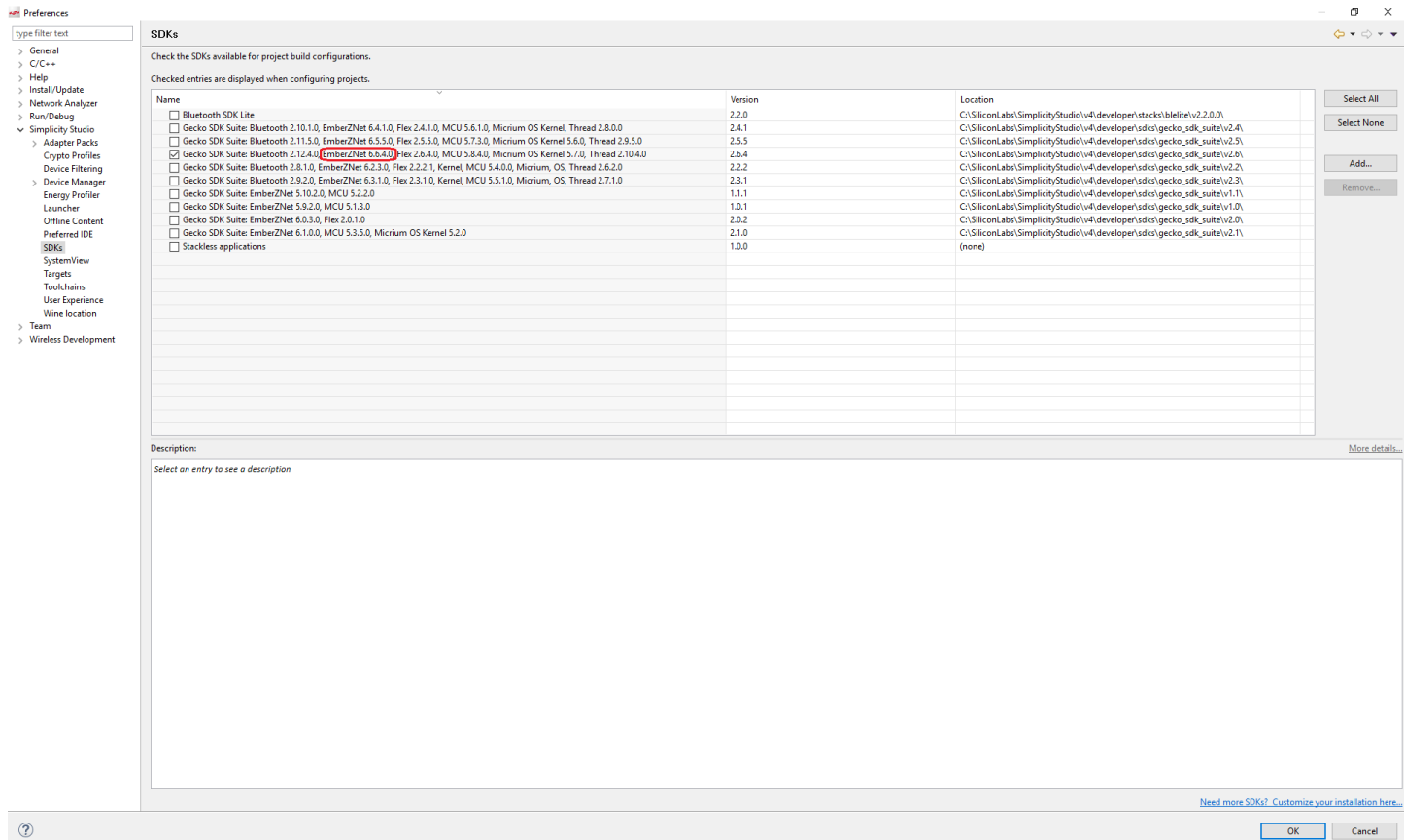


Figure 1 Check SDK in Simplicity Studio

### 1.2 Check Toolchains

1. Launch Simplicity Studio v4.
2. “Windows”→”Preference”→”Simplicity Studio”→”Toolchains”, make sure GCC toolchain is installed.

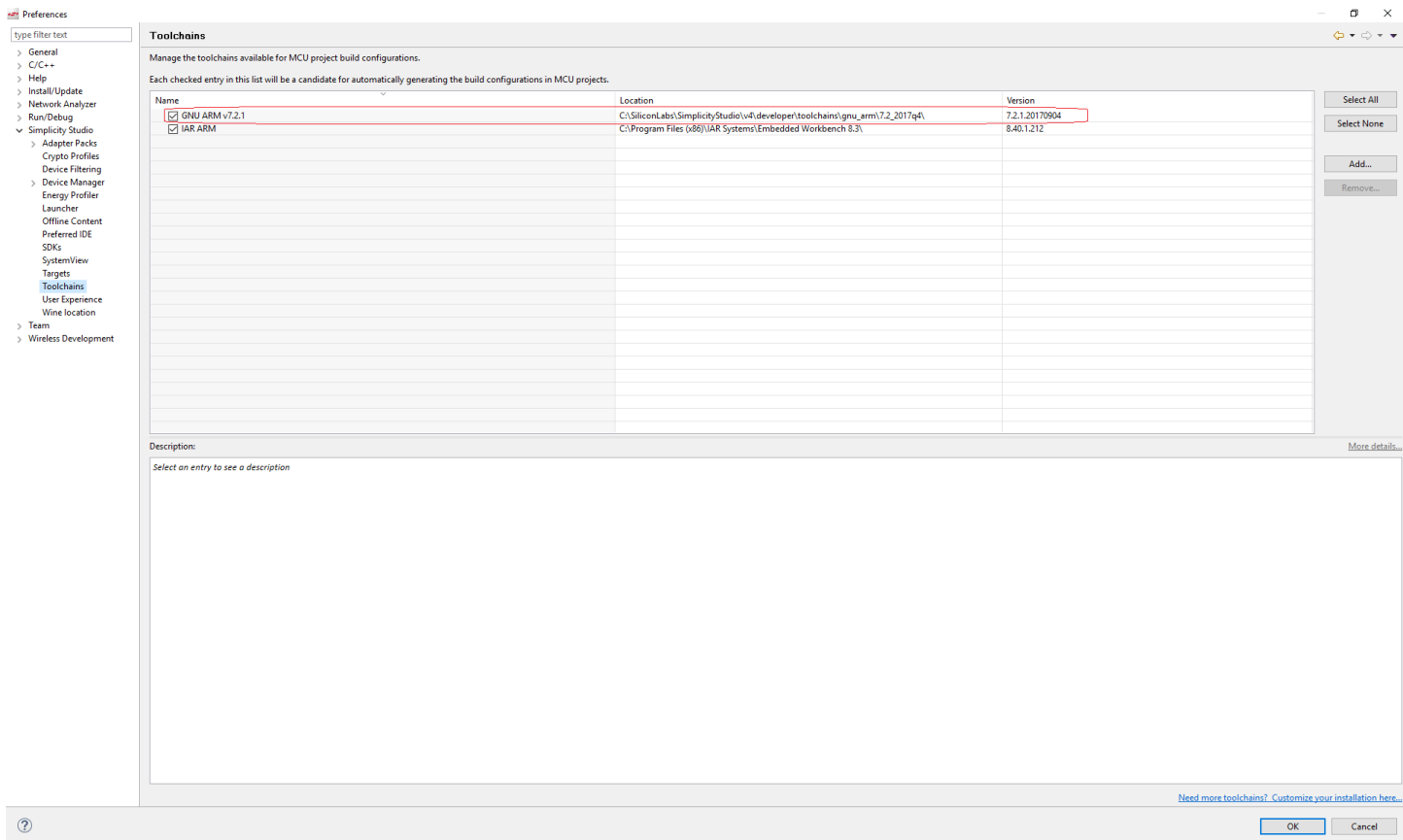


Figure 2 Check Toolchain in Simplicity Studio

### 1.3 Install the software tools:

#### 1. Cygwin;

Extract the package cygwinx86.rar and copy it to the root directory of disk C (or any other disk partition).

Double click “cygwin\_here.reg” then select a folder and right click, check if there is a menu to start Cygwin in this folder.

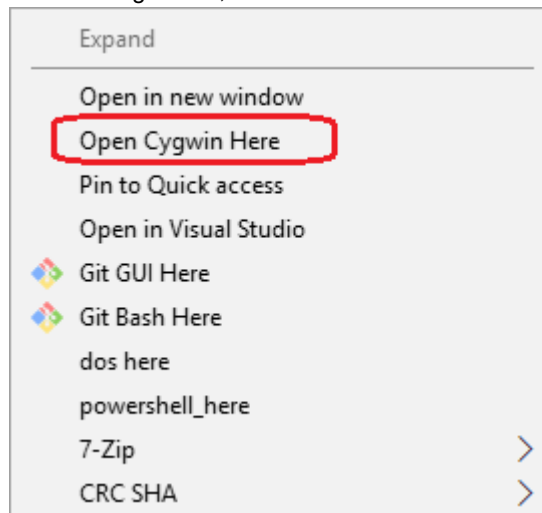
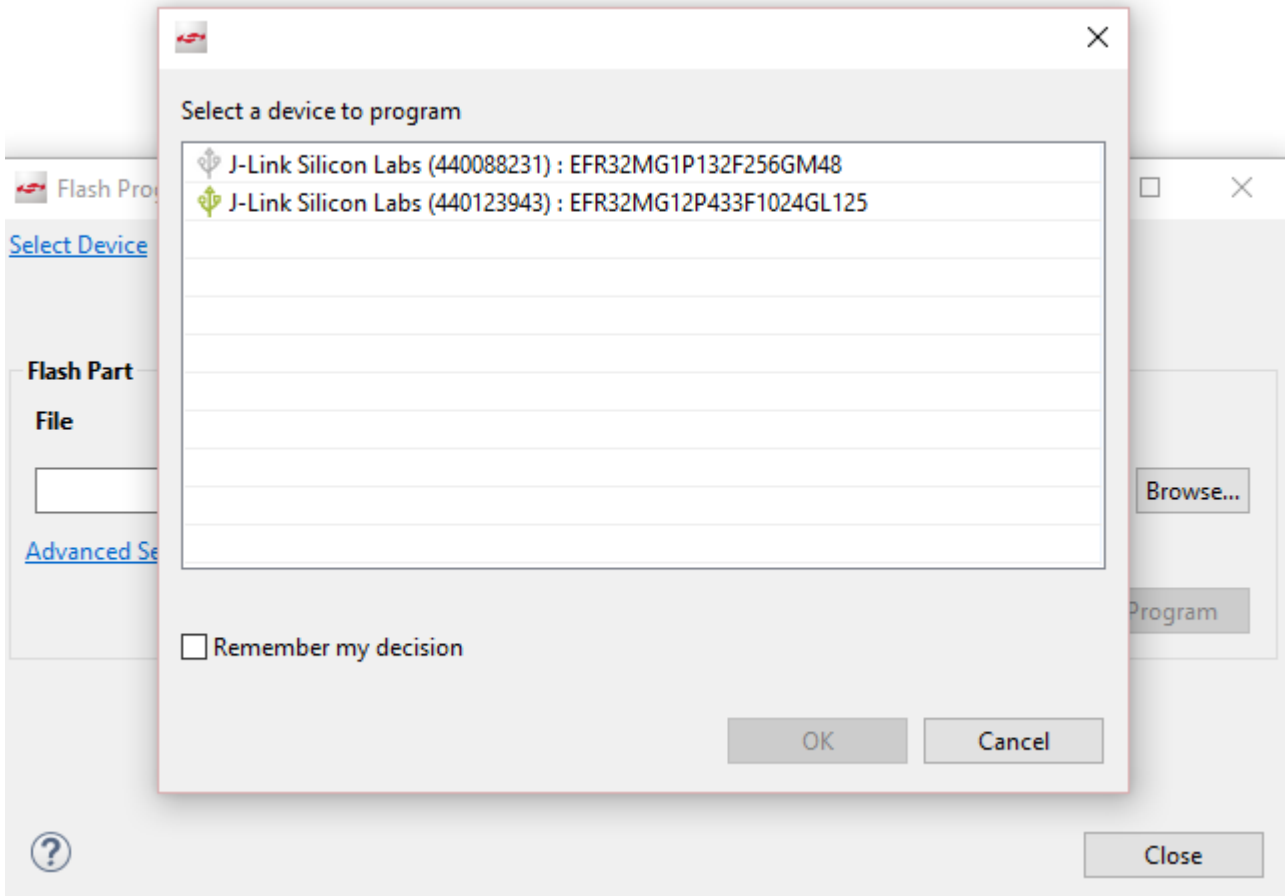


Figure 3 Right-click menu to start cygwin

### 1.4 How to flash the program

#### 1. Start Simplicity Studio, then connect your device to PC;

2. In the menu bar, find the icon  for “Flash Programmer”, press it;
3. In the popup window, select the device;



**Figure 4 Select device**

4. Then in the next window, click “browse” to select the generated image ( .s37 or .hex) of your project, press “Program”. You can also press “erase” if needed. Normally you just need to erase the device once before when you start the hands-on. The generated image is in the binary folder of your project. You can select it in “Project Explorer”.

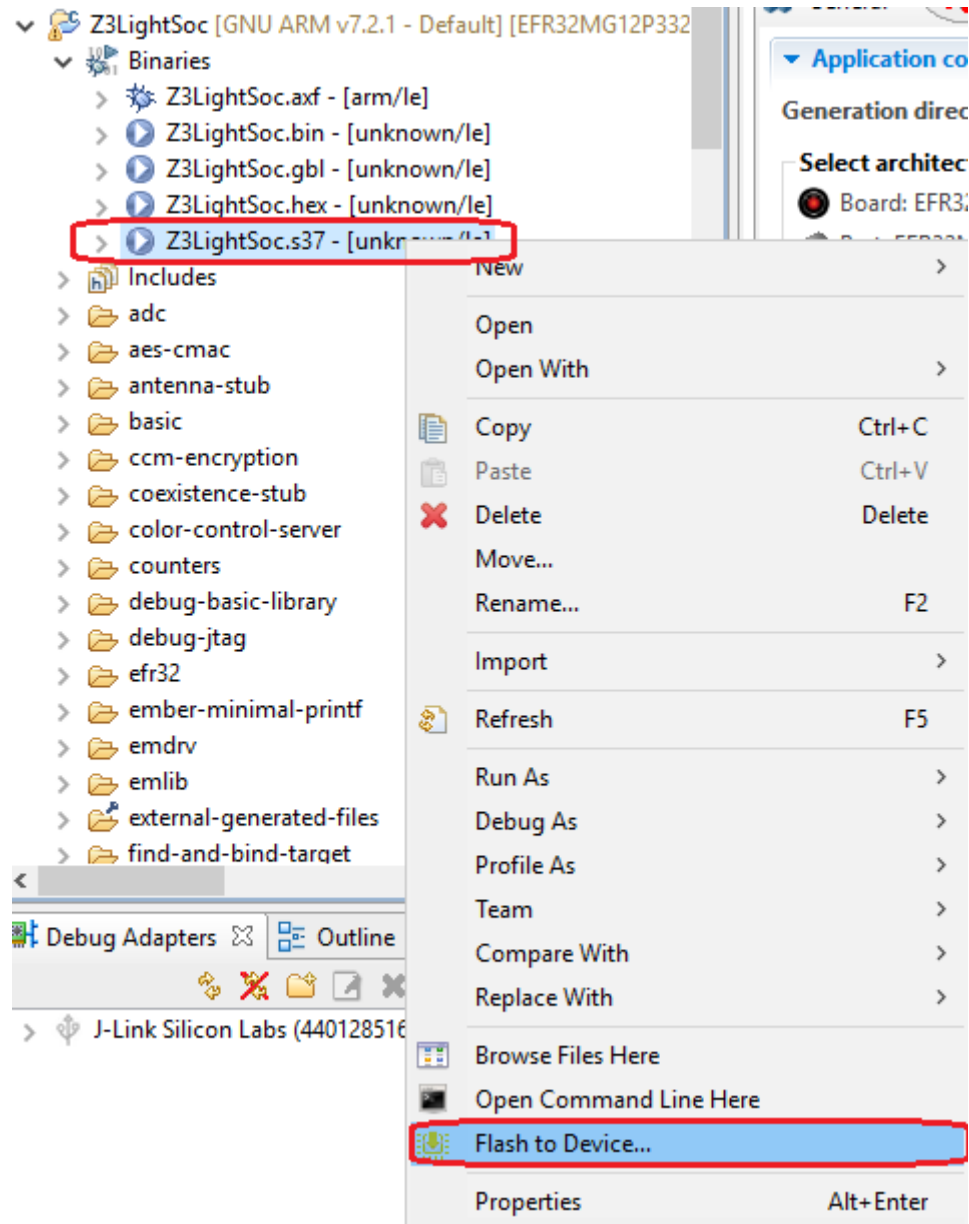


Figure 5 Select image

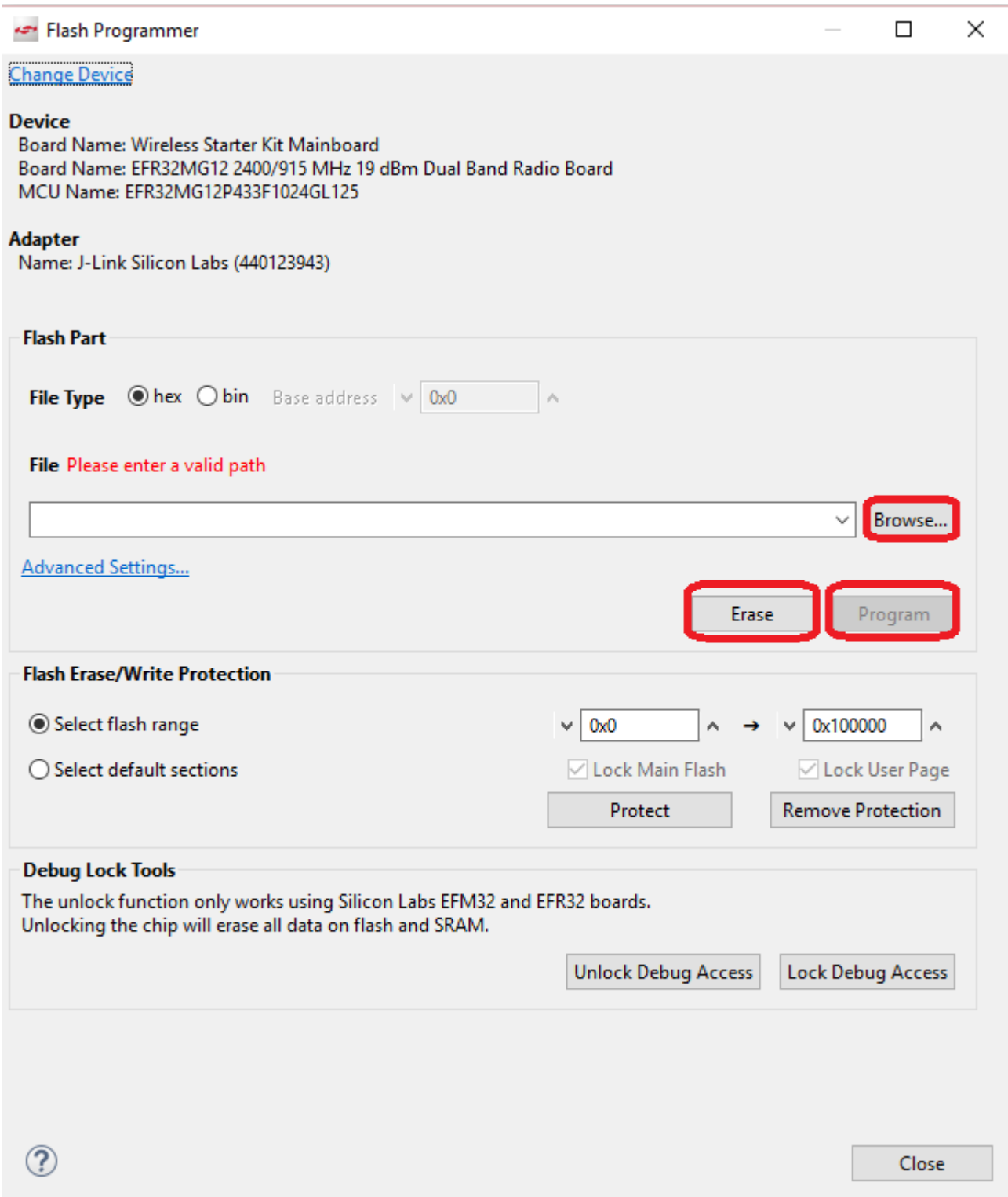
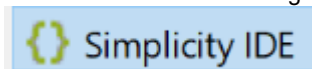


Figure 6 Flash application

### 1.5 How to open console

Simplicity Studio has integrated a console so that it's convenient to debug through console. To use the console, you need:

1. Change to "Simplicity IDE" perspective:
2. Select your adapter in the "Debug Adapters" window, right click and select "connect";



3. Select your adapter in the “Debug Adapters” window, right click and select “Launch console”;

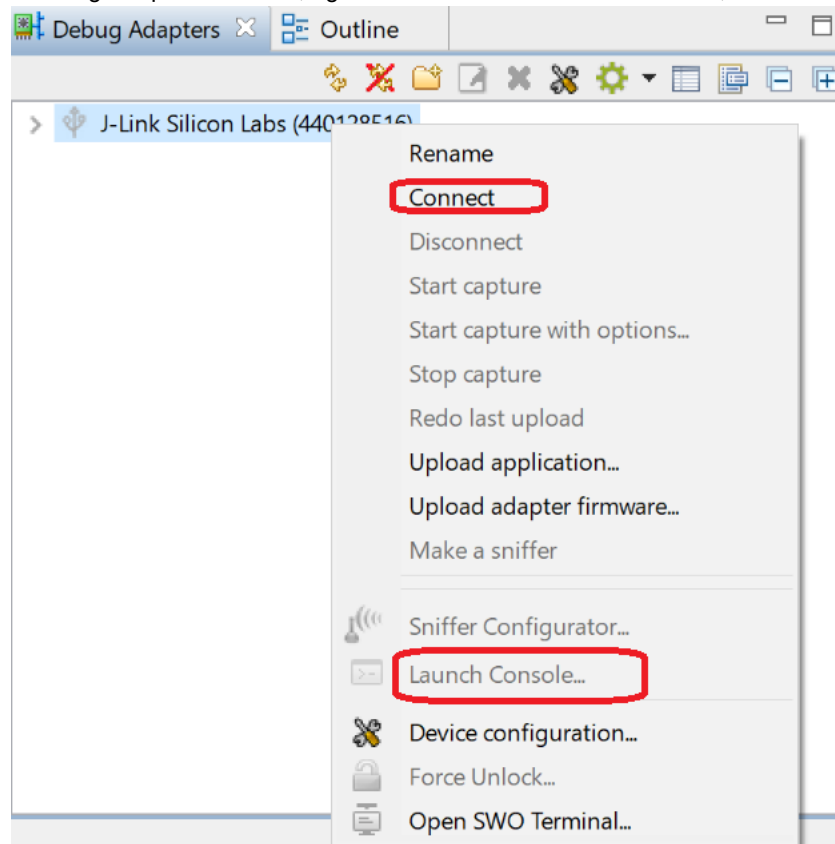
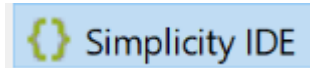
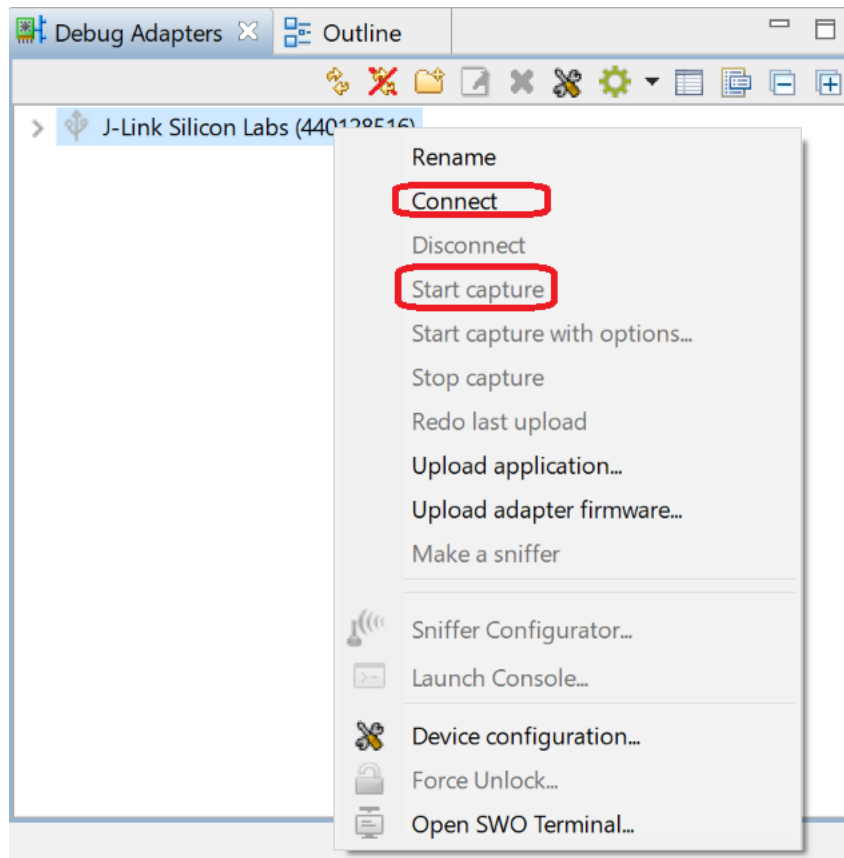


Figure 7 Launch console

## 1.6 How to start Capture



1. Change to “Simplicity IDE” perspective
2. Select your adapter in the “Debug Adapters” window, right click and select “connect”;
3. Select your adapter in the “Debug Adapters” window, right click and select “Start Capture”;

**Figure 8 Start Capture**

## 2 Build the bootloader for OTA client

1. Go to File -> New -> Project. This will bring up the New Project Wizard
2. Select "Silicon Labs AppBuilder Project". Click Next.
3. Select "Gecko Bootloader". Click Next.
4. Select the latest version. (Gecko Bootloader 1.9.2). Click Next.
5. Select "SPI Flash Storage Bootloader (single image)". Click Next.
6. Name your project (Whatever name you want). Click Next.
7. Select board and compiler. Then finish.



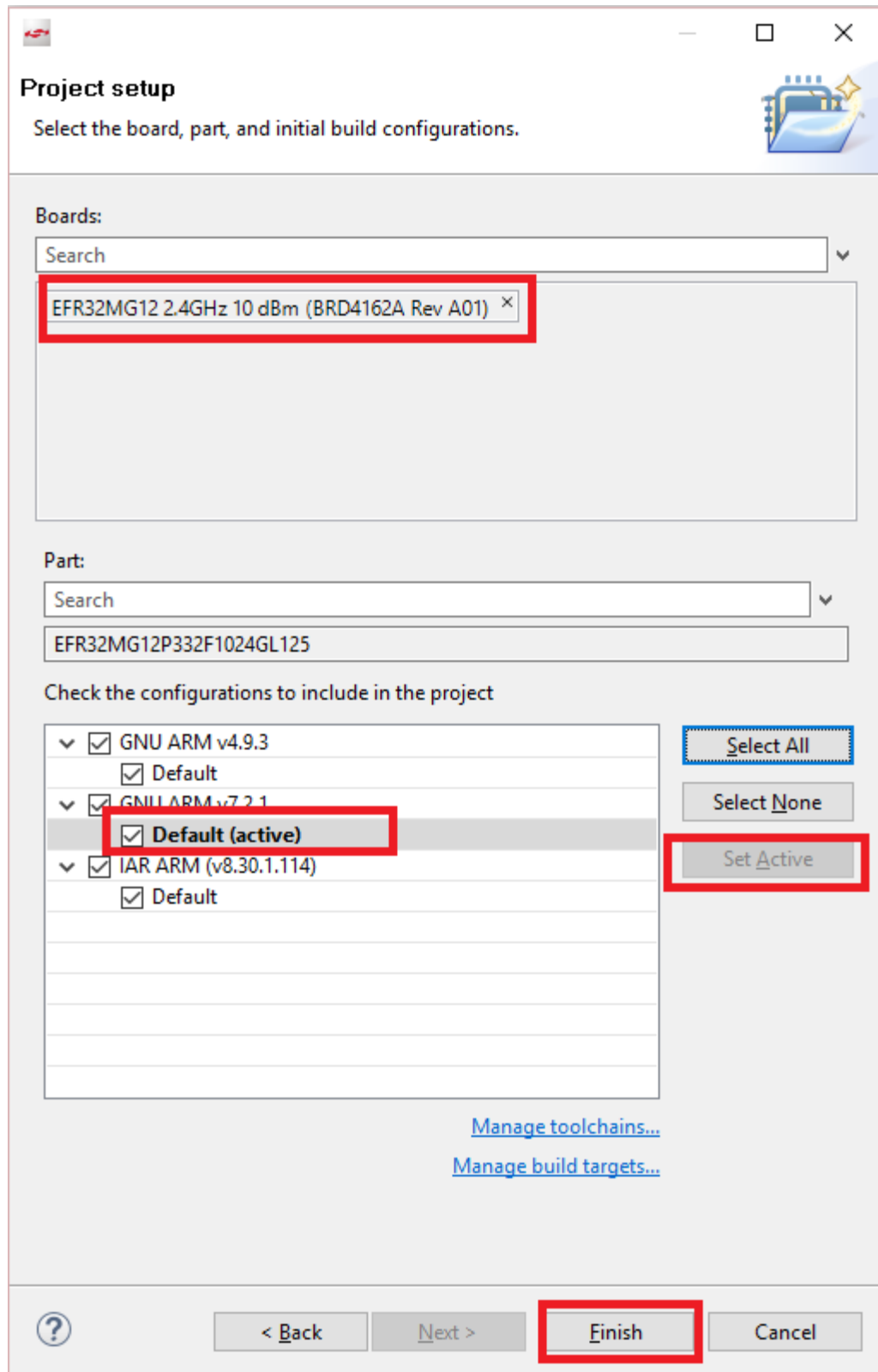



Figure 9 Select board and compiler

8. The new project should have been created now, with the project configuration file (an .isc file) open.
9. Click **Generate**. Notice the project files appearing in Project Explorer. A window saying Generation successful will appear. Click OK.

10. Select the project in Project Explorer window and compile your project by clicking on the Build icon . Ensure that the build completes with 0 errors.

### 3 Build the old version client image

1. Go to File -> New -> Project. This will bring up the New Project Wizard
2. Select "Silicon Labs AppBuilder Project". Click Next.
3. Select "Silicon Labs Zigbee". Click Next.
4. Select our latest EmberZNet stack for Soc (in this case EmberZNet 6.6.4 GA Soc). Click Next.
5. Select "ZigbeeMinimal". Click Next.
6. Name your project, such as "Client". Click Next.
7. In next window (Project Setup), select board to BRD4162A, and compiler to "GCC v7.2" (If you don't have it, please install any other). Click Finish.

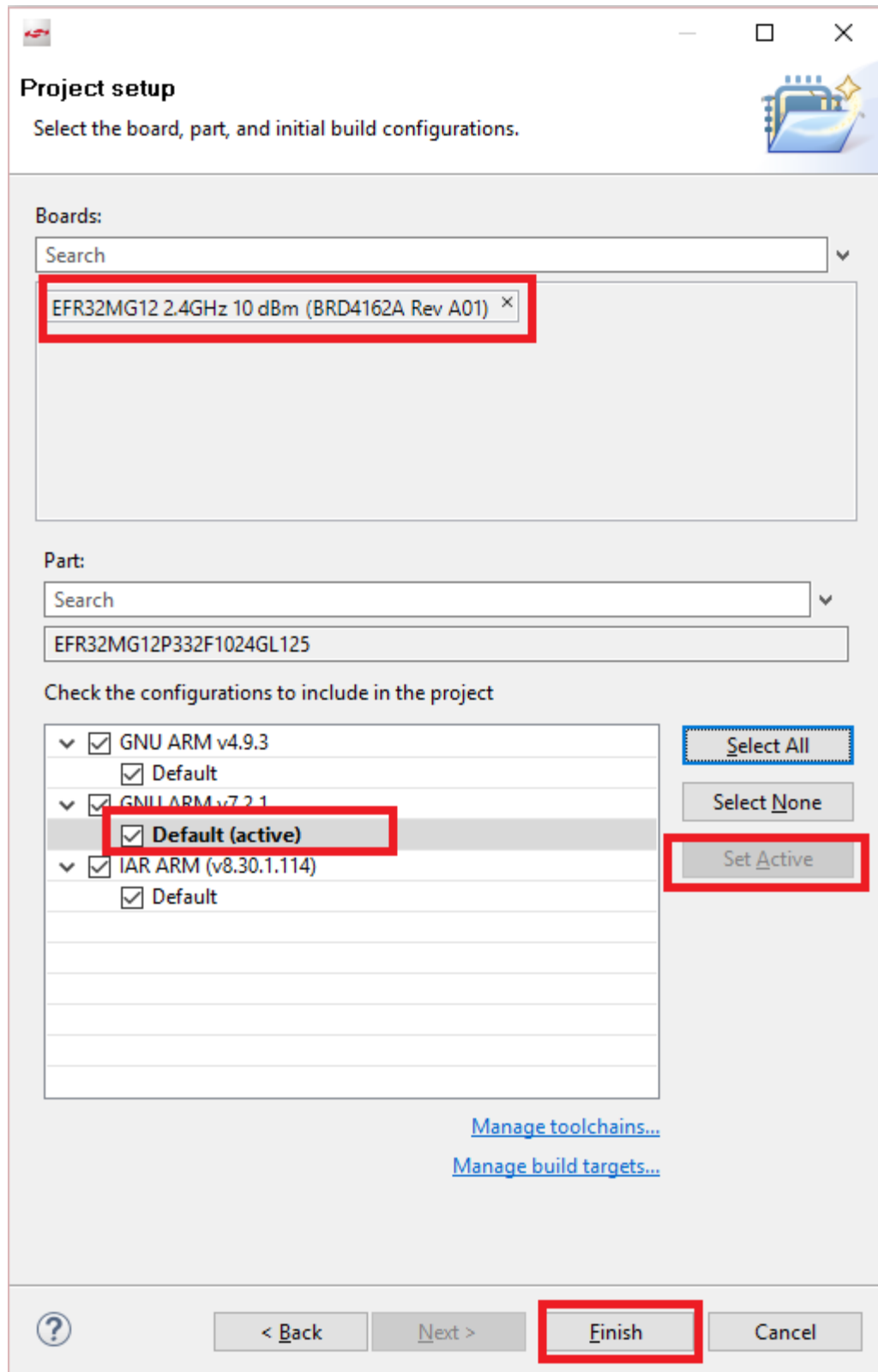


Figure 10 select board and compiler

8. In “ZCL Clusters” tab,
  - a. In “ZCL device type” field, set “ZCL device type” to “HA On/Off Switch”

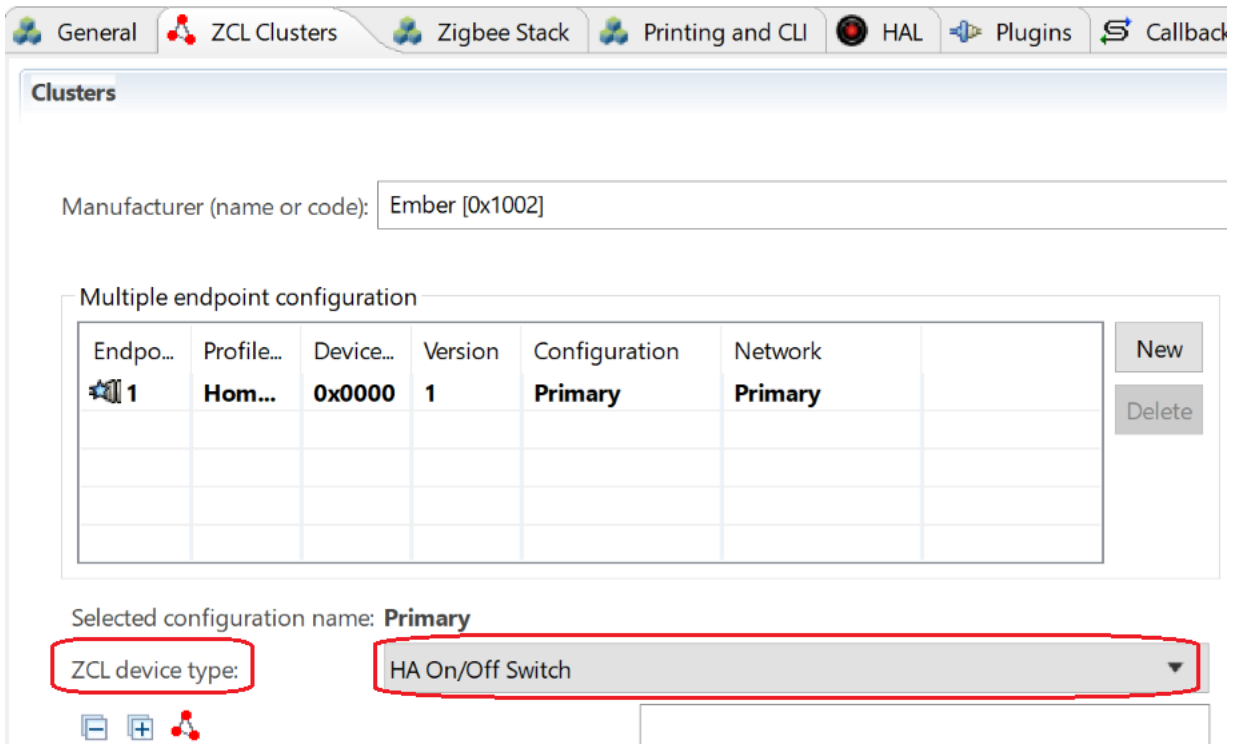


Figure 11 select device type

## b. Enable the client side of “Over the Air Bootloading”

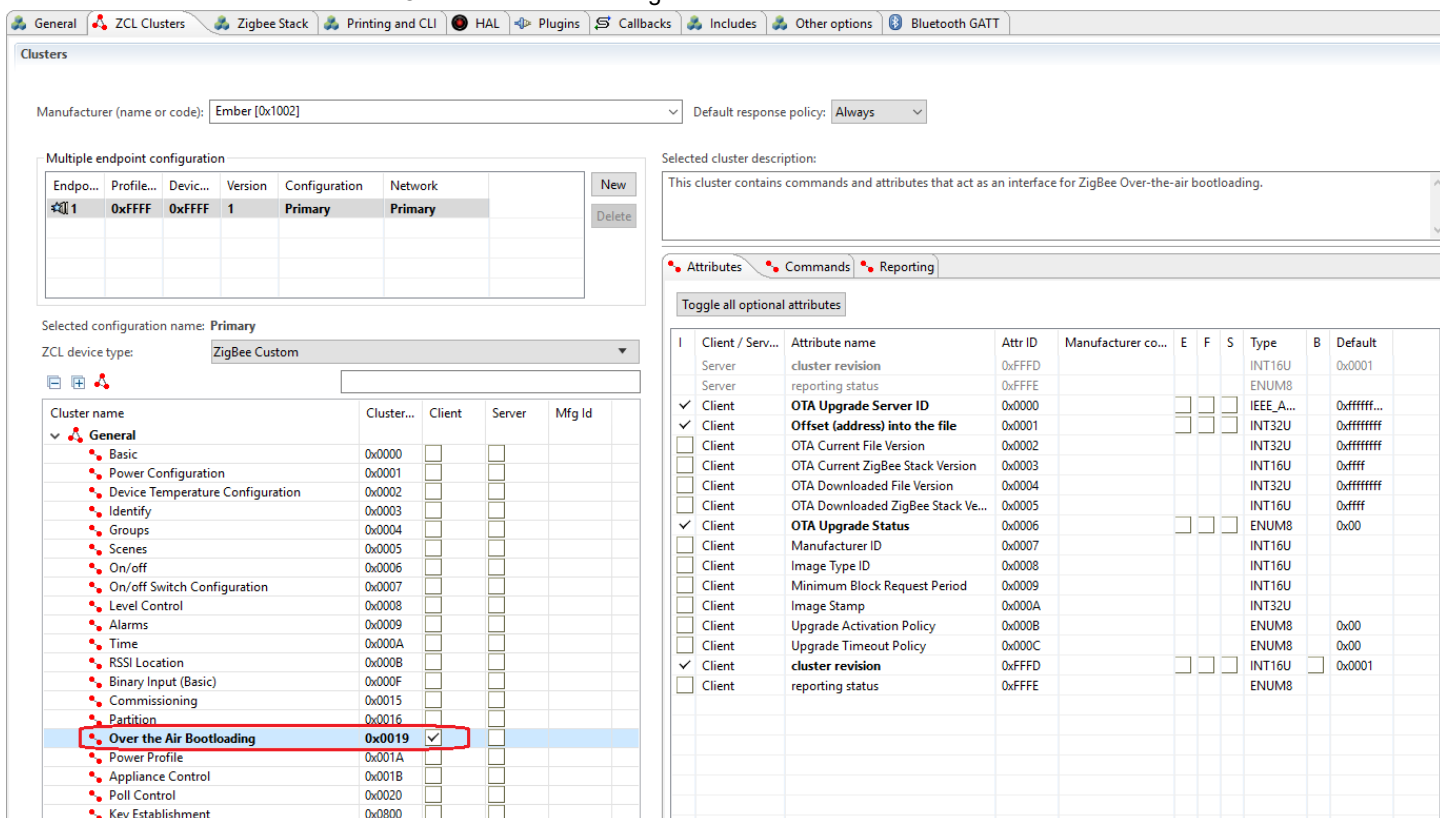


Figure 12 Cluster setting

## 9. Turn to “plugins” tab, select the following plugins:

- Install Code Library
- OTA Bootload Cluster Client

c. OTA Bootload Cluster Client Policy, in the properties page, set the firmware version to 100.

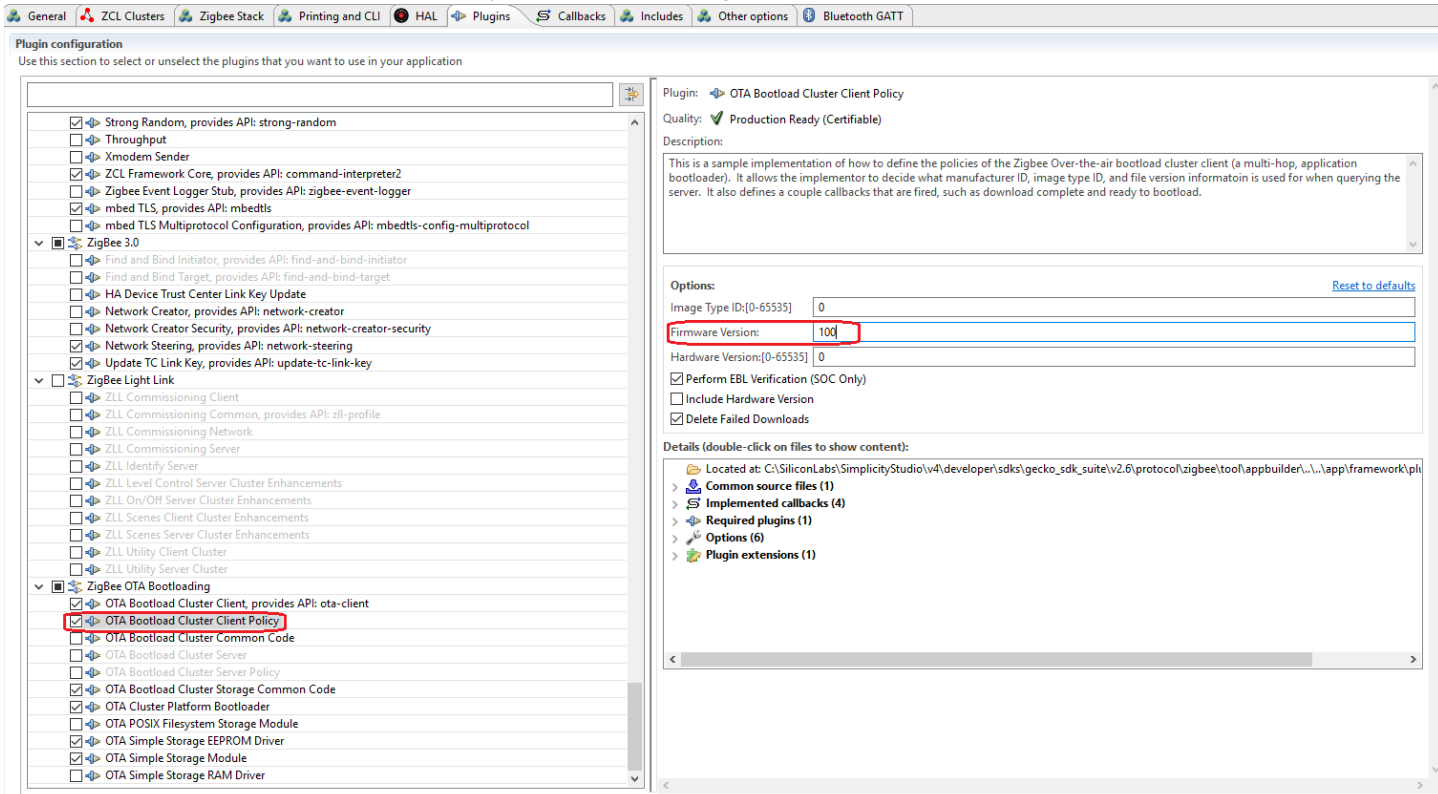


Figure 13 firmware version

- d. OTA Bootload Cluster Common Code
- e. OTA Bootload Cluster Storage Common Code
- f. OTA Cluster Platform Bootloader
- g. OTA Simple Storage Module
- h. OTA Simple Storage EEPROM Driver, in the properties, set the “read-modify-write” option to “false”

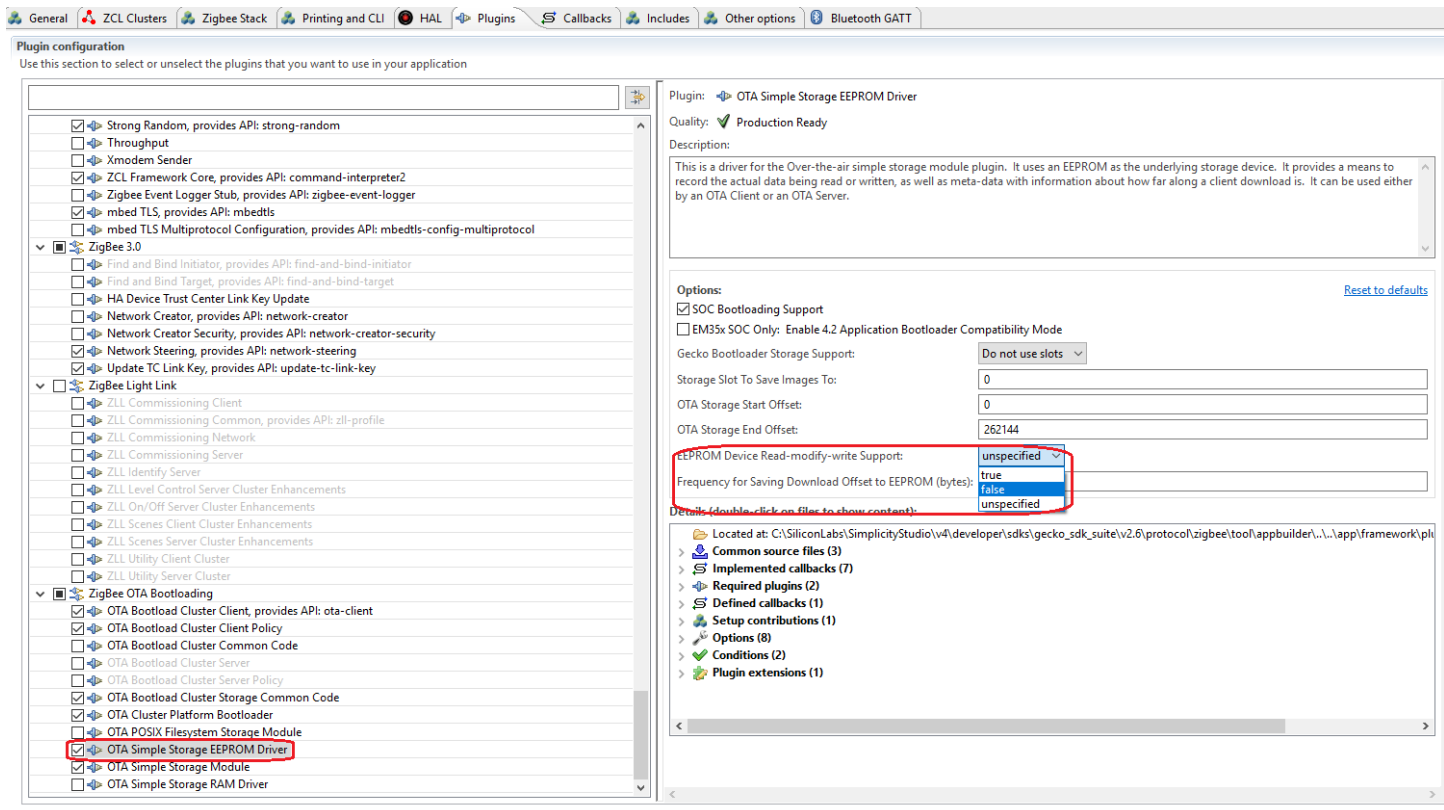



Figure 14 OTA Storage setting

10. Turn to "Printing and CLI" tab, enable the following debug options.
  - a. Unfold "Individual plugin debug printing"
    - i. enable "OTA Simple Storage EEPROM Driver", both "compiled in" and "Enabled at start up";
    - ii. enable "OTA Bootload Cluster Storage Common Code", both "compiled in" and "Enabled at start up";
    - iii. enable "OTA Cluster Platform Bootloader", both "compiled in" and "Enabled at start up";
    - iv. enable "OTA Bootloader Cluster Client", both "compiled in" and "Enabled at start up";
    - v. enable "OTA Bootloader Cluster Client Policy", both "compiled in" and "Enabled at start up";
    - vi. enable "OTA Simple Storage Module", both "compiled in" and "Enabled at start up";
  - b. Unfold "Application specific debug printing"
    - i. enable "OTA Bootloader Cluster", both "compiled in" and "Enabled at start up";
11. Click "Generate". Notice the project files appearing in Project Explorer. A window saying "generating successfully" will appear. Click OK.
12. Select the project in Project Explorer window and compile your project by clicking on the Build icon . Ensure that the build completes with 0 errors.
13. In the build directory, backup the following file to a new folder "V100":
  - a. Client.ota
  - b. Client.s37

#### 4 Build the new version client image

1. Open the isc file of the client project, turn to "plugins" tab, select the plugins "OTA Bootload Cluster Client Policy", in the properties page, set the firmware version to 200.
2. Save and generate the project, then build.
3. In the build directory, backup the following file to a new folder "V200":
  - a. Client.ota
  - b. Client.s37

#### 5 Build the server

We will still use the Z3GatewayHost + NCP as the OTA server.



## 6 Test

1. Flash the bootloader (xxx-combined.s37) and ncp (ncp.s37) image to WSTK (NCP);
2. Flash the bootloader (bootloader-storage-spiflash-single\_xxx-combined.s37) and old version image (Client.s37) to your client.
3. In the “build/exe” folder under the host project, there should be a directory named “ota-files” (If it doesn't exist, create one). Copy the **new version** of the Client.ota to this directory.
4. Find the COM port number of your NCP board.
  - a. Start a cmd window:

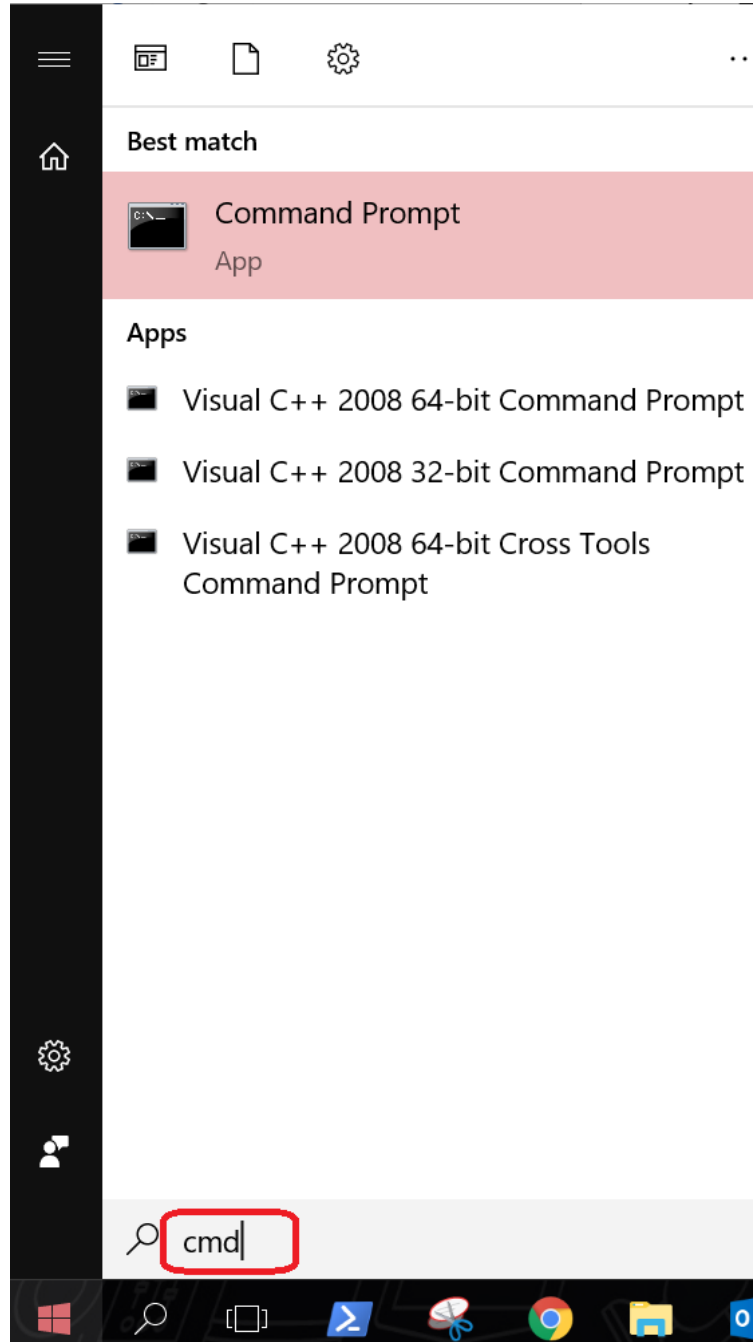


Figure 15 start cmd window

- b. Change to C:\SiliconLabs\SimplicityStudio\v4\developer\adapter\_packs\serial. Then run command:

```
serial.exe -ports
```



```
C:\SiliconLabs\SimplicityStudio\v4\developer\adapter_packs\serial>serial.exe -ports
{
  "ports" :
  [
    {
      "Name" : "COM3",
      "OSID" : "COM3",
      "SerialNumber" : ""
    },
    {
      "Name" : "COM17",
      "OSID" : "COM17",
      "SerialNumber" : "000440128516"
    }
  ]
}
```

Figure 16 get COM port of the WSTK

**5. Make sure your WSTK for NCP is disconnected in Simplicity Studio.**

6. Start Cygwin, and switch to the directory of the Z3GatewayHost project, then switch to “build/exe” directory. Run the following command to start the host:

```
//COM17 is the COM port of the NCP
./Z3GatewayHost.exe -p COM17
```

```
/cygdrive/c/Users/[redacted]/Documents/v4_workspace_2.6/Z3GatewayHost
$ cd build/exe/
/cygdrive/c/Users/[redacted]/Documents/v4_workspace_2.6/Z3GatewayHost/build/exe
$ ./Z3GatewayHost.exe -p COM17
Reset info: 11 (SOFTWARE)
ezsp ver 0x07 stack type 0x02 stack ver. [6.6.4 GA build 180]
Ezsp Config: set source route table size to 0x0064:Success: set
Ezsp Config: set security level to 0x0005:Success: set
Ezsp Config: set address table size to 0x0002:Success: set
Ezsp Config: set TC addr cache to 0x0002:Success: set
Ezsp Config: set stack profile to 0x0002:Success: set
Ezsp Config: set MAC indirect TX timeout to 0x1E00:Success: set
Ezsp Config: set max hops to 0x001E:Success: set
Ezsp Config: set tx power mode to 0x8000:Success: set
Ezsp Config: set supported networks to 0x0001:Success: set
Ezsp Value : set end device keep alive support mode to 0x00000003:Success: set
Ezsp Policy: set binding modify to "allow for valid endpoints & clusters only":Success: set
Ezsp Policy: set message content in msgSent to "return":Success: set
Ezsp Value : set maximum incoming transfer size to 0x00000052:Success: set
Ezsp Value : set maximum outgoing transfer size to 0x00000052:Success: set
Ezsp Config: set binding table size to 0x0010:Success: set
Ezsp Config: set key table size to 0x0040:Success: set
Ezsp Config: set max end device children to 0x0020:Success: set
Ezsp Config: set aps unicast message count to 0x000A:Success: set
Ezsp Config: set broadcast table size to 0x000F:Success: set
Ezsp Config: set neighbor table size to 0x0010:Success: set
NCP supports maxing out packet buffers
Ezsp Config: set packet buffers to 179
Ezsp Config: set end device poll timeout to 0x0001:Success: set
Ezsp Config: set zll group addresses to 0x0000:Success: set
Ezsp Config: set zll rssi threshold to 0xFFD8:Success: set
Ezsp Config: set transient key timeout to 0x00B4:Success: set
Ezsp Endpoint 1 added, profile 0x0104, in clusters: 8, out clusters 20
Ezsp Endpoint 242 added, profile 0xA1E0, in clusters: 0, out clusters 1
MQTT Client Init
```

Figure 17 Start host

You should be able to see the detail info of the OTA image:

```
Found OTA file 'Z3SwitchSoc_Ota_4167A.ota'
Manufacturer ID: 0x1002
Image Type ID: 0x0000
Version: 0x00000064
Header String: EBL Z3SwitchSoc_Ota_4167A
Found 1 files
```

7. On the console of the host, run the following command to create a network.

```
plugin network-creator start 1
```

8. Program a install code string following the steps in section 7.1 of <https://github.com/MarkDing/IoT-Developer-Boot-Camp/wiki/Zigbee-Hands-on-Forming-and-Joining> . After that, derive a link key from the install code with the provided tool HashingTool.exe.



**Figure 18 Derive link key from install code**

9. On the console of the host, use the following command to open the network for the device:

```
//000B57FFFEA8EF42 is the Eui64 of the client
//41618FC0C83B0E14A589954B16E31466 is the install code derived link key

plugin network-creator-security open-with-key {000B57FFFEA8EF42}
{41618FC0C83B0E14A589954B16E31466}
```

10. Open console of the client, then run the following command to start joining.

```
plugin network-steering start 0
```

11. Add the NWK key to Simplicity Studio, then start to capture.

12. On the console of the client, use the following command to start OTA upgrading.

```
plugin ota-client start
```

13. On the console of the client, use the following command to check the firmware version.

```
plugin ota-client info
```