Control Zigbee Device with MQTT Worksheet

In this worksheet we provide a step-by-step guide to create, build and run Z3GatewayHost and NCP sample 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

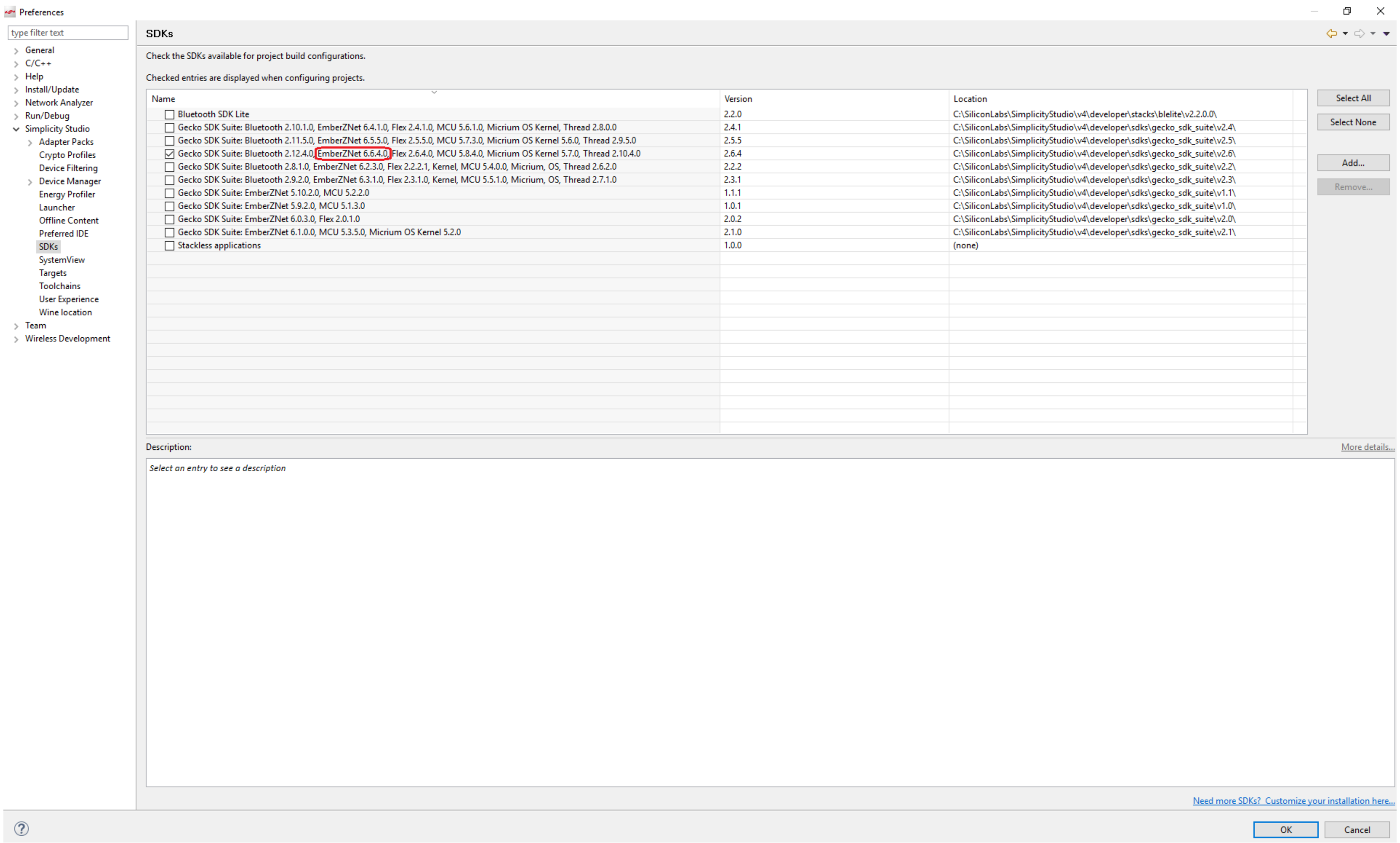
- Step-by-step guide to creating, building and running ZigBee 3.0 applications based on EmberZNet 6.6.4
- Use Simplicity Studio v4 as the development tool
- ZigBee end device polling
- Zigbee end device keepalive and aging
- Zigbee end device rejoin

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



SDKs

Check the SDKs available for project build configurations.

Checked entries are displayed when configuring projects.

Name	Version	Location
□ Bluetooth SDK Lite	2.2.0	C:\SiliconLabs\SimplicityStudio\v4\developer\stacks\blelite\v2.2.0.0\
□ Gecko SDK Suite: Bluetooth 2.10.1.0, EmberZNet 6.4.1.0, Flex 2.4.1.0, MCU 5.6.1.0, Micrium OS Kernel, Thread 2.8.0.0	2.4.1	C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suite\v2.4\
□ Gecko SDK Suite: Bluetooth 2.11.5.0, EmberZNet 6.5.5.0, Flex 2.5.5.0, MCU 5.7.3.0, Micrium OS Kernel 5.6.0, Thread 2.9.5.0	2.5.5	C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suite\v2.5\
☑ Gecko SDK Suite: Bluetooth 2.12.4.0, EmberZNet 6.6.4.0, Flex 2.6.4.0, MCU 5.8.4.0, Micrium OS Kernel 5.7.0, Thread 2.10.4.0	2.6.4	C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suite\v2.6\
□ Gecko SDK Suite: Bluetooth 2.8.1.0, EmberZNet 6.2.3.0, Flex 2.2.2.1, Kernel, MCU 5.4.0.0, Micrium, OS, Thread 2.6.2.0	2.2.2	C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suite\v2.2\
□ Gecko SDK Suite: Bluetooth 2.9.2.0, EmberZNet 6.3.1.0, Flex 2.3.1.0, Kernel, MCU 5.5.1.0, Micrium, OS, Thread 2.7.1.0	2.3.1	C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suite\v2.3\
□ Gecko SDK Suite: EmberZNet 5.10.2.0, MCU 5.2.2.0	1.1.1	C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suite\v1.1\
□ Gecko SDK Suite: EmberZNet 5.9.2.0, MCU 5.1.3.0	1.0.1	C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suite\v1.0\
□ Gecko SDK Suite: EmberZNet 6.0.3.0, Flex 2.0.1.0	2.0.2	C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suite\v2.0\
□ Gecko SDK Suite: EmberZNet 6.1.0.0, MCU 5.3.5.0, Micrium OS Kernel 5.2.0	2.1.0	C:\SiliconLabs\SimplicityStudio\v4\developer\sdks\gecko_sdk_suite\v2.1\
□ Stackless applications	1.0.0	(none)

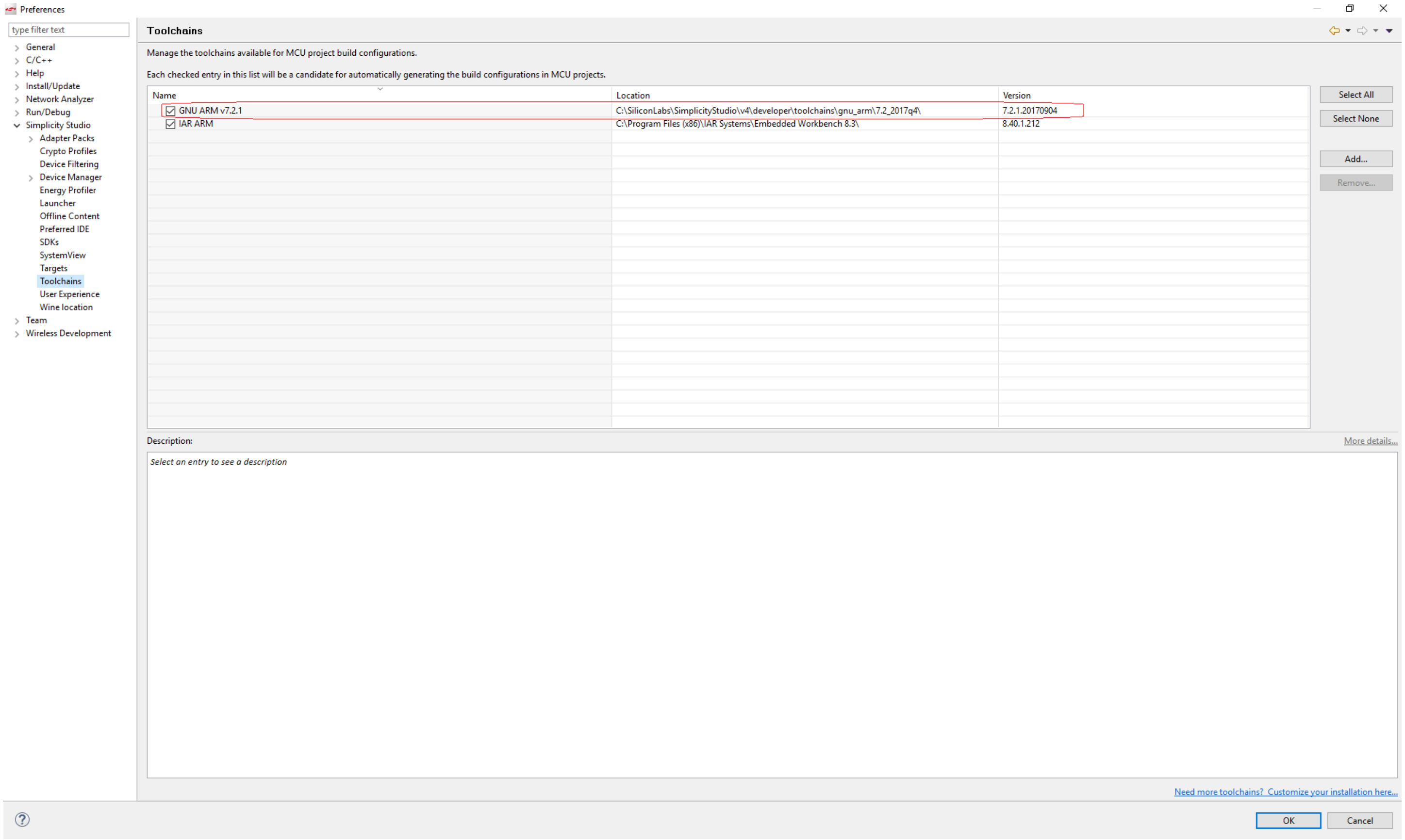
Description:

Select an entry to see a description

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.



Toolchains

Manage the toolchains available for MCU project build configurations.

Each checked entry in this list will be a candidate for automatically generating the build configurations in MCU projects.

Name	Location	Version
☑ GNU ARM v7.2.1	C:\SiliconLabs\SimplicityStudio\v4\developer\toolchains\gnu_arm\7.2_2017q4\	7.2.1.20170904
☑ IAR ARM	C:\Program Files (x86)\IAR Systems\Embedded Workbench 8.3\	8.40.1.212

Description:

Select an entry to see a description

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.

You can also extract the package cygwinx86.rar to other location as long as there is only ASCII characters in the path. In this case you will need to modify the file “cygwin_here.reg” and modify the path according to your location.

```
Windows Registry Editor Version 5.00

[HKEY_CLASSES_ROOT\Folder\shell\open_cygwin]
@="Open Cygwin Here"

[HKEY_CLASSES_ROOT\Folder\shell\open_cygwin\command]
@="C:\\cygwinx86\\bin\\mintty.exe /bin/sh -lc 'cd \"`cygpath \"%V\"`\"; bash'"

[HKEY_CLASSES_ROOT\Directory\Background\shell\open_cygwin]
@="Open Cygwin Here"

[HKEY_CLASSES_ROOT\Directory\Background\shell\open_cygwin\command]
@="C:\\cygwinx86\\bin\\mintty.exe /bin/sh -lc 'cd \"`cygpath \"%V\"`\"; bash'"
```

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

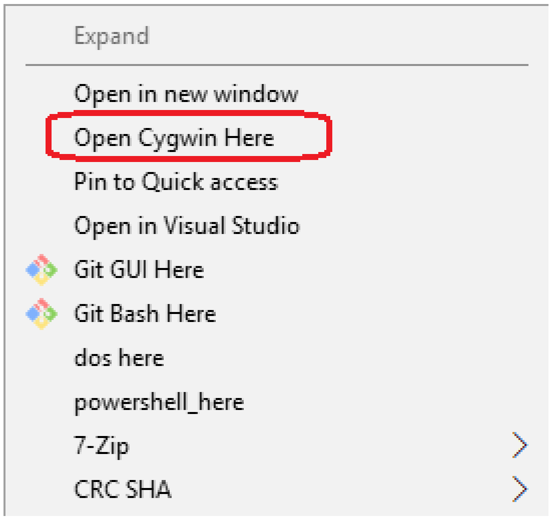
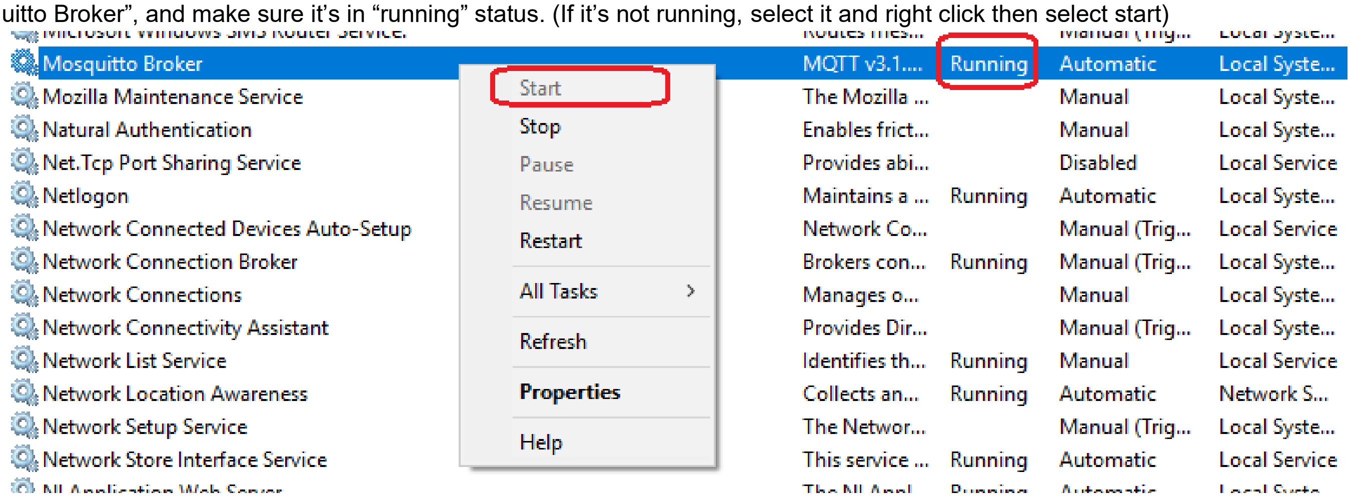


Figure 3 Right-click menu to start cygwin

2. MQTT broker;

Install mosquitto-1.6.7-install-windows-x64.exe, then press “Win+R” and input “services.msc” to start the service manager.

Find “Mosquitto Broker”, and make sure it’s in “running” status. (If it’s not running, select it and right click then select start)



Microsoft Windows SMS Router Service.	Routes mes...		Manual (Trig...	Local Syste...
Mosquitto Broker	MQTT v3.1....	Running	Automatic	Local Syste...
Mozilla Maintenance Service	The Mozilla ...		Manual	Local Syste...
Natural Authentication	Enables frict...		Manual	Local Syste...
Net.Tcp Port Sharing Service	Provides abi...		Disabled	Local Service
Netlogon	Maintains a ...	Running	Automatic	Local Syste...
Network Connected Devices Auto-Setup	Network Co...		Manual (Trig...	Local Service
Network Connection Broker	Brokers con...	Running	Manual (Trig...	Local Syste...
Network Connections	Manages o...		Manual	Local Syste...
Network Connectivity Assistant	Provides Dir...		Manual (Trig...	Local Syste...
Network List Service	Identifies th...	Running	Manual	Local Service
Network Location Awareness	Collects an...	Running	Automatic	Network S...
Network Setup Service	The Networ...		Manual (Trig...	Local Syste...
Network Store Interface Service	This service ...	Running	Automatic	Local Service
NI Application Web Server	The NI Appl...	Running	Automatic	Local Syste

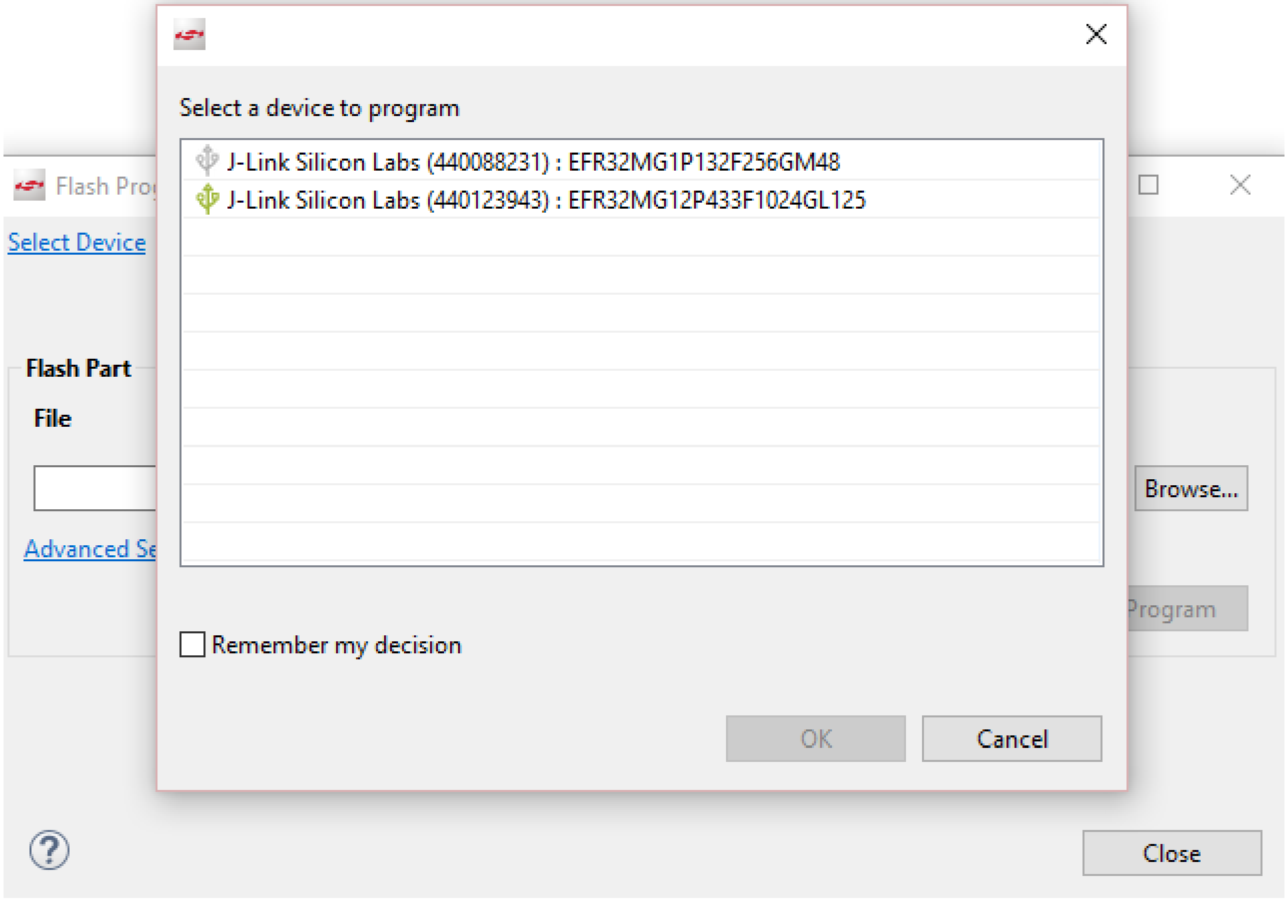
Figure 4 Check and start MQTT broker

3. MQTT client

Install MQTTBox-win.exe.

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;



Select a device to program

J-Link Silicon Labs (440088231) : EFR32MG1P132F256GM48

J-Link Silicon Labs (440123943) : EFR32MG12P433F1024GL125

☐ Remember my decision

Figure 5 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”.

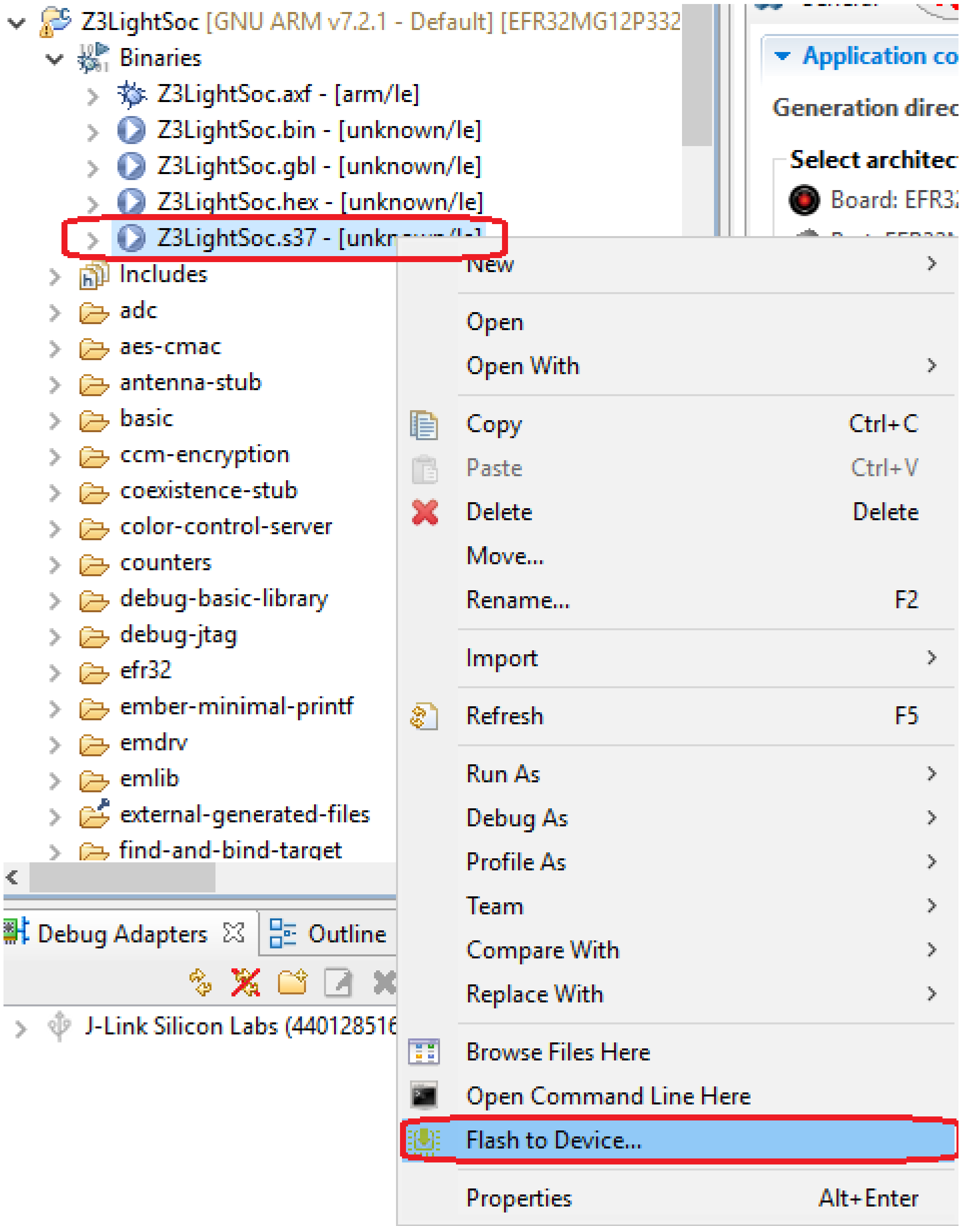
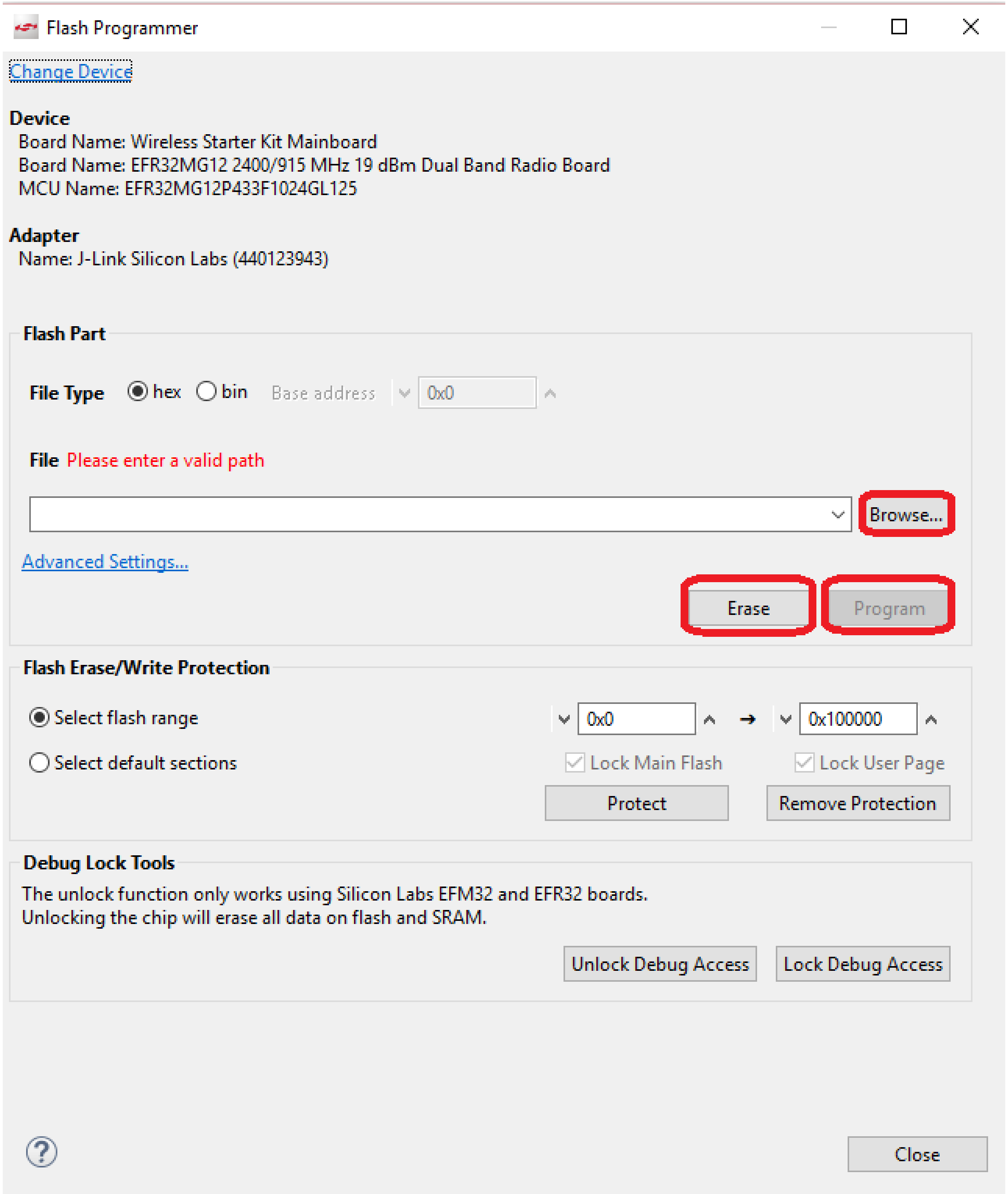


Figure 6 Select image



Flash Programmer

Change Device

Device
Board Name: Wireless Starter Kit Mainboard
Board Name: EFR32MG12 2400/915 MHz 19 dBm Dual Band Radio Board
MCU Name: EFR32MG12P433F1024GL125

Adapter
Name: J-Link Silicon Labs (440123943)

Flash Part

File Type ● hex ○ bin Base address 0x0

File Please enter a valid path

Advanced Settings...

Flash Erase/Write Protection

● Select flash range 0x0 → 0x100000

○ Select default sections ☑ Lock Main Flash ☑ Lock User Page

Debug Lock Tools

The unlock function only works using Silicon Labs EFM32 and EFR32 boards.
Unlocking the chip will erase all data on flash and SRAM.

Figure 7 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 {} Simplicity IDE;
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”;

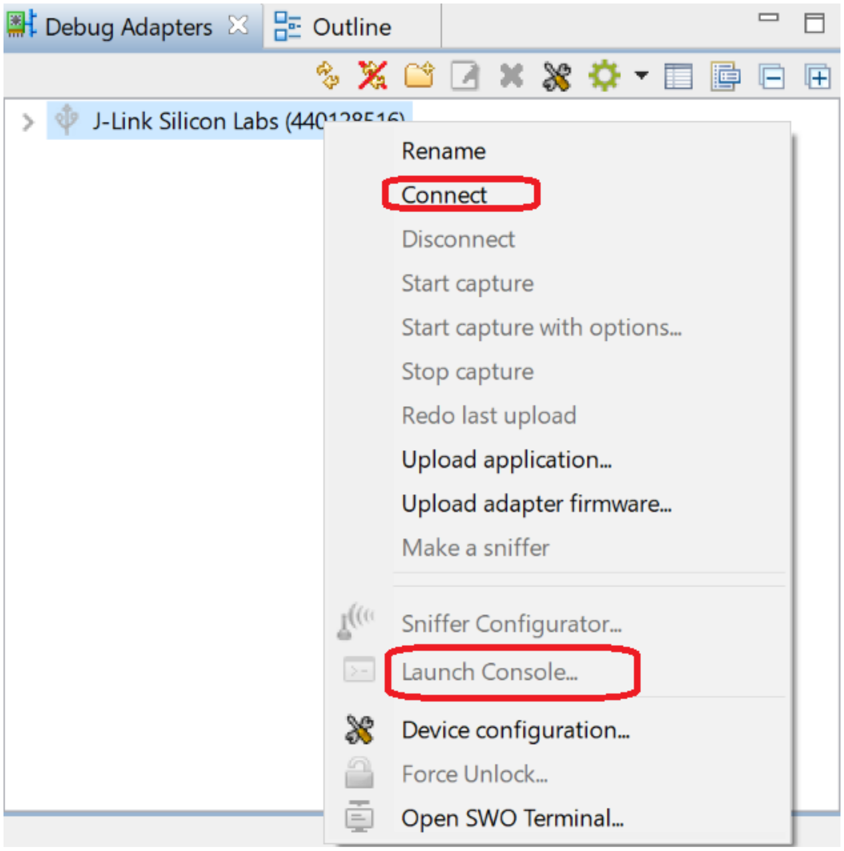
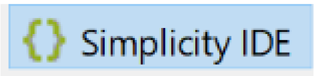


Figure 8 Launch console

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”;

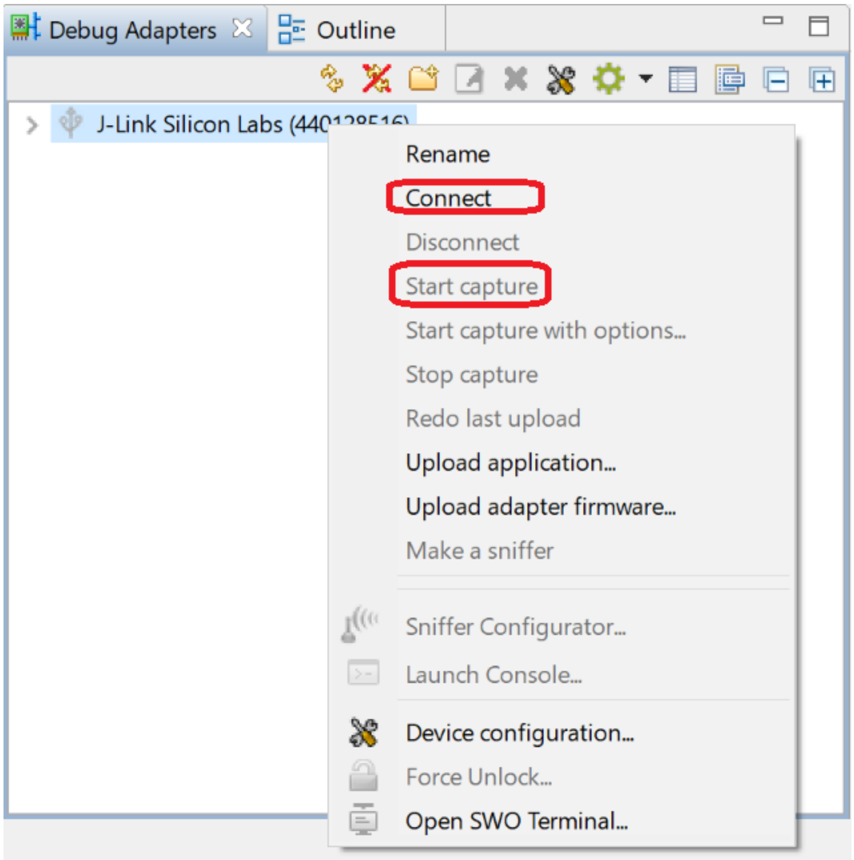
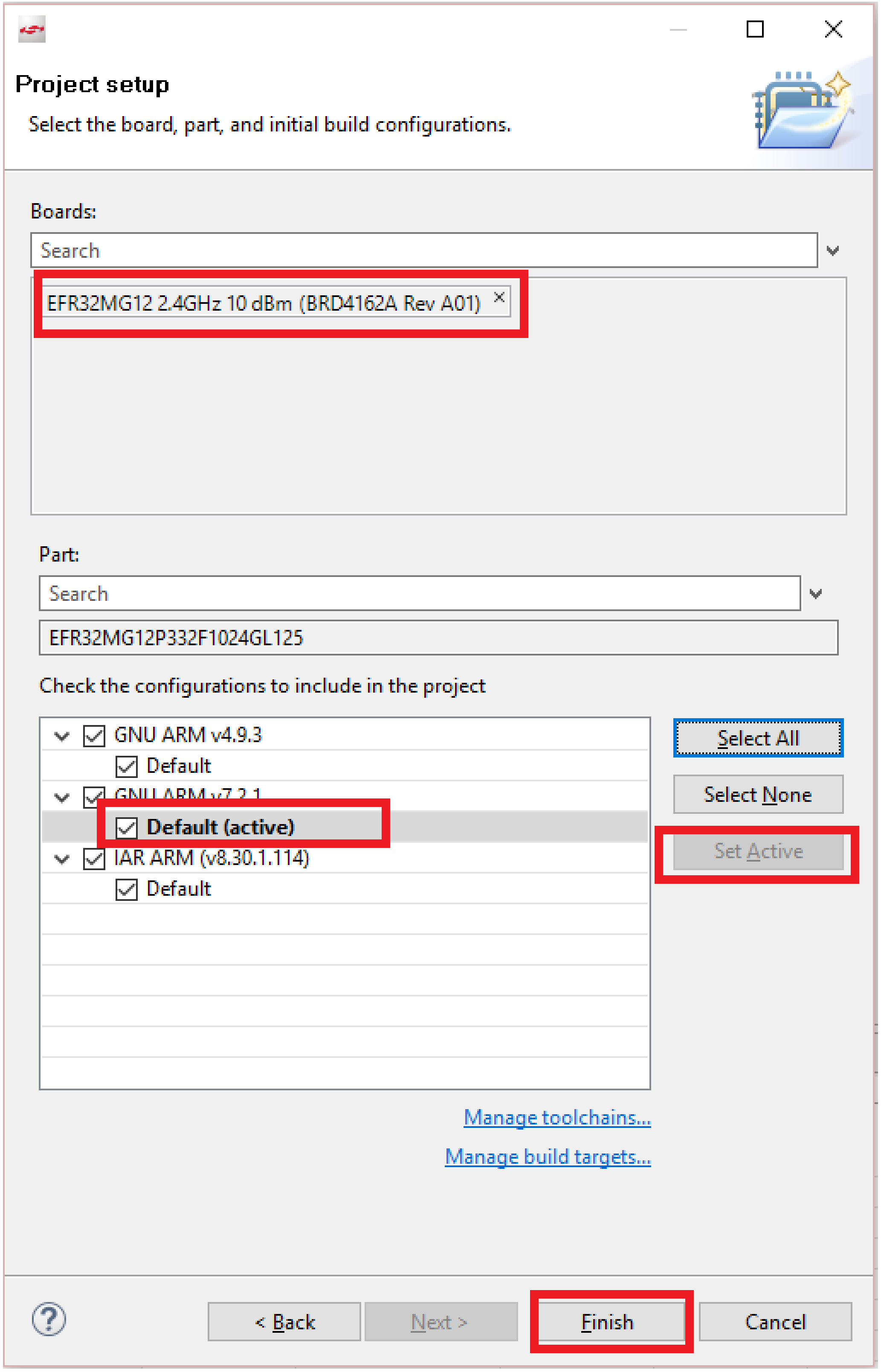


Figure 9 Start Capture

2 Build the bootloader

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 “UART XMODEM Bootloader”. Click Next.
6. Name your project (Whatever name you want). Click Next.
7. Select board and compiler. Then finish.



Project setup

Select the board, part, and initial build configurations.

Boards:

EFR32MG12 2.4GHz 10 dBm (BRD4162A Rev A01)

Part:

EFR32MG12P332F1024GL125

Check the configurations to include in the project

- ☑ GNU ARM v4.9.3
  - ☑ Default
- ☑ GNU ARM v7.2.1
  - ☑ **Default (active)**
- ☑ IAR ARM (v8.30.1.114)
  - ☑ Default

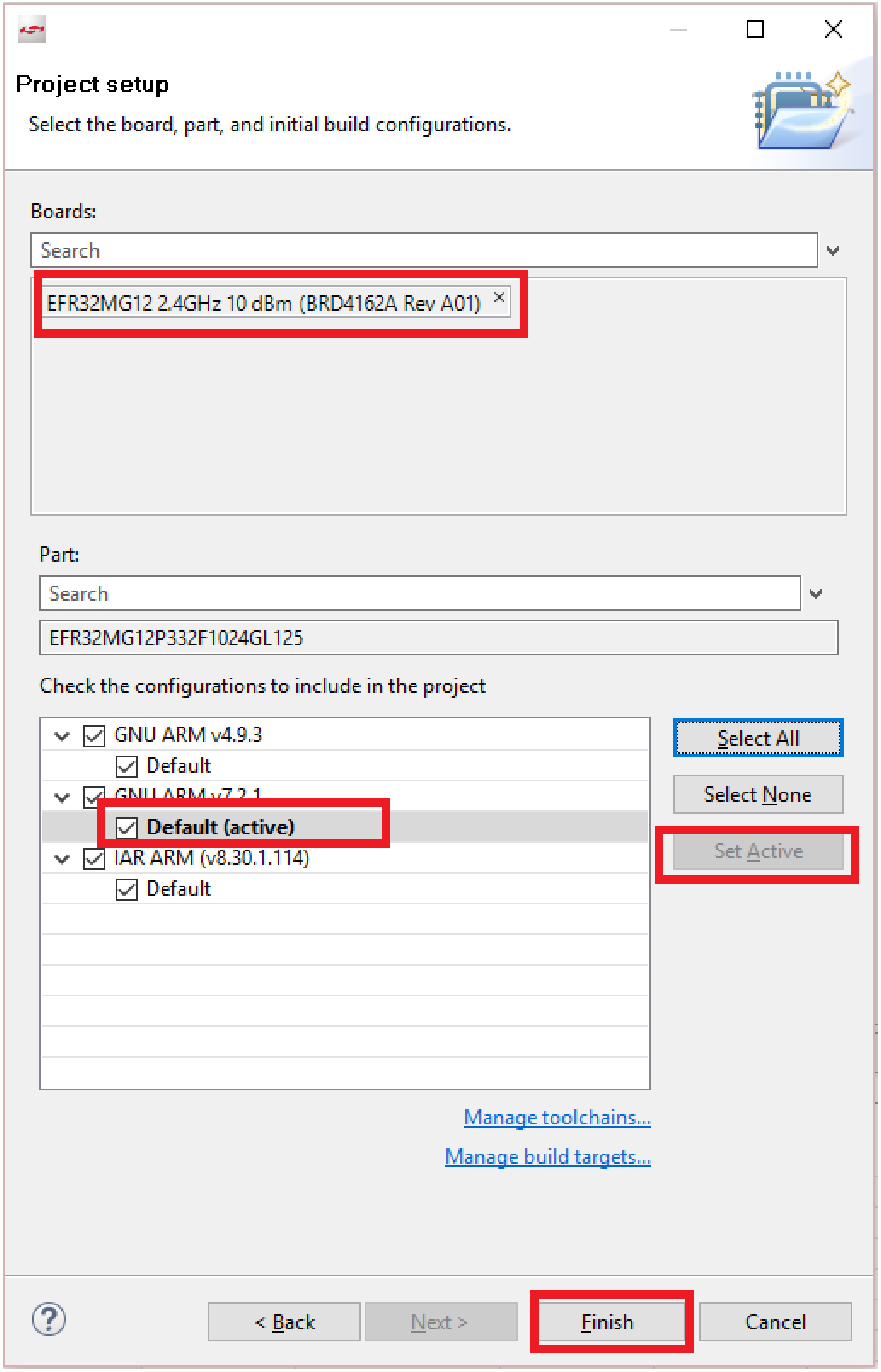
Figure 10 Select board and compiler

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 NCP 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 “Customizable network coprocessor (NCP) Applications”. Click Next.
4. Select “NCP UART HW (Hardware Flow Control)”. Click Next.
5. Name your project, such as “ncp”. Click Next.
6. 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.



Project setup

Select the board, part, and initial build configurations.

Boards:

Search

EFR32MG12 2.4GHz 10 dBm (BRD4162A Rev A01) ×

Part:

Search

EFR32MG12P332F1024GL125

Check the configurations to include in the project

- ☑ GNU ARM v4.9.3
  - ☑ Default
- ☑ GNU ARM v7.2.1
  - ☑ **Default (active)**
- ☑ IAR ARM (v8.30.1.114)
  - ☑ Default

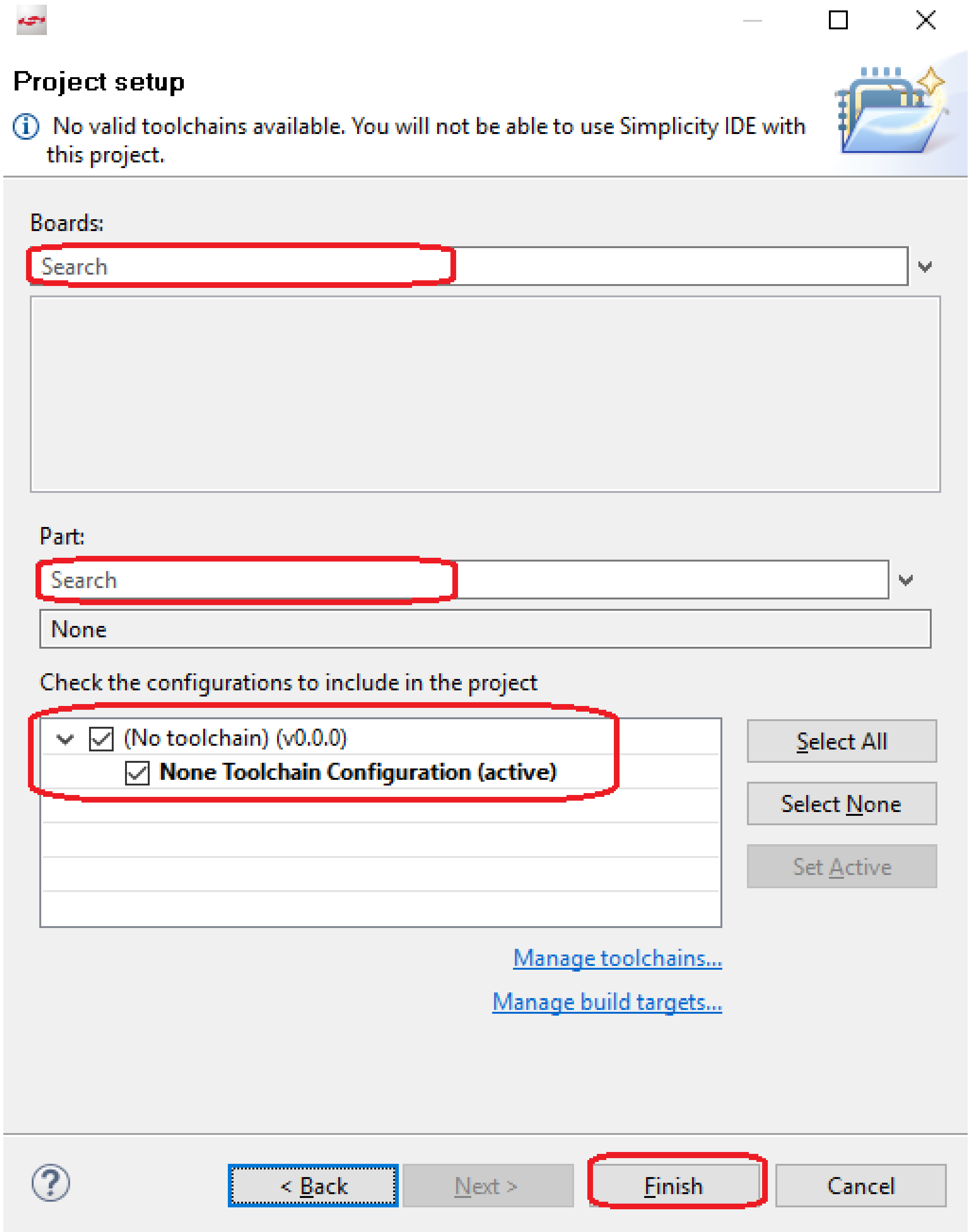
Figure 11 select board and compiler

7. Click “Generate”. Notice the project files appearing in Project Explorer. A window saying “generating successfully” will appear. Click OK.

8. 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.

4 Build the host program

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 host (in this case EmberZNet 6.6.4 GA Host). Click Next.
5. Select “Z3Gateway”. Click Next.
6. Name your project, such as “Z3GatewayHost”. Click Next.
7. In next window (Project Setup), leave the “Boards” and “Parts” field empty, then finish.



Project setup

ⓘ No valid toolchains available. You will not be able to use Simplicity IDE with this project.

Boards: Search

Part: Search

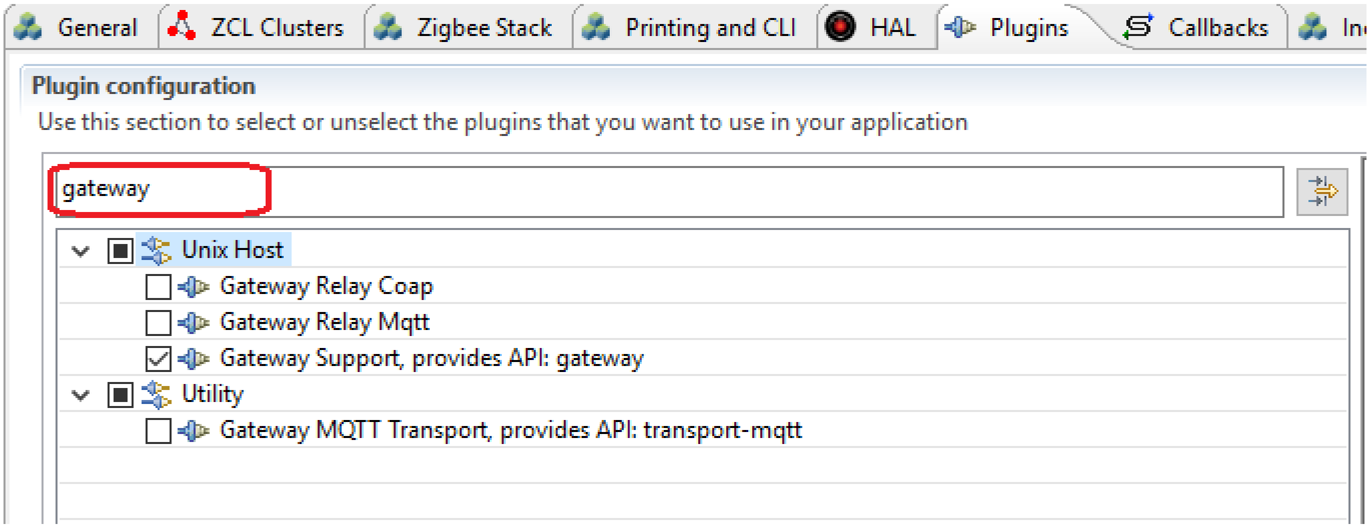
None

Check the configurations to include in the project

- ☑ (No toolchain) (v0.0.0)
  - ☑ **None Toolchain Configuration (active)**

Figure 12 select board and compiler

8. You need to select the following plugins. You can input the name in the filter to navigate to the plugins quickly.



Plugin configuration

Use this section to select or unselect the plugins that you want to use in your application

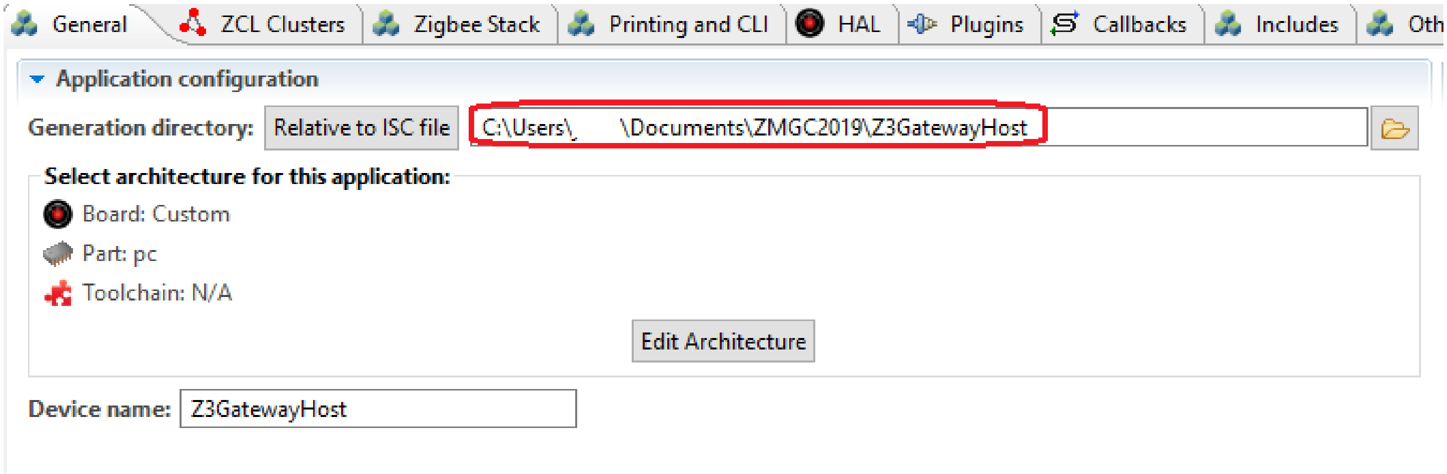
gateway

- ■ Unix Host
  - □ Gateway Relay Coap
  - □ Gateway Relay Mqtt
  - ☑ Gateway Support, provides API: gateway
- ■ Utility
  - □ Gateway MQTT Transport, provides API: transport-mqtt

Figure 13 use filter to navigate the plugins

a. Gateway MQTT Transport
b. Paho MQTT
c. CJSON
d. Gateway Support

9. Use “Ctrl+S” to save, then click “Generate”. Notice the project files appearing in Project Explorer. A window saying “generating successfully” will appear. Click OK.
10. If you have a right-click menu to start Cygwin, then just change to the host project directory and then right click to start Cygwin here. Otherwise, change to Cygwin you extracted, run “Cygwin.bat” to start Cygwin. Change to the project folder and run command “make”. You can find the project directory in “General” tab:



Application configuration

Generation directory: Relative to ISC file C:\Users_ \Documents\ZMGC2019\Z3GatewayHost

Select architecture for this application:

Board: Custom

Part: pc

Toolchain: N/A

Device name: Z3GatewayHost

Figure 14 find project directory

Pay attention that you need to replace the windows partition string to Cygwin format. (If you have a right-click menu to start Cygwin, you can directly change to the host project directory and the right click to start cgywin)

e.g:

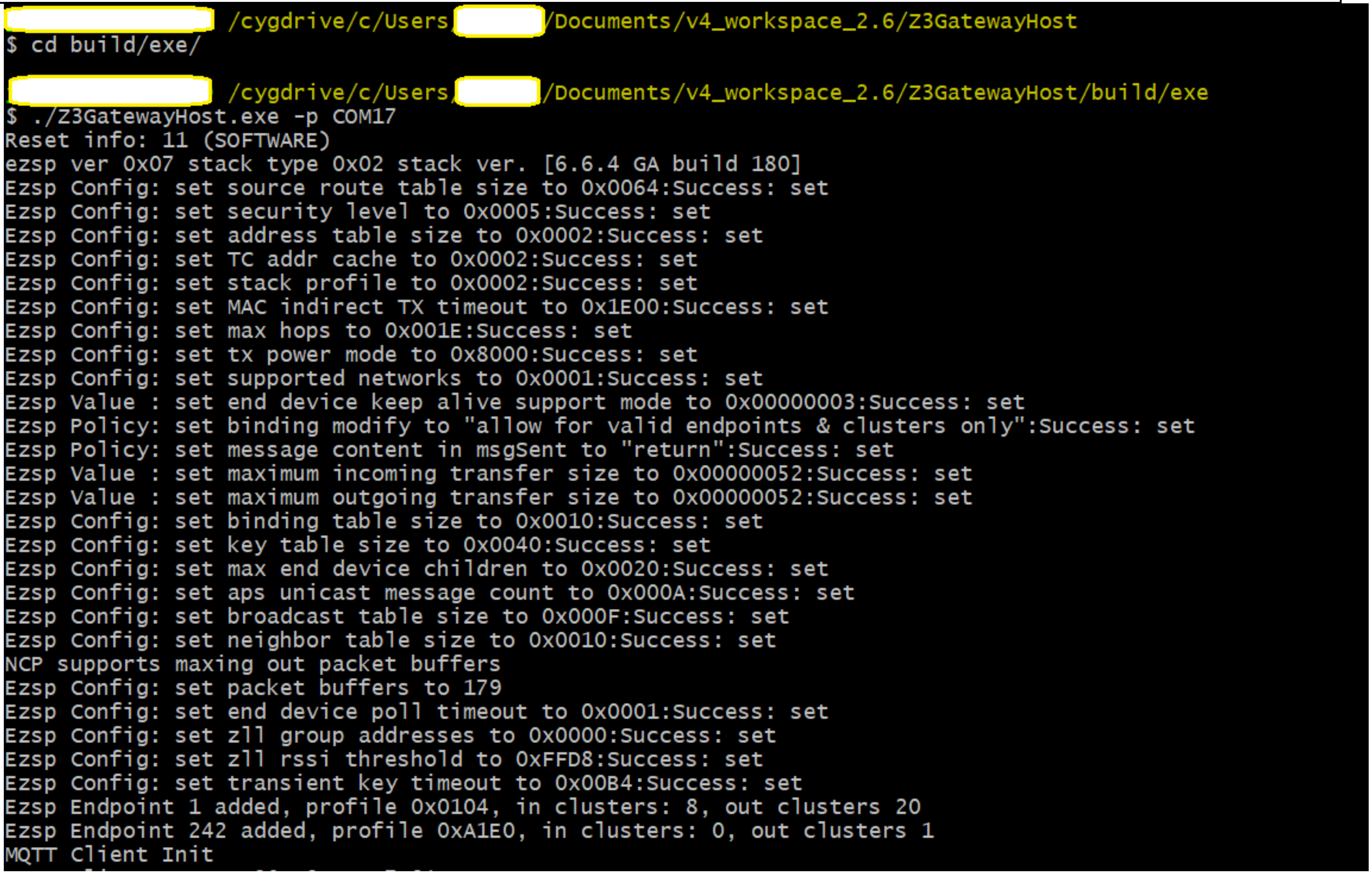
```
//Windows path: C:\SiliconLabs\SimplicityStudio
cd /cygdrive/c/SiliconLabs/SimplicityStudio
```

11. Wait until the build finished. If there is no error, there would be a program “Z3GatewayHost.exe” generated in “build\exe” under your project folder.

5 Test

1. Flash the bootloader (xxx-combined.s37) and ncp (ncp.s37) image to WSTK (NCP);
2. Flash the provided bootloader (bootloader-storage-internal-single-combined-Z3light.s37) and Z3Light (Z3LightSoc.s37) sample to your light.
3. Find the COM port number of your NCP board. You can just connect the WSTK for NCP and unplug all other WSTKs, then check the COM port number in “Device Manager” on your PC.
4. **Make sure your WSTK for NCP is disconnected in Simplicity Studio.**
5. 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
```



```
[redacted] /cygdrive/c/Users/[redacted]/Documents/v4_workspace_2.6/Z3GatewayHost
$ cd build/exe/

[redacted] /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 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 15 Start host

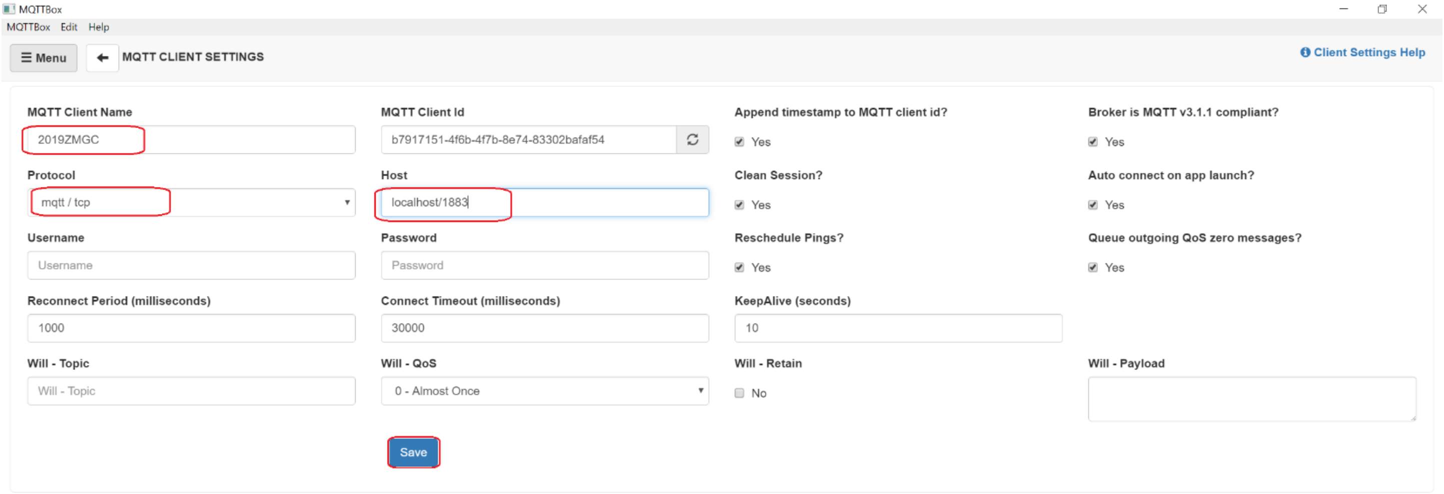
Wait until you see the following message. The host program should be able to connect to the MQTT broker in a few seconds.

```
MQTT connected to broker
MQTT connected, starting gateway heartbeat and command processing
Subscribing to topic "gw/90FD9FFFFE7B81BD/commands" using QoS2
Subscribing to topic "gw/90FD9FFFFE7B81BD/publishstate" using QoS2
Subscribing to topic "gw/90FD9FFFFE7B81BD/updatesettings" using QoS2
```

6. Start MQTT client (MQTTBox).

a. Press button “Create MQTT Client” to create a new client.

b. Fill the “MQTT Client Name”, “Protocol” and “Host” field, and leave the other fields as is. After that, press “save”.



MQTT Client Name: 2019ZMGC

MQTT Client Id: b7917151-4f6b-4f7b-8e74-83302bafaf54

Append timestamp to MQTT client id? ☑ Yes

Broker is MQTT v3.1.1 compliant? ☑ Yes

Protocol: mqtt / tcp

Host: localhost/1883

Clean Session? ☑ Yes

Auto connect on app launch? ☑ Yes

Username: Username

Password: Password

Reschedule Pings? ☑ Yes

Queue outgoing QoS zero messages? ☑ Yes

Reconnect Period (milliseconds): 1000

Connect Timeout (milliseconds): 30000

KeepAlive (seconds): 10

Will - Topic: Will - Topic

Will - QoS: 0 - Almost Once

Will - Retain: ☐ No

Will - Payload:

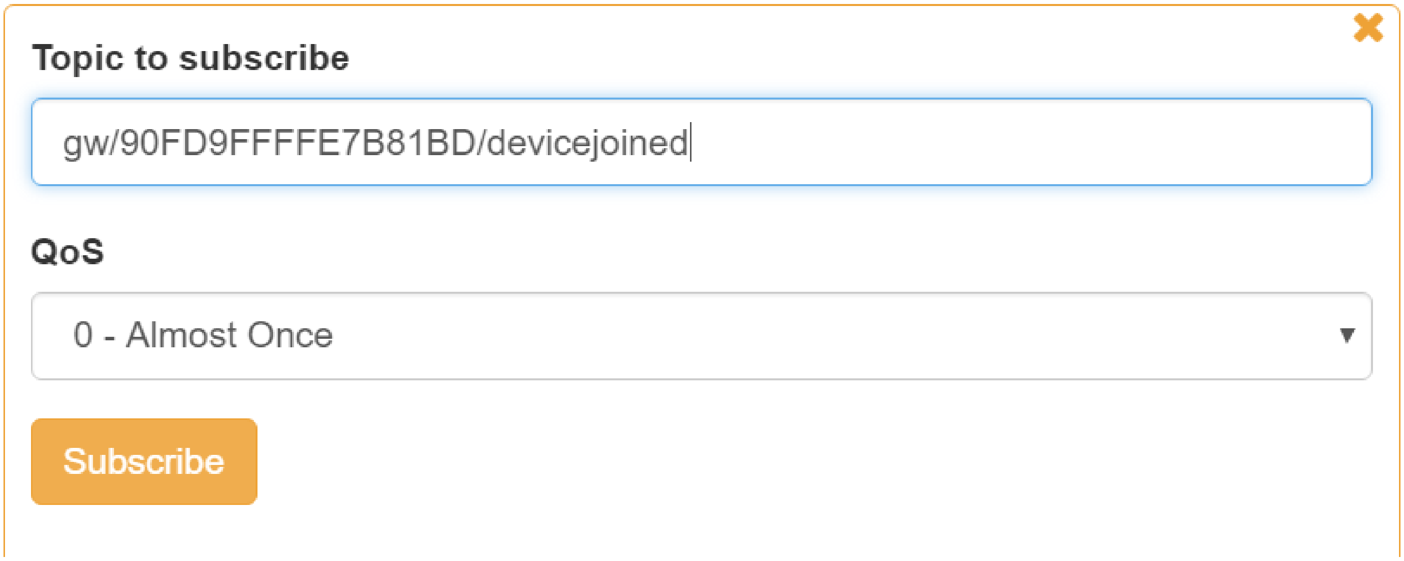
Figure 16 create mqtt client

7. Subscribe the following topic to monitor the device status change.

```
gw/<eui64_id>/devicejoined
gw/<eui64_id>/deviceleft
```

You need to replace the field <eui64_id> to the real Eui64 of your NCP. You can run “info” command in the host console to get that.

```
Z3GatewayHost>info
MFG String:
AppBuilder MFG Code: 0x1002
node [(>)000B57FFFEA8EF3C] chan [0] pwr [-1]
panID [0xFFFF] nodeID [0xFFFE] xpan [0x(>)0000000000000000]
ezsp ver 0x07 stack type 0x02 stack ver. [6.6.4 GA build 180]
nodeType [0x04]
Security level [05]
network state [00] Buffs: 59 / 72
Ep cnt: 2
ep 1 [endpoint enabled, device enabled] nwk [0] profile [0x0104] devId [0x0007] ver [0x01]
    out(client) cluster: 0x0000 (Basic)
    in (server) cluster: 0x0000 (Basic)
    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: 0x0006 (On/off)
    out(client) cluster: 0x0008 (Level Control)
    in (server) cluster: 0x0008 (Level Control)
    in (server) cluster: 0x000A (Time)
    in (server) cluster: 0x0019 (Over the Air Bootloading)
    in (server) cluster: 0x001A (Power Profile)
    out(client) cluster: 0x0020 (Poll Control)
    out(client) cluster: 0x0300 (Color Control)
    in (server) cluster: 0x0300 (Color Control)
    out(client) cluster: 0x0400 (Illuminance Measurement)
    out(client) cluster: 0x0402 (Temperature Measurement)
    out(client) cluster: 0x0405 (Relative Humidity Measurement)
    out(client) cluster: 0x0406 (Occupancy Sensing)
    out(client) cluster: 0x0500 (IAS Zone)
    out(client) cluster: 0x0702 (Simple Metering)
    out(client) cluster: 0x0B01 (Meter Identification)
    out(client) cluster: 0x0B03 (Appliance Statistics)
    out(client) cluster: 0x0B04 (Electrical Measurement)
    out(client) cluster: 0xFC01 (Configuration Cluster)
    out(client) cluster: 0xFC02 (MFGLIB Cluster)
ep 242 [endpoint enabled, device enabled] nwk [0] profile [0xA1E0] devId [0x0061] ver [0x01]
    out(client) cluster: 0x0021 (Green Power)
Nwk cnt: 1
nwk 0 [Primary (pro)]
  nodeType [0x01]
  securityProfile [0x05]
```


Topic to subscribe

gw/90FD9FFFFE7B81BD/devicejoined

QoS

0 - Almost Once

Figure 17 subscribe MQTT topic

8. Publish the following topic to Z3GatewayHost, so that the input command can be run on the host.
    a. Create a network with the following command

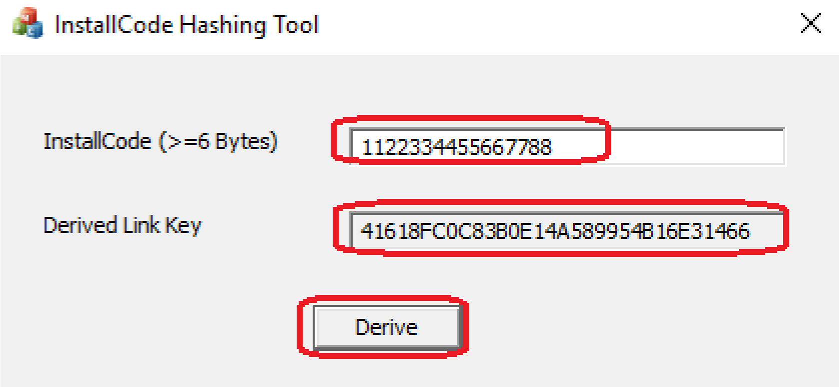
Topic: (You need to replace the red string with the Eui64 of your gateway)

```
Topic: gw/<eui64_id>/commands
```

Payload:

```
{
"commands":[{
"command":
"plugin network-creator start 1",
"postDelayMs":0
}]
}
```

9. 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.



InstallCode Hashing Tool

InstallCode (>=6 Bytes) 1122334455667788

Derived Link Key 41618FC0C83B0E14A589954B16E31466

Figure 18 derive link key from install code

10. Use the following command to open network for the light:

Topic: (You need to replace the red string with the Eui64 of your gateway)

```
Topic: gw/<eui64_id>/commands
```

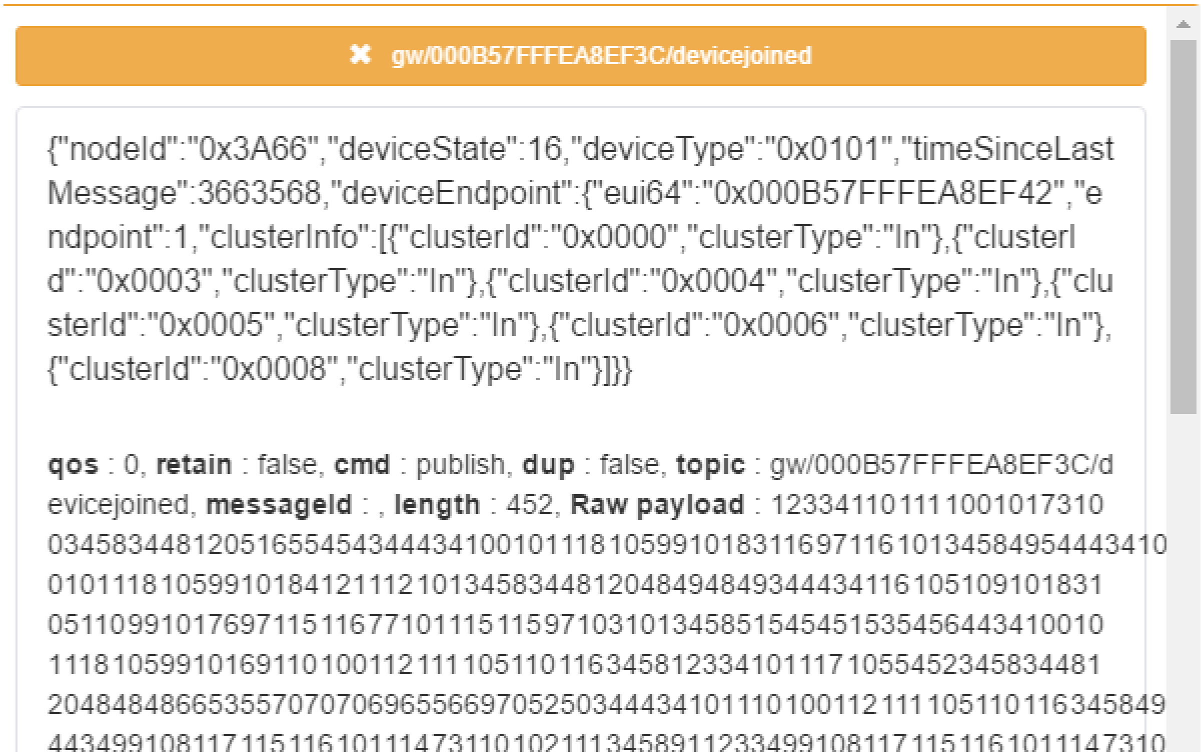
Payload: (The red string is the Eui64 of the light. The blue string is the derived link key)

```
{
"commands":[{
"command":
" plugin network-creator-security open-with-key {000B57FFFEA8EF42}
{41618FC0C83B0E14A589954B16E31466}",
"postDelayMs":0
}]
}
```

11. Run the following command on the console of the light to start joining:

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

12. When the light joined the network, you should be able to see some messages on the subscribed topic “gw/<eui64_id>/devicejoined“ :



{"nodeId":"0x3A66","deviceState":16,"deviceType":"0x0101","timeSinceLastMessage":3663568,"deviceEndpoint":{"eui64":"0x000B57FFFEA8EF42","endpoint":1,"clusterInfo":[{"clusterId":"0x0000","clusterType":"In"},{"clusterId":"0x0003","clusterType":"In"},{"clusterId":"0x0004","clusterType":"In"},{"clusterId":"0x0005","clusterType":"In"},{"clusterId":"0x0006","clusterType":"In"},{"clusterId":"0x0008","clusterType":"In"}]}}

qos : 0, **retain** : false, **cmd** : publish, **dup** : false, **topic** : gw/000B57FFFEA8EF3C/devicejoined, **messageId** : , **length** : 452, **Raw payload** :

123341101111001017310034583448120516554543444341001011181059910183116971161013458495444341001011181059910184121112101345834481204849484934443411610510910183105110991017697115116771011151159710310134585154545153545644341001011181059910169110100112111105110116345812334101117105545234583448120484848665355707070696556697052503444341011101001121111051101163458494434991081171151161011147311010211134589112334991081171151161011147310

Figure 19 Message on subscribed topic

13. Use the following command to control the light. The LED0 should toggle after you run this command.

Topic: (You need to replace the red string with the Eui64 of your gateway)

```
Topic: gw/<eui64_id>/commands
```

Payload: (here the 0x1234 is the node id of the light. You need to modify it according to your envi-ronment.)

```
{
"commands":[
{
"command":
"zcl on-off toggle",
"postDelayMs":0
},
{
"command":
" send 0x1234 1 1",
"postDelayMs":0
}
]
}
```