



Quick Start Guide: Build Z3GatewayFreeRTOS Cloud Gateway Example on AmebaD SDK and Run on LCGW

This article serves as a quick guidance for users wishing to build Silicon Labs' Z3GatewayFreeRTOS cloud gateway example application on Realtek's AmebaD Wi-Fi SDK.

Silicon Labs' EFR32MG21/MG13 is used as the Zigbee NCP while the targetted NCP host is Realtek RTL872xCS/DN Wi-Fi SoC.

This document is aligned with the v1.0.x release of the Z3GatewayFreeRTOS Library and the cloud gateway application example source-codes.

KEY FEATURES

- Hardware and firmware preparations
- Source-code generation and compilation on amebaD sdk
- Flasing AmebaD application firmware on LCGW
- Runing example cloud gateway application

Table of Content

1	<i>Hardware and Firmware Preparations</i>	2
1.1	The Following Hardware are Needed	2
1.2	NCP Firmware Preparation	3
1.3	NCP Firmware Programming	3
2	<i>Build Environment Preparation and Source-code Compiling on AmebaD SDK</i>	5
2.1	Set up Linux Build Environment for AmebaD SDK	5
2.2	Generate and Compile Z3GatewayFreeRTOS Library Source Code Package on AmebaD SDK	6
3	<i>Flash the AmebaD Firmware Application</i>	8
3.1	Flash the AmebaD Application Firmware Binaries	8
4	<i>Run the Aliyun Gateway Example Application</i>	9
4.1	Download Alibaba Cloud Intelligence Mobile Phone App	9
4.2	Mobile Phone App Provision Wi-Fi to LCGW and Add it to Aliyun	9
4.3	Adding Zigbee Light to network and Aliyun	14
4.4	Adding Zigbee Switch to Network and Aliyun	18
4.5	Important Notice	22
4.6	Further Reference	24
5	<i>AmebaD AT commands and Z3GatewayFreeRTOS CLI commands</i>	25
5.1	AmebaD Console AT commands	25
5.2	Gateway CLI Console Commands	25
6	<i>Document Revision History</i>	27
	Revision 0.1	27
	Revision 0.2	27
	Revision 1.0	27
	Revision 1.0.1	27
	Revision 1.0.2	27

1 Hardware and Firmware Preparations

1.1 The Following Hardware are Needed

- LCGWv2 Kit



- LCGWv2 board jumpers:

Jumpers

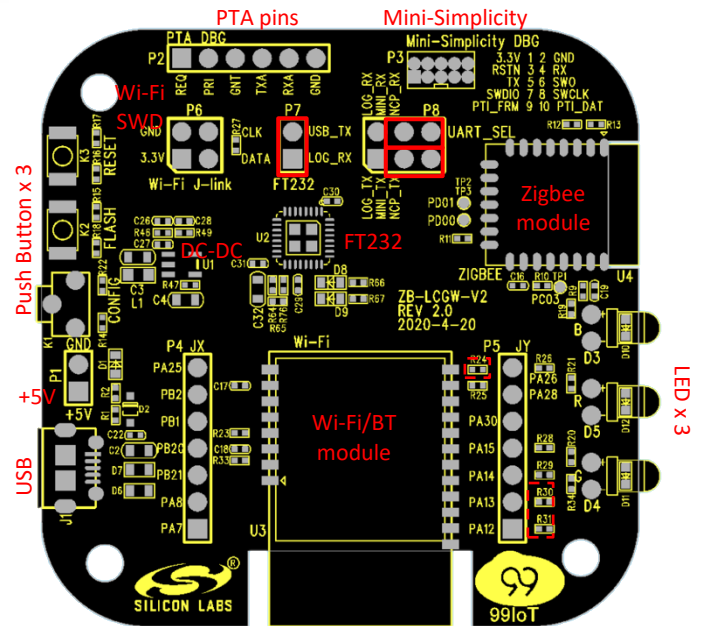
- P7: Default shorted to enable AmebaD log message and firmware upgrade over USB
- P8: Mini-simplicity TX, RX pins route to Zigbee module NCP or AmebaD log UART

Wi-Fi J-Link

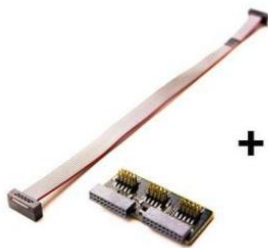
- Remove R24 to enable Wi-Fi SWD (disconnects SWD_CLK from NCP reset circuit)

Zigbee CLI over Mini-Simplicity header

- Remove R30, R31 to disconnect NCP UART from AmebaD host UART



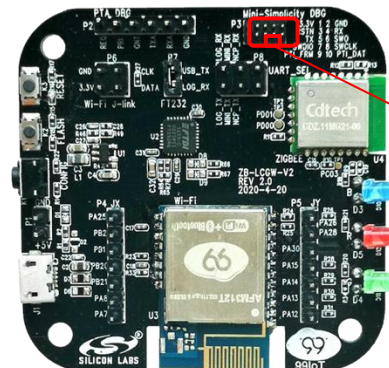
- WSTK BRD4001A Kit with BRD8010A debug adaptor (please refers to UG386: EFR32xG21 2.4 GHz 10 dBm Wireless Starter Kit User's Guide)



Simplicity Debug Adapter Board with 10-pin ribbon cable



Simplicity Debug Adapter Board connected to a Wireless Starter kit



10-pin Mini-Simplicity Header

1.2 NCP Firmware Preparation

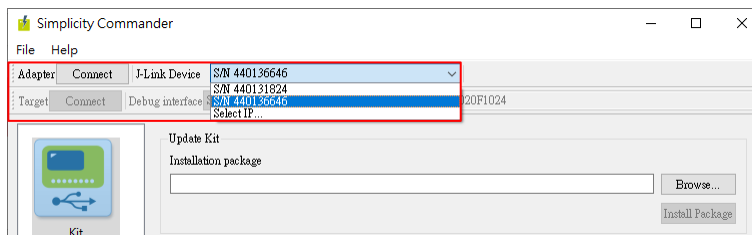
- Please refer to User Guide document of “*NCP and Bootloader Firmware Customization for LCGW*”

1.3 NCP Firmware Programming

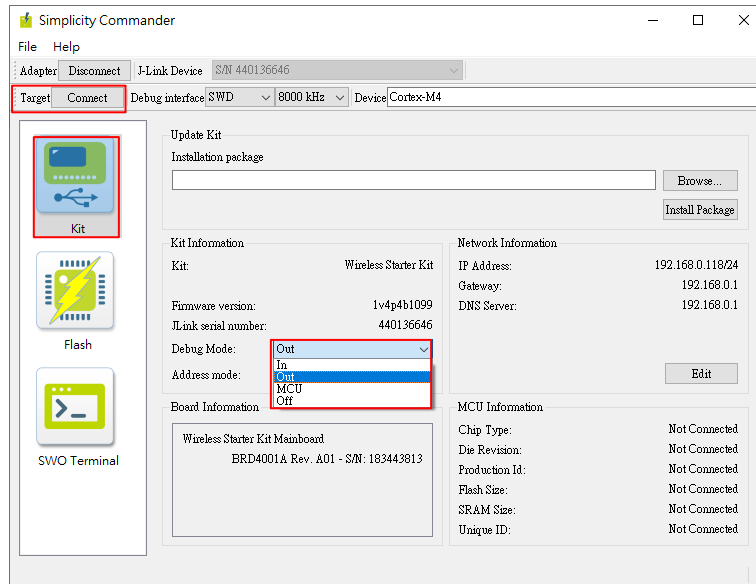
- There should be a factory firmware in the Zigbee NCP chip in the LCGWv2 Kit. To reprogram the firmware:
Note: Make sure Wi-Fi module Ameba firmware has been flashed (refer to section 3.1) before flashing the NCP firmware.
 - Open the bottom plaster cover of the LCGWv2 kit
 - Plug in BRD8010A debug adaptor on to WSTK BRD4001A
 - Connect the 10-pin mini-simplicity cable on BRD8010A debug adaptor
 - Connects the WSTK to a PC with latest Simplicity Studio/Simplicity Commander installed
 - Plug in the other end of mini-simplicity cable onto LCGWv2 P3 header (note for probe pin 1 orientation)



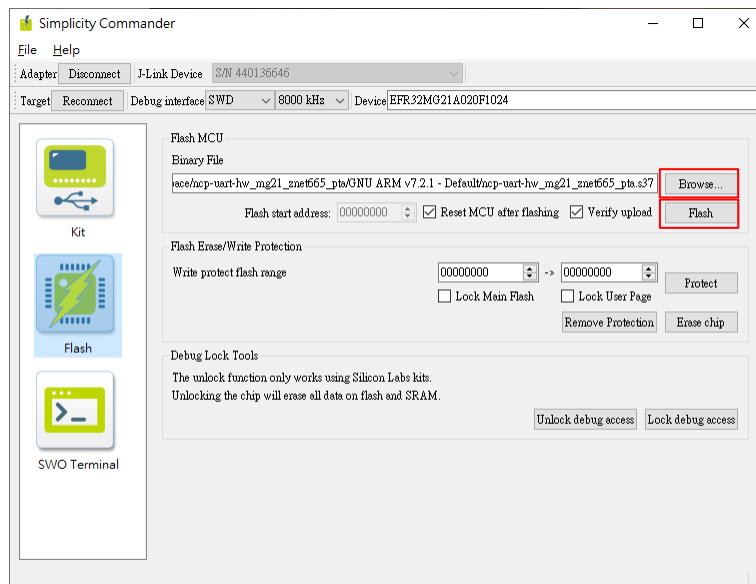
- Open the Simplicity Commander select from top “Adapter” J-Link Device your USB connected WSTK, click “Connect”



7. In “Kit” UI, change the debug mode of WSTK to “OUT” mode as below and then click the top “Target” “Connect”. Commander should be able to identify the target device to be “EFR32MG21A02F1024”.



8. Switch to “Flash” UI, browse to select the NCP .s37 file, e.g. ncp-hw-znet655_pta.s37, and then click “Flash” to flash the firmware binary. It is optional to click “Erase chip” first, browse and flash the Bootloader .s37 file then the NCP.s37 file sequentially onto the LCGWv2 HW.



2 Build Environment Preparation and Source-code Compiling on AmebaD SDK

There are two versions of Realtek's AmebaD FreeRTOS SDK, namely the public version and the NDA version:

- Public version: github on https://github.com/ambiot/ambd_sdk
- NDA version: Required signed NDA, please contact Realtek, its distributors or its IDHs for details

Silicon Labs' Z3GatewayFreeRTOS library and the Aliyun gateway example could be built on top of both the public version AmebaD SDK and the NDA version SDK. The difference is that the RTL872xCS/DN Wi-Fi and EFR32MG13/MG21 Zigbee 2.4GHz ISM band PTA coexistence is only available on the NDA version AmebaD SDK through a SW-PTA patch applied on it.

2.1 Set up Linux Build Environment for AmebaD SDK.

Public version AmebaD SDK:

- Git clone Realtek's public version AmebaD SDK repository to local folder, e.g. user root ~

```
$ cd ~
$ git clone https://github.com/ambiot/ambd_sdk.git
$ cd ambd_sdk
```

- Aliyun gateway application example in the LCGW v1.0.x GA release package is built on top of AmebaD SDK's latest commit d64c72293cd6d4b75f6b6f37eaa61376e1290e9c dated Jan 20, 2020, run below if your cloned repository is not the case:

```
$ git reset --hard d64c72293cd6d4b75f6b6f37eaa61376e1290e9c
```

- Fix some bash shell and python scripts whose executable mode were not set correctly when running on Linux:

```
$ chmod +x ~/ambd_sdk/project/realtek_amebaD_va0_example/GCC-RELEASE/project_lp/asdk/gnu_utility/prepend_header.sh
$ chmod +x ~/ambd_sdk/project/realtek_amebaD_va0_example/GCC-RELEASE/project_lp/asdk/gnu_utility/image_tool/imagetool.sh
$ chmod +x ~/ambd_sdk/project/realtek_amebaD_va0_example/GCC-RELEASE/project_lp/asdk/gnu_utility/pad.sh
$ chmod +x ~/ambd_sdk/project/realtek_amebaD_va0_example/GCC-RELEASE/project_hp/asdk/gnu_utility/prepend_header.sh
$ chmod +x ~/ambd_sdk/project/realtek_amebaD_va0_example/GCC-RELEASE/project_hp/asdk/gnu_utility/image_tool/imagetool.sh
$ chmod +x ~/ambd_sdk/project/realtek_amebaD_va0_example/GCC-RELEASE/project_hp/asdk/gnu_utility/pad.sh
$ chmod +x ~/ambd_sdk/project/realtek_amebaD_va0_example/GCC-RELEASE/project_hp/asdk/code_analyze.py
```

NDA version AmebaD SDK:

- Please contact Realtek, its distributor for the NDA version AmebaD SDK and related patches including the SW-PTA patch
- Extract Realtek official sdk-amebad_v6.2b SDK, i.e. "00016286-sdk-amebad_v6.2b-RC.tar.bz2" to a local folder, e.g. ~
- Apply the necessary AmebaD SDK patches to the SDK local folder, e.g. sdk-amebad_v6.2b, including
 - "00016574-v6.2b_intergrated_critical_patch_gcc_IAR_v39533.zip"
 - "00017141-v6.2b_critical_patch_gcc_IAR_v39533_v40238.zip"
 - "20200324_6.2b_other_patch_sw_pta_for_gongniu_r41263.zip" (← this is the SW-PTA patch required)

Reference material:

Please refer to Realtek AmebaD SDK document, AN0400 Ameba-D Application Note to compile and flash download:

<https://www.amebaiot.com/en/sdk-download-manual-8722dm/>

- We recommend Ubuntu 18.04 for Linux env development, it can be native Linux or a Virtual Machine Linux client, while Windows 10 WSL Linux is not recommended. BTW, Cywin 32-bit under Windows will also work but the build building time is much longer.
 - Note: You need to install 32-bit version of build packages on your Ubuntu 18.04 64-bit system as below

```
$ sudo apt-get update
$ sudo apt-get install build-essential gcc-multilib gcc-4.8-multilib g++-multilib
  g++-4.8-multilib lib32z1 lib32ncurses5 libc6-dev libgmp-dev libmpfr-dev libmpc-dev
$ sudo dpkg --add-architecture i386
```

- After AmebaD environment setup, it is good to build the default sample project `realtek_amebaD_va0_example` and flash the built binaries on to LCGWv2 kit to see if everything is ok before moving to the next step

2.2 Generate and Compile Z3GatewayFreeRTOS Library Source Code Package on AmebaD SDK

- Installation of Simplicity Studio v4 and the Gecko SDK on Linux
To build the z3 controller software in AmebaD, we need to start with Simplicity Studio Linux version, we can download the package from www.silabs.com

- Extract the downloaded SimplicityStudio-v4.tgz into a local folder
- Make sure the current user is a sudoer, go to the local folder and run setup:

```
$ cd SimplicityStudio_v4/
$ sudo ./setup.sh
```

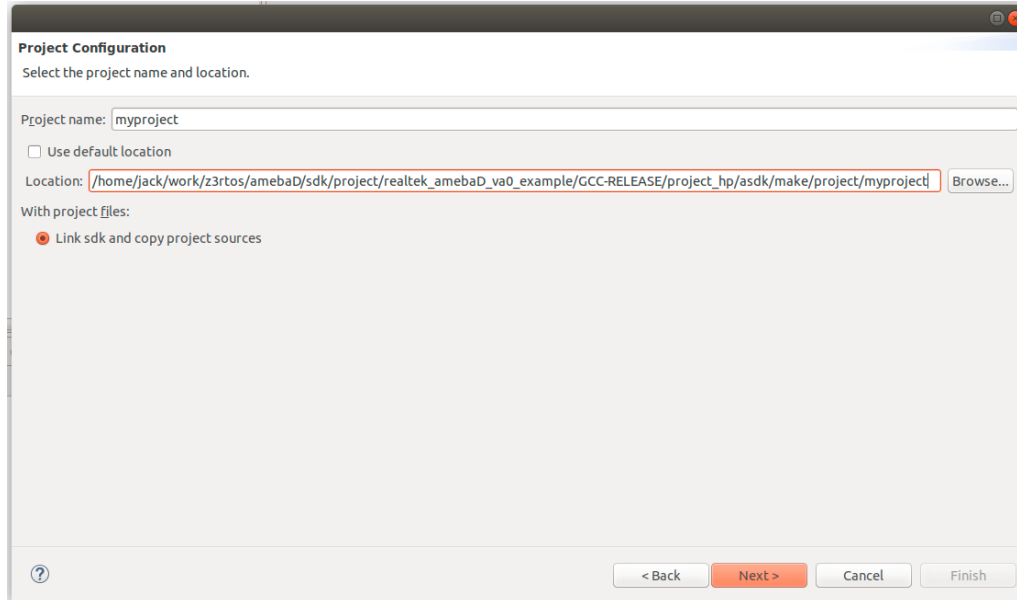
- After setup, you should have a Simplicity Studio icon in your application list:



- Click the Simplicity Studio icon to run the Studio
- You will be asked to agree the licenses, login, and then the software update process will auto start.
- When the installation manager pops up, choose Install by Product Group and then choose zigbee->next->choose packages (we recommend defaults)->agree all the licenses->finish
- It will start downloading packages, it's going to take a while depending on your internet speed.
- The released Z3GatewayFreeRTOS package depends on earlier ZNet SDK v6.6. If this doesn't match, please go to "Launcher" -> "Update Software" -> "Package Manger" -> "SDKs" -> select "Categories" = "EmberZNet SDK", "Version"="All" and pick "EmberZNet SDK – 6.6.5.0" to reinstall it.
- Now you are ready to do Zigbee development.
- Copy the provided gecko_sdk_znet_v6.6_patch.zip into Simplicity Studio install folder where the gecko SDK is located and unzip it. For example:
~/SimplicityStudio_v4/developer/sdks/gecko_sdk_suite/
- Merge the gecko SDK ZNet patch folder to the gecko-sdk-2.6, like:
\$ cp -rf ./gecko_sdk_znet_v6.6_patch/* ./v2.6/
- Import the patched SDK to Simplicity Studio, go to Windows->Preferences->Simplicity Studio->SDKs, then click Add, choose the patched v2.6 folder
- You should see a Gecko SDK Suite with version 6.6.5 is imported.

- Apply Alibaba iotkit-embedded (aka linkit) and cloud gateway example application patches to AmebaD SDK:
- **Note: If you're using NDA version AmebaD SDK, please contact Silicon Labs for Aliyun gateway example code specific to it.**
 1. Unzip the released `amebad_sdk_v6.2_aliyun_gw_example_patch.zip` source package to a local folder, e.g. `~`
 2. Apply the patches onto AmebaD SDK:
`$ cp -rf ./amebad_sdk_v6.2_aliyun_gw_example_patch/* ./ambd_sdk/`
- Creating Z3GatewayFreeRTOS project from Simplicity Studio
 1. In Simplicity Studio, new a project->select Silicon Labs AppBuilder Project->Select Silicon Labs Zigbee-> select EmberZnet 6.6.x GA Host->Z3GatewayFreeRTOS
 2. Name the project as "`lib_z3ctrl`" for the Aliyun cloud gateway application example. Alternatively, you may use the default project name (Z3GatewayFreeRTOS) or another project name for your own project. It is **REQUIRED** to select the project location at specific location in the AmebaD SDK project:

i.e. `<AmebaD SDK>/project/<AmebaD Project Name>/GCC-RELEASE/project_hp/asdk/make/project/`



3. After creating the project, click "Generate" on the main project page.
 4. Now you should see a project source tree generated under `project_hp/asdk/make/project/lib_z3ctrl/` folder.
- Build the AmebaD application firmware binaries
 Now you can build the whole project with reference to Realtek AN0400 AmebaD App Note. i.e.
 1. Build the `project_lp` KM0 project:
`$ cd ./ambd_sdk/project/realtek_amebaD_va0_example/GCC-RELEASE/project_lp`
`$ make all`
 2. Build the `project_hp` KM4 project:
`$ cd ./ambd_sdk/project/realtek_amebaD_va0_example/GCC-RELEASE/project_hp`
`$ make all`

3 Flash the AmebaD Firmware Application

3.1 Flash the AmebaD Application Firmware Binaries

- Realtek's Windows ImageTool.exe is recommended for flashing the AmebaD firmware to the LCGW HW.
Set LCGWv2 kit into Uart Download and Flashing mode:
 - Open the bottom plastic cover of the LCGWv2 kit
 - Plug the micro-USB cable to LCGWv2 and power it up from PC's USB port
 - Press and hold the "FLASH" button and then press the "RESET" button
 - Release the "FLASH" button, kit should be in flashing mode already
- Config ImageTool flash utility:
 - Select "AmebaD(8721D)" in "Chip Select", select the COM port corresponding to LCGWv2 kit.
 - Browse below flashing binaries and set their corresponding flashing addresses:

The Ameba application binaries:

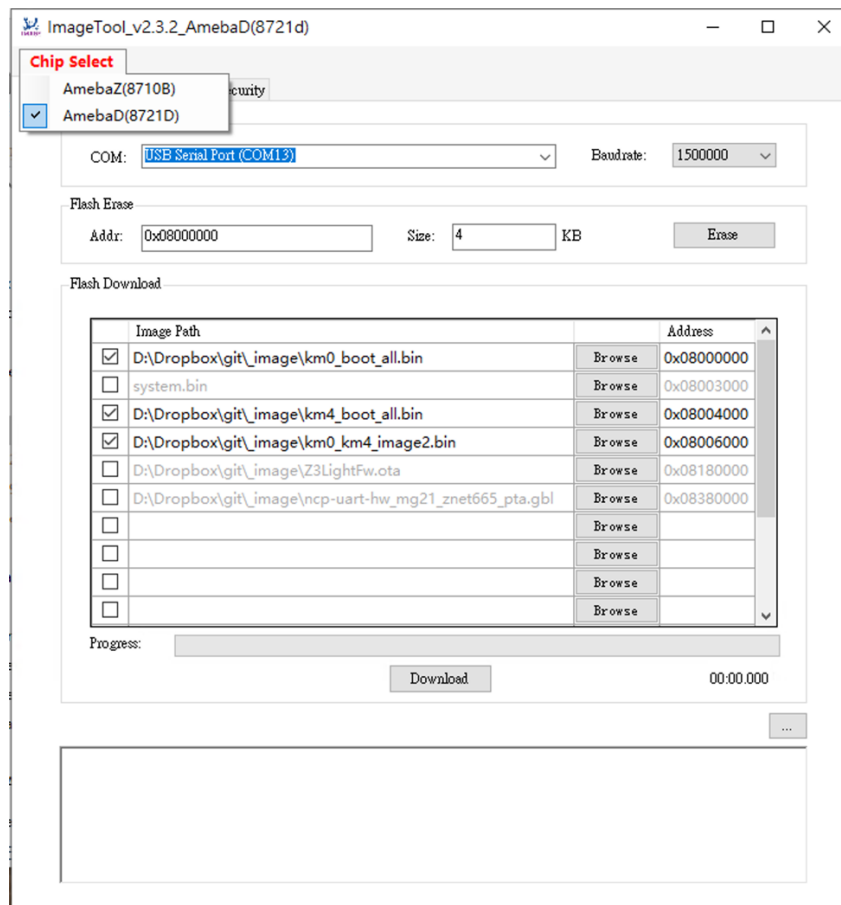
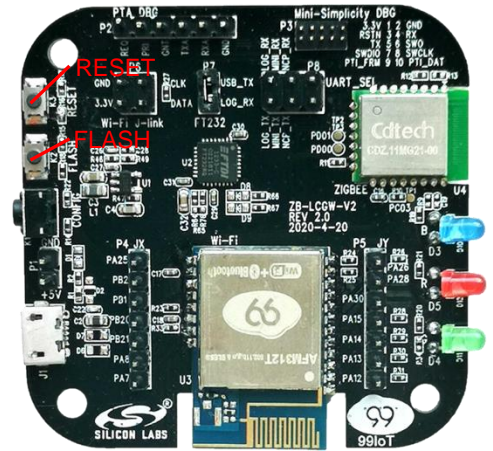
 - km0_boot_all.bin at 0x08000000
 - km4_boot_all.bin at 0x08004000
 - Km0_km4_image2.bin at 0x08006000

Optional for testing device OTA feature, we need to flash the .ota Z3Light image file:

 - Z3LightFw.ota at 0x08180000

Optional for testing device OTA feature, we need to flash the .gbl NCP image file:

 - ncp-uart-hw_mg21_znet665_pta.gbl at 0x08380000
 - Click the "Download" button and wait for the download and flashing complete.



4 Run the Aliyun Gateway Example Application

4.1 Download Alibaba Cloud Intelligence Mobile Phone App

- Download and install Alibaba “Cloud Intelligence” mobile App on your mobile phone by scanning below QR code:
Note: There are two mobile App editions, i.e. User edition (recommended) and Developer edition.



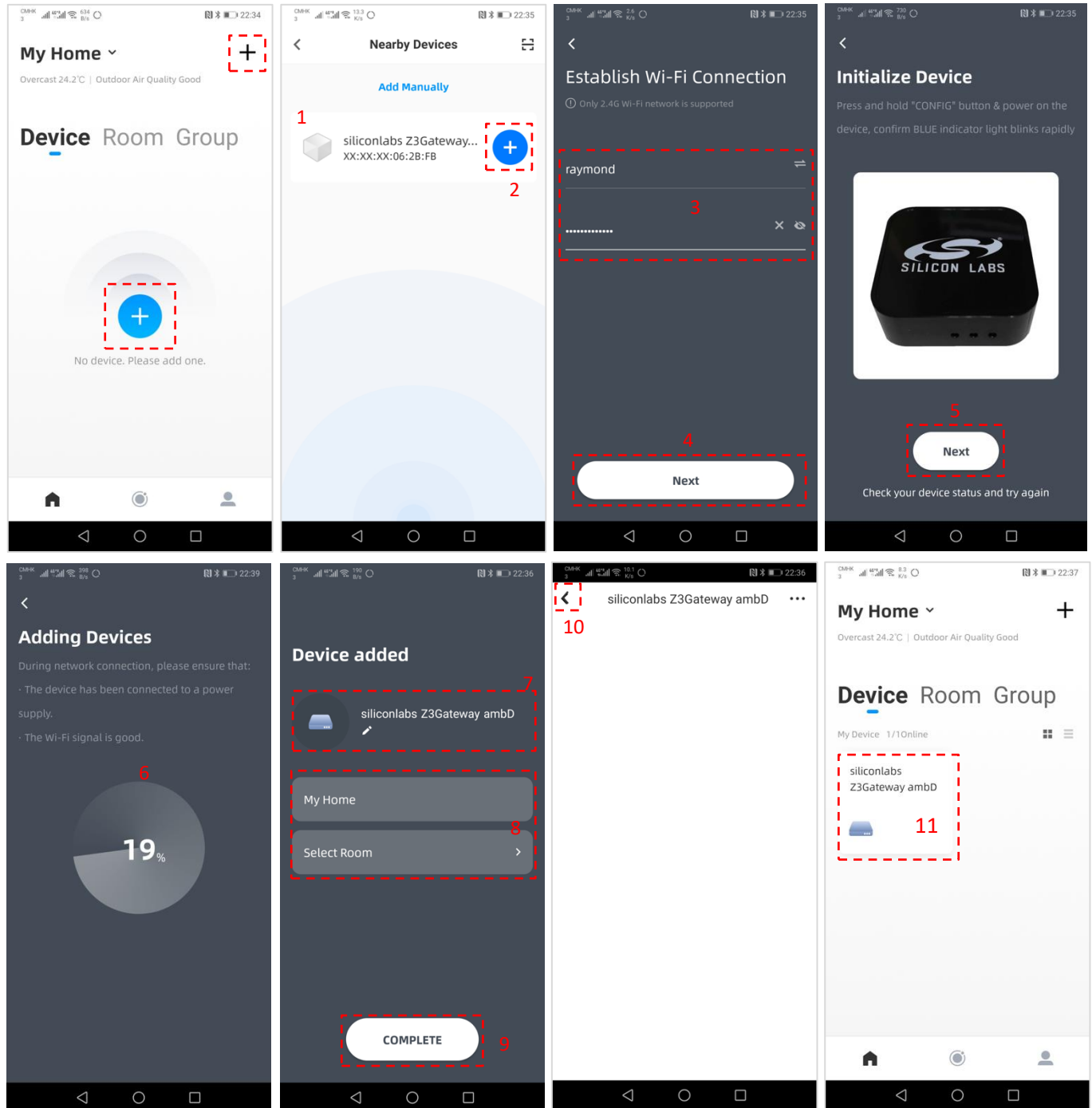
- There is a shared test account for the Cloud Intelligence App, please log in with below credential:
 - user name: silabs_tester@sina.com
 - password: Silabs2014
- After successfully logging in Cloud Intelligence App, you should be able to see the LCGW device icon but in grayed (offline) state, perhaps together with some other devices in grayed (offline) state. Or an empty device list on main page.
Note: Since this is a shared user account for testing purpose, anyone who logs in the Cloud Intelligence App with this account successfully will kick out any earlier logged in users using the same credential from their mobile phones.
- If the LCGW device icon isn't in grayed, i.e. it is activated and online currently, this means someone's LCGW hardware pre-flashed with the default demo triple-unit-group is still up and running. We need to make sure no other LCGW kits with factory firmware's default triple-element-groups of LCGW and certain devices are running concurrently.
Note: Two GW devices using the same triple-unit-group to register Aliyun will cause mal-functioning for both devices.

4.2 Mobile Phone App Provision Wi-Fi to LCGW and Add it to Aliyun

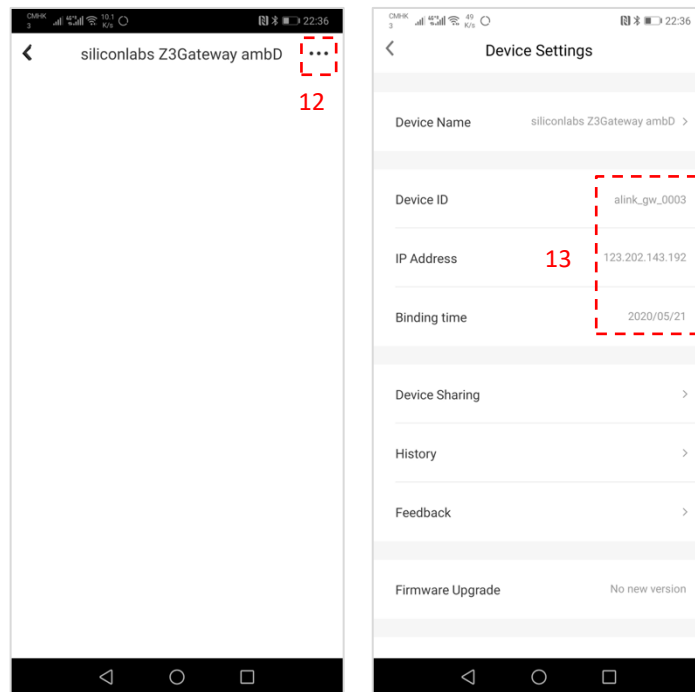
- Step 1: Get the LCGW into the Wi-Fi provisioning mode
 - Unplug the micro-USB cable from the LCCWv2's USB port
 - Use a toothpick or a ball pen tip to punch the hidden COFIG button at the back and hold the button
 - Re-plug the micro-USB cable to LCGWv2 to power it up from PC's USB port
 - Release the CONFIG button, Blue LED indicator is fast blinking as the indication of GW Wi-Fi provisioning mode



- Step 2: Connect your mobile phone to the Wi-Fi network that the LCGWv2 is targeted to be connected to internet
Note: The mobile phone Wi-Fi must be connected to a 2.4GHz Wi-Fi network, otherwise provisioning will fail.
- Step 3: Tap to run “Cloud Intelligence” App, tap the top-right “+” icon to scan for nearby Alibaba Wi-Fi/BT-mesh IoT devices...
 - On Android phones, LCGW will be discovered automatically...
 - “siliconlabs Z3Gateway ambD” appears, tap right-side blue “(+)” icon
 - Enter Wi-Fi SSID and Password
 - Tap “Next” button
 - You'll see GW Wi-Fi (BLUE) indicator fast blinking, tap “Next” button to continue...



6. You will see Adding Devices in progress...
7. Wait LCGW device added in less than 1 minute. Optional to rename GW.
8. Optional to rename the house and select room
9. Tap "Complete" button
10. In GW panel, tap "<" icon to return main page. See the LED indicators:
 - If GW connected to Aliyun: Cloud (**GREEN**) indicator turns on
 - Wi-Fi (**BLUE**) and Zigbee (**RED**) indicators restore to slow blinking
11. "siliconlabs Z3Gateway ambD" appears in the device list, tap device icon
12. The GW Control Panel is blank as nothing to control. Tap top-right "..." icon to view GW's settings page
13. Check GW's device ID, IP address, bonding time, etc. information



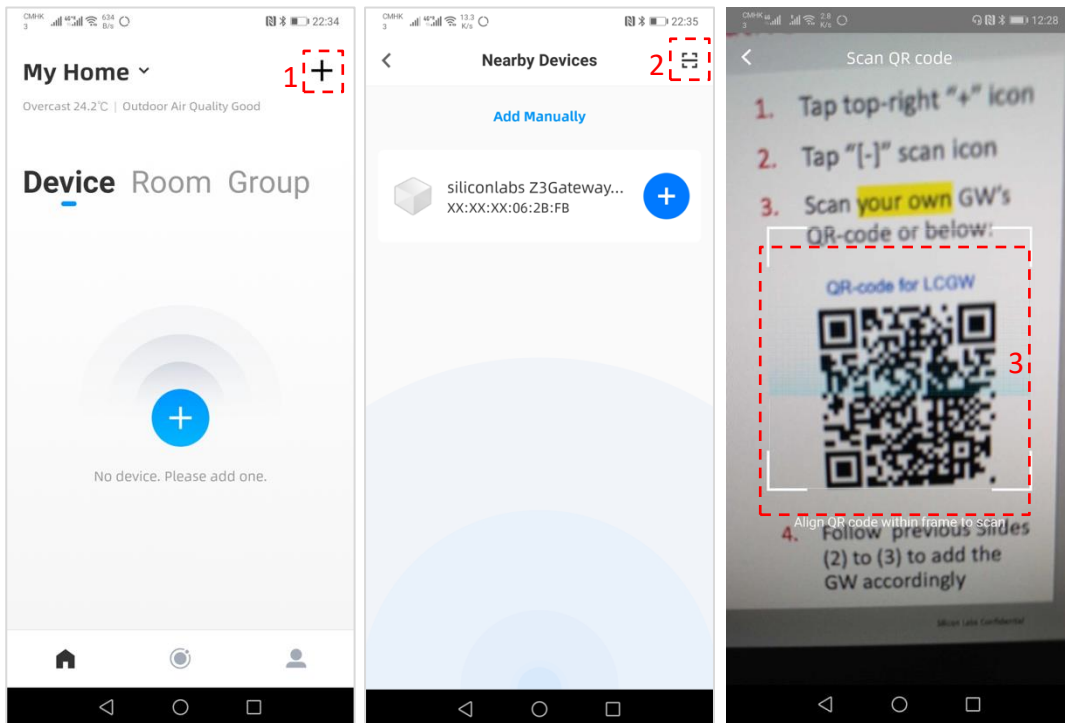
Alternative Method 1: Scan the LCGW's QR-code

1. Tap top-right "+" icon
2. Tap "[+]" scan icon
3. Scan LCGW's QR-code or below:

QR-code for LCGWv2



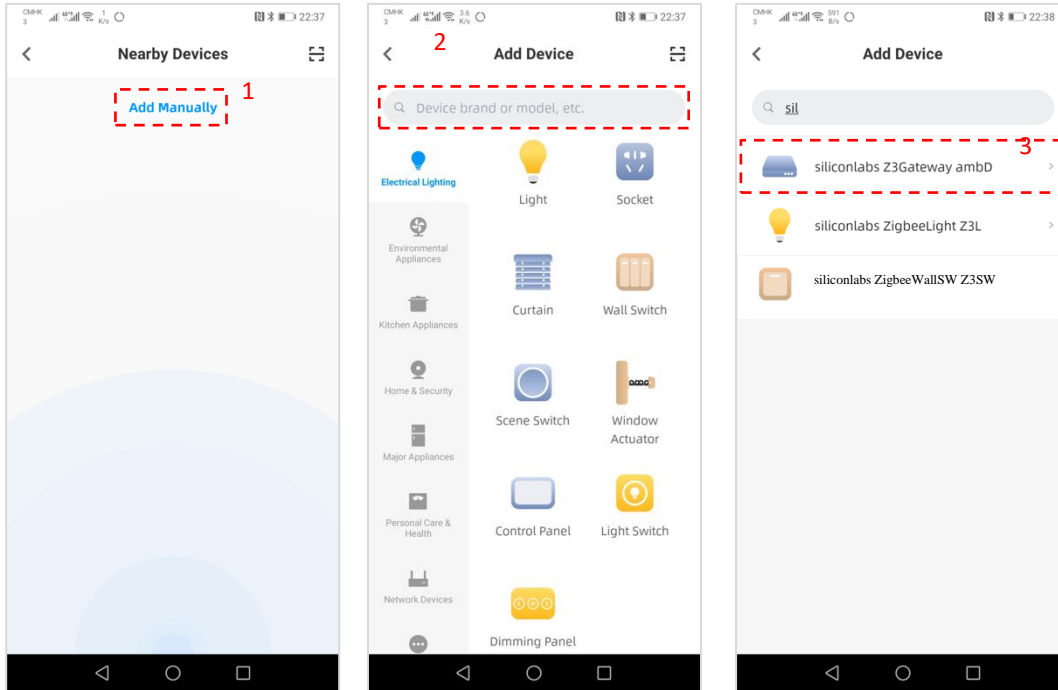
4. Follow sub-step (3) to (10) of previous Step 3 to add the LCGW accordingly



Alternative Method 2: Manually add the LCGW device

Note: Only the product that is in the Mass Production stage and whose name has passed the inspection of Aliyun platform will be shown in the Product List.

1. Tap top-middle “Add Manually” icon
2. In “Add Device” panel search box, enter “siliconlabs” (or your device name) to filter our devices
3. Tap from result listed “siliconlabs Z3Gateway ambD” (or your defined product) to add the GW
4. Follow sub-step (3) to (10) of previous Step 3 to add the LCGW accordingly



4.3 Adding Zigbee Light to network and Aliyun

Three different methods to add Zigbee sub-devices:

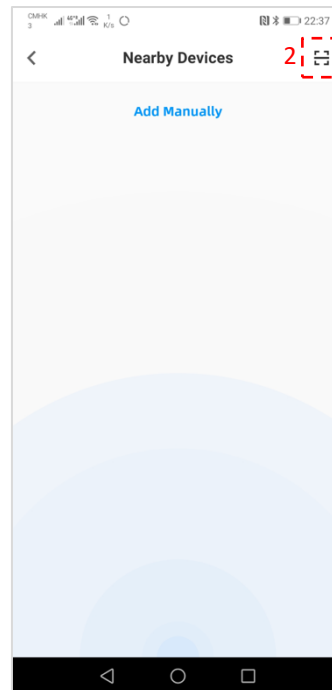
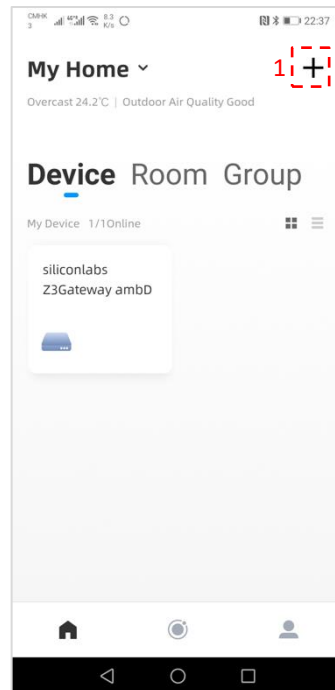
- (1) Scan device's product QR-code generated from living.aliyun.com
- (2) Manually select a device product from full product list
- (3) LCGW single CONFIG button to permit Zigbee device joining in 180s using default global link-key.

Note: Method 3 CONFIG button method requires the first device of a product has been added via QR-code successfully.

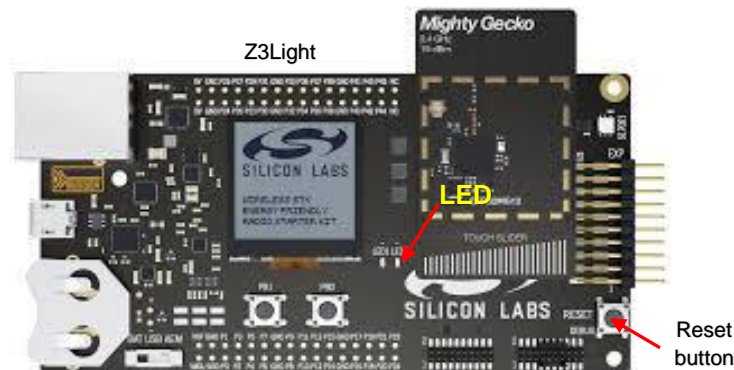
Method 1: Scan Zigbee Light's product QR-code

- Before adding a Zigbee device, the gateway should have automatically formed a Zigbee network.
- Step 1: Scan the device's QR-code by the Cloud Intelligence mobile phone App
 1. Tap top-right "+" icon
 2. Tap "[+]" scan icon
 3. Scan device's QR-code or below:

QR-code for Z3Light



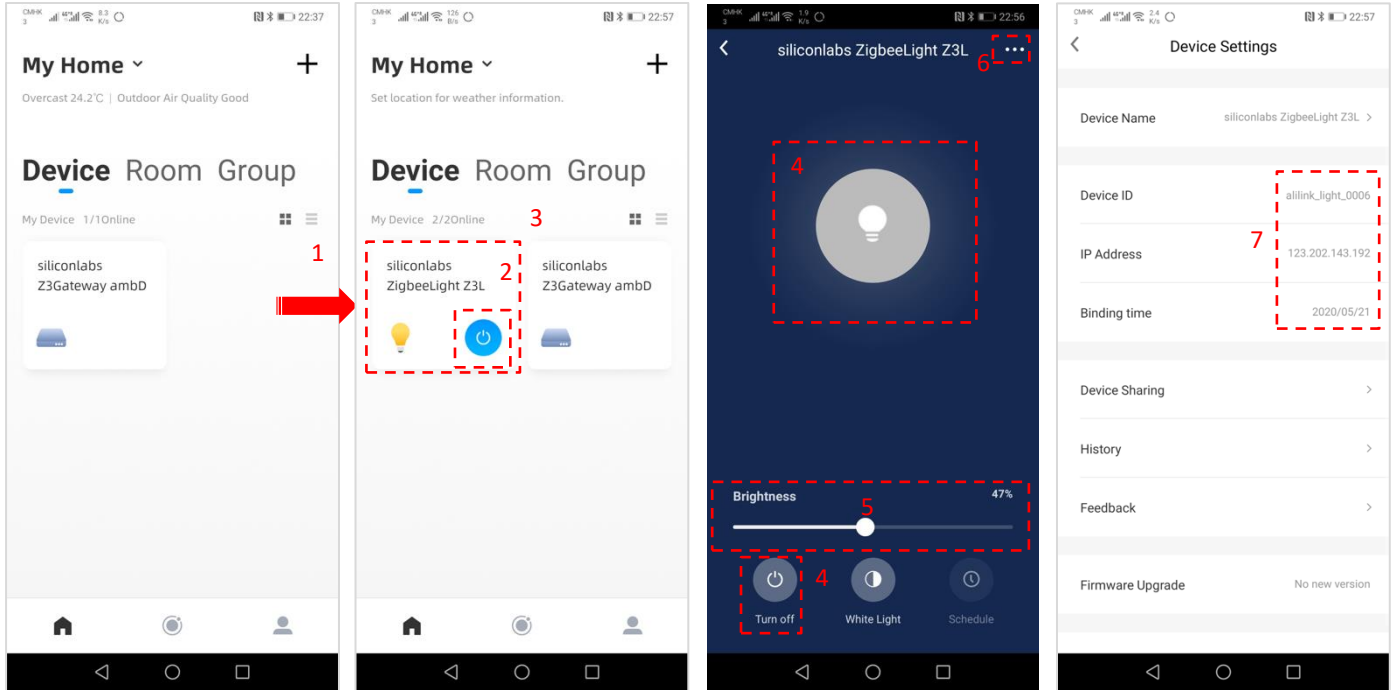
- Step 2: Run the Z3LightSoC FW on WSTK to scan Zigbee network:
 1. Connect WSTK with BRD4161/4162A radio board flashed with Z3LightSoC firmware to PC via USB
 2. In Simplicity Studio Debug Adapters, right click the Z3Light WSTK and select "launch Console"
 3. Go to "Serial 1" window, press "Enter" in terminal until you see the prompts, e.g. "Z3LightSoC>"
 4. Type "network leave" then press "Enter" to force Z3Light to leave network
 5. Press WSTK's Target RESET button, Z3Light will start searching network after reset
(One could check the logs in terminal to confirm device has joined network successfully.)



- Step 3: Add Z3Light device
 1. Tap "Add Device" button on the App
 2. You will see Adding Devices in progress and GW Zigbee (RED) indicator is fast blinking, 100ms on 100ms off. Wait till adding device process completes in less than 1 minute. Once the device joins successfully or the 180s expires, the Zigbee (RED) indicator returns back to slow blinking (100ms on 1900ms off)
 3. Tap "<" to return main page, or
 4. Tap "..." to view the Light settings page



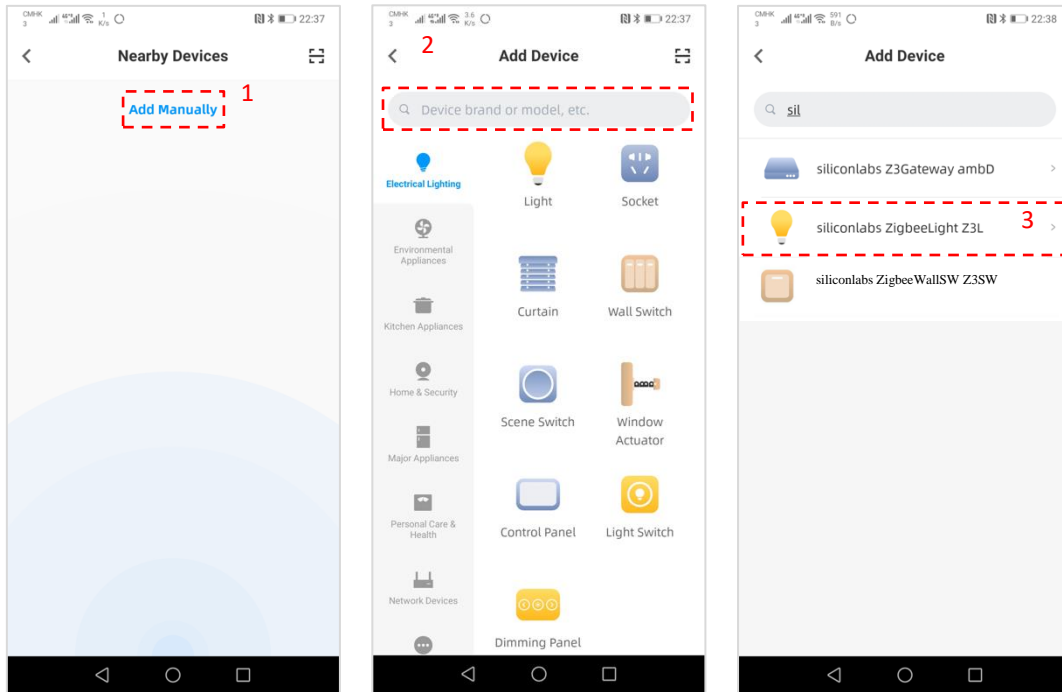
- Step 4: Z3Light Control and Settings
 1. After successful adding the device, it'll appear on the Device page of the App. (If not, please refresh the page.)
 2. Tap the "On/Off" button to toggle the Z3 Light on/off
 3. Tap "siliconlabs ZigbeeLight Z3L" icon itself to open the Light Control panel
 4. Tap middle "Bulb" icon or bottom "On/Off" button to toggle on/off
 5. Drag the Brightness bar to adjust brightness from 0 to 100% (Note that the WSTK's LED doesn't support the brightness adjustment.)
 6. Tap "..." to view "Device Settings" page
 7. In "Device Settings" page, check the device ID, IP address, bonding time, etc. information



Method 2: Manually add the Zigbee Light device

Note: Only the product that is in the Mass Production stage and whose name has passed the inspection of Aliyun platform will be shown in the Product List.

1. Tap top “Add Manually” icon
2. In “Add Device” panel search box, enter “siliconlabs” to filter our devices
3. Tap from result list “siliconlabs ZigbeeLight Z3L” to add the Zigbee Light
4. Follow previous Steps 2 to 4 to add the device accordingly and do the device controlling



Method 3: Single CONFIG button open network for Zigbee device joining

1. Follow previous Step 2 to force the device SoC on WSTK to leave network and scan Zigbee network
2. Set LCGWv2 to “Open” network for device joining:
 1. Use a ball pen tip to punch the hidden CONFIG button once, i.e. less than 500ms
 2. Release the CONFIG button, Zigbee (RED) indicator is fast blinking (100ms on 100ms off) as the indication
 3. Gateway “Permit Join” with default global link-key for 180s
3. Once the device joins successfully or the 180s expires, the Zigbee (RED) indicator returns back to slow blinking (100ms on 1900ms off)
4. If the device joins successfully, follow the previous Step 4 to do the device controlling.

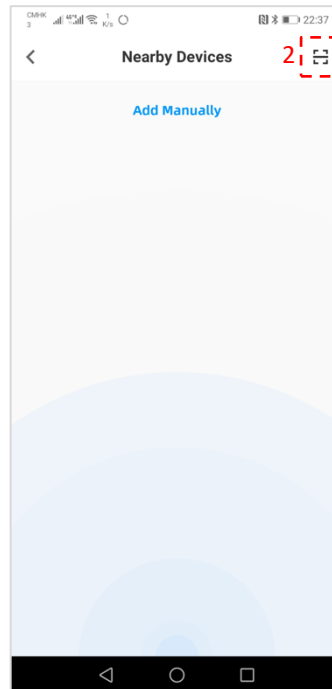
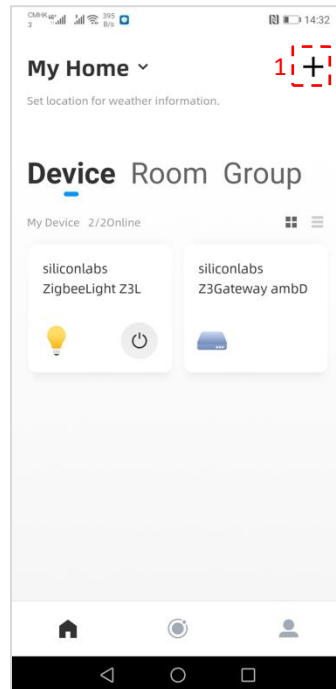


4.4 Adding Zigbee Switch to Network and Aliyun

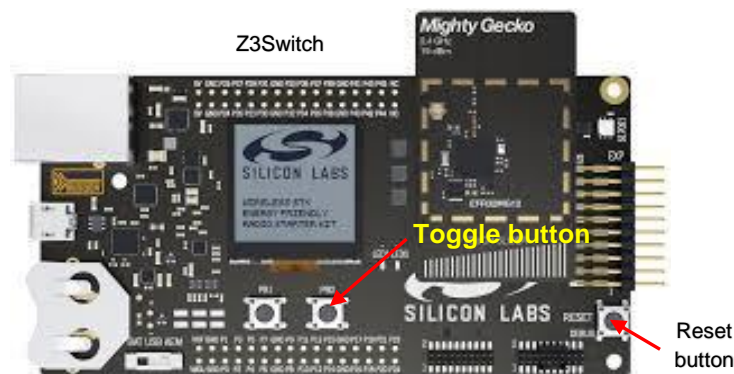
Method 1: Scan Zigbee Switch's product QR-code

- Before adding a Zigbee device, the gateway should have automatically formed a Zigbee network.
- Step 1: Scan the device's QR-code by the Cloud Intelligence mobile phone App
 - Tap top-right "+" icon
 - Tap "[+]" scan icon
 - Scan device's QR-code or below:

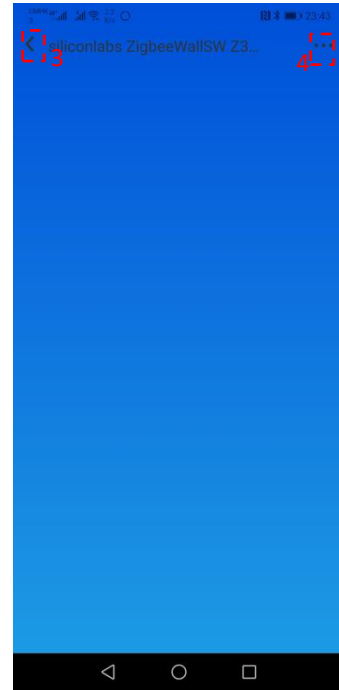
QR-code for Z3Switch



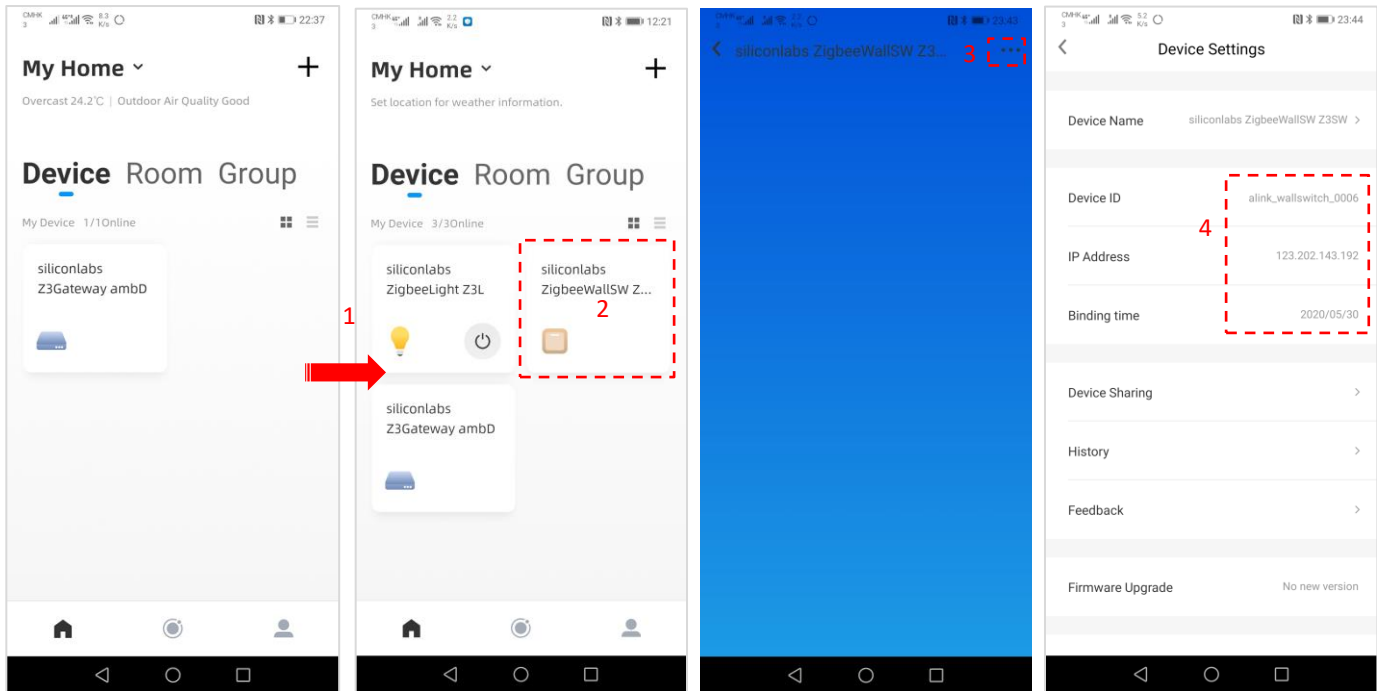
- Step 2: Run the Z3SwitchSoC FW on WSTK to scan Zigbee network:
 - Connect WSTK with BRD4161/4162A radio board flashed with Z3SwitchSoC firmware to PC via USB
 - In Simplicity Studio Debug Adapters, right click the Z3Switch WSTK and select "launch Console"
 - Go to "Serial 1" window, press "Enter" in terminal until you see the prompts, e.g. "Z3SwitchSoC>"
 - Type "**network leave**" then press "Enter" to force Z3Switch to leave network
 - Type "**plugin network-steering start 60**" then press "Enter" to command Z3Switch to start searching network (One could check the logs in terminal to confirm device has joined network successfully.)



- Step 3: Add the Z3Switch device
 1. Tap "Add Device" button
 2. You will see Adding Devices in progress and GW Zigbee (RED) indicator is fast blinking, 100ms on 100ms off. Wait till adding device process completes in less than 1 minute. Once the device joins successfully or the 180s expires, the Zigbee (RED) indicator returns back to slow blinking (100ms on 1900ms off)
 3. Tap "<" to return main page, or
 4. Tap "..." to view "Device Settings" page



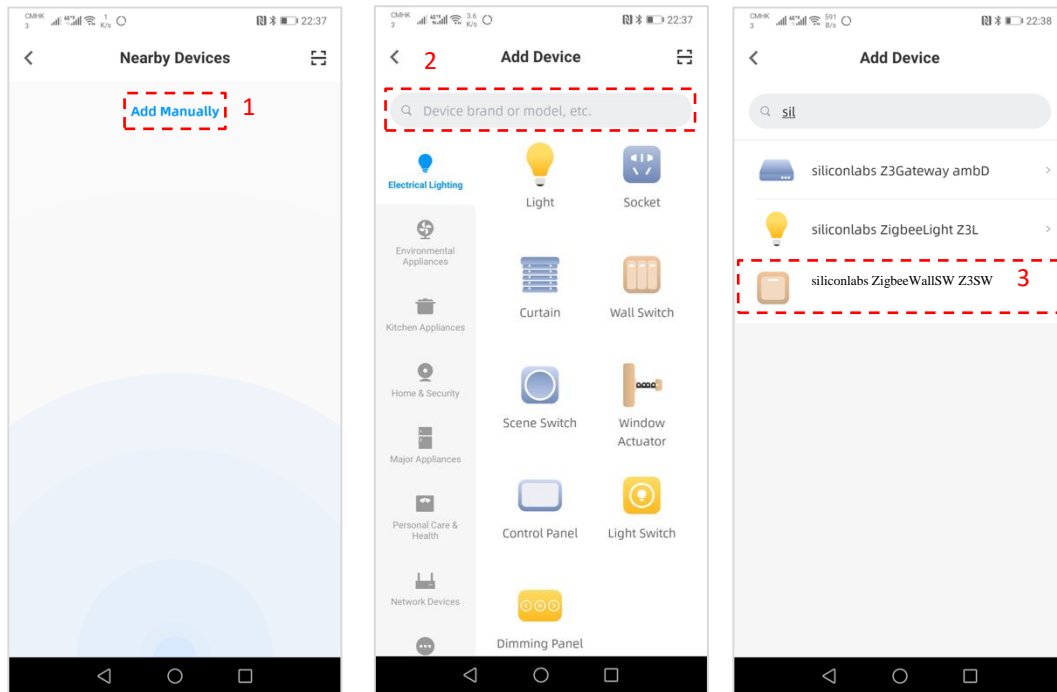
- Step 4: Z3Switch Settings
 1. After successful adding the device, it'll appear on the Device page of the App. (If not, please refresh the page.)
 2. Tap "siliconlabs ZigbeeWallSW Z3SW" icon itself to open the Switch Control panel
 3. Tap "..." to view "Device Settings" page
 4. In "Device Settings" page, check the device ID, IP address, bonding time, etc. information



Method 2: Manually add the Zigbee Switch device

Note: Only the product that is in the Mass Production stage and whose name has passed the inspection of Aliyun platform will be shown in the Product List.

1. Tap top “Add Manually” icon
2. In “Add Device” panel search box, enter “siliconlabs” to filter our devices
3. Tap from result list “siliconlabs ZigbeeWallSW Z3SW” to add the Zigbee Switch
4. Follow previous Steps 2 to 4 to add the device accordingly and do the device controlling



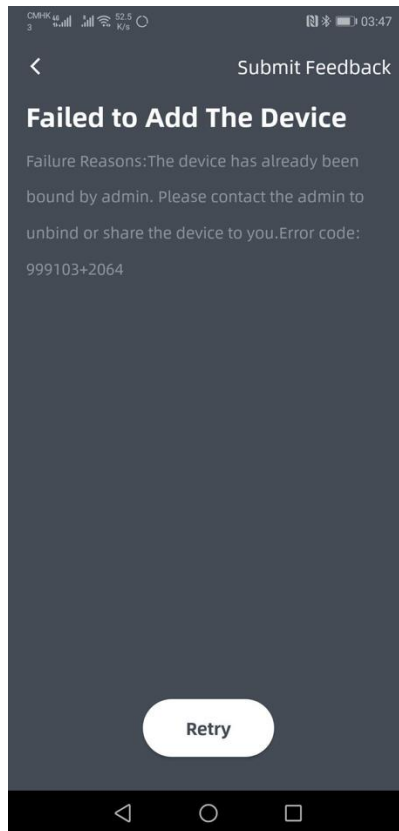
Method 3: Single CONFIG button open network for Zigbee device joining

1. Follow previous Step 2 to force the device SoC on WSTK to leave network and scan Zigbee network
2. Set LCGWv2 to “Open” network for device joining:
 1. Use a ball pen tip to punch the hidden CONFIG button once, i.e. less than 500ms
 2. Release the CONFIG button, Zigbee (RED) indicator is fast blinking (100ms on 100ms off) as the indication
 3. Gateway “Permit Join” with default global link-key for 180s
3. Once the device joins successfully or the 180s expires, the Zigbee (RED) indicator returns back to slow blinking (100ms on 1900ms off)
4. If the device joins successfully, follow the previous Step 4 to do the device controlling.



4.5 Important Notice

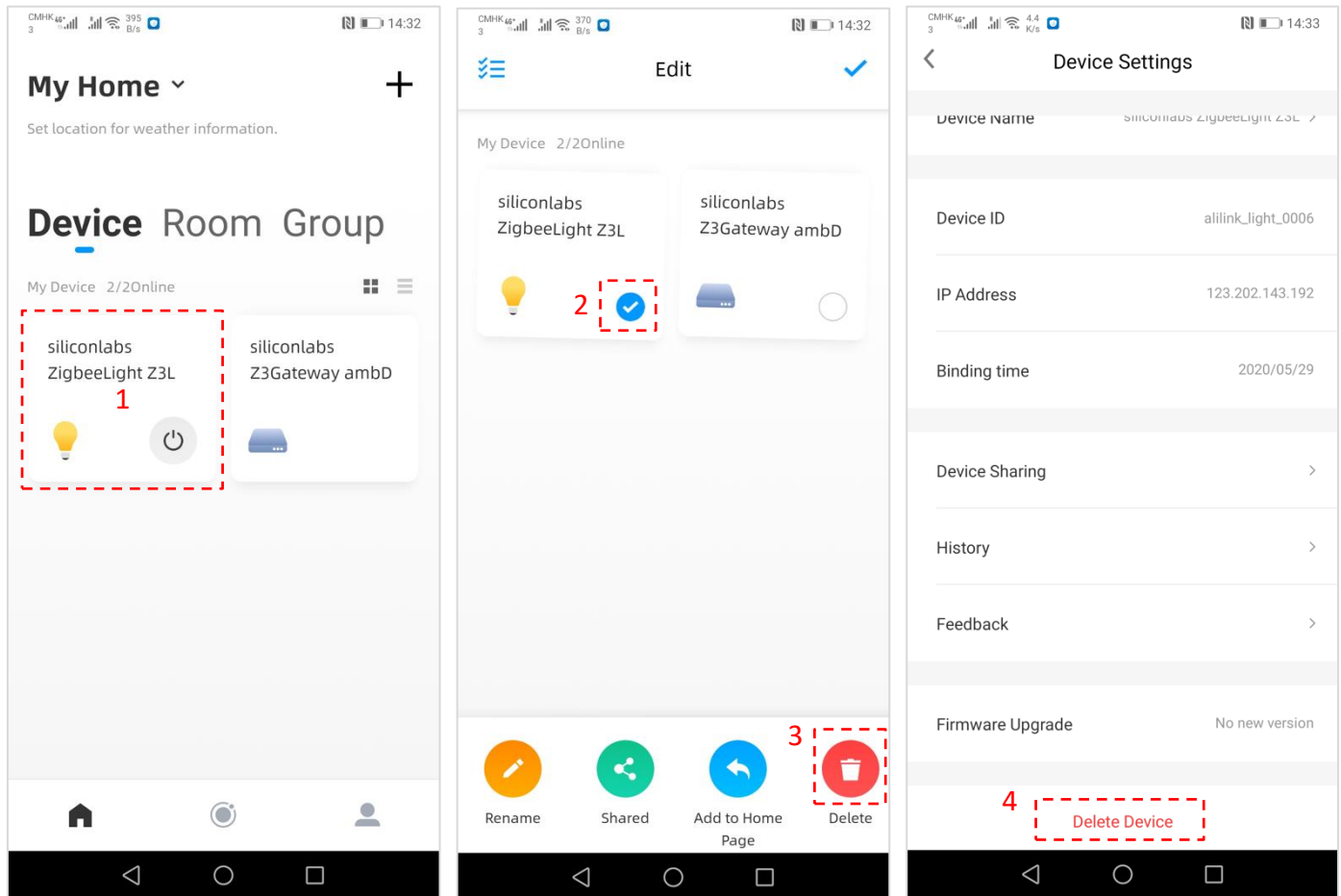
By using the public Cloud Intelligence mobile phone App, no matter the User edition or the Developer edition, once a device, including the gateway itself, is added (bound) to one user account, the other user account will be unable to add (bind) the same device again, i.e. the Triple-unit-group can't be bound unless it's unbound beforehand. See below error message on the mobile App.



- This is very important when the default pre-defined Triple-unit-groups of the gateway and test devices have been used in the gateway project sample code in a shared manner among multiple parties. To avoid conflicts of the default pre-defined Triple-unit-groups, a user should
 - use the shared test account below to log into the Cloud Intelligence App,
 - username: silabs_tester@sina.com
 - password: Silabs2014
 - or use his/her own device Triple-unit-groups in the sample code and log into the Cloud Intelligence App by his/her personal account. (Recommended)
- If one does add the LCGW with the default pre-defined Triple-unit-groups into the App logged in with other account rather than the shared test account, one should unbind the LCGW and the underlying devices which are associating with the default pre-defined Triple-unit-groups after his/her evaluation in the mobile phone App. The following will show how to unbind a device.
 1. Tap and hold a device from the main page device list to switch to the Edit mode
 2. Tap the “Radio” button(s) to select the device(s) for next operation
 3. Tap the bottom “Delete” icon button to delete the device, i.e. unbind the device from the mobile phone App

Alternatively, user may tap a device icon in the main page device list and go to its Device Control panel, then tap the top-right “...” icon to open its Device Settings panel, then

4. Scroll down to the very bottom, tap “Delete Device” button and tap again “Delete Device” in the confirmation dialog



The device then will be unbound to the mobile App so that the associated Triple-unit-group is free to be used by other user accounts. After unbinding all the underlying Zigbee devices, we need to unbind the gateway itself too in a similar manner to free its associated Triple-unit-group.

4.6 Further Reference

- Please refer to user guide document “[ug_Aliyun_iotkit_integration_of_Z3GatewayFreeRTOS_for_LCGW](#)” for further gateway and device operation details for Aliyun.

5 AmebaD AT commands and Z3GatewayFreeRTOS CLI commands

5.1 AmebaD Console AT commands

- Below Realtek AmebaD AT command are used to scan, configure and connect the GW to a Wi-Fi network:

```
#help
WLAN AT COMMAND SET:
=====
1. Wlan Scan for Network Access Point
   # ATWS
2. Connect to an AES AP
   # ATW0=SSID
   # ATW1=PASSPHRASE
   # ATWC
3. Create an AES AP
   # ATW3=SSID
   # ATW4=PASSPHRASE
   # ATW5=CHANNEL
   # ATWA
4. Ping
   # ATWI=xxx.xxx.xxx.xxx
```

5.2 Gateway CLI Console Commands

- To enable full functions of Z3GatewayFreeRTOS CLI functions and disable the cloud features, besides keeping **#define LOGSEV_ZIGBEE_CLI_CMD 1**, customers are required to modify the following macros in `inc_hp/platform_opts.h` and rebuild an image.
#define CLOUD_GATEWAY_ENABLE from 1 to 0
#define LOGSEV_PROCESS_UP_MSG from 0 to 1

Note: Only a limited set of CLI commands are available if “#define LOGSEV_PROCESS_UP_MSG” is kept to 0.

For running the Aliyun Gateway example application in Chapter 4, **LOGSEV_PROCESS_UP_MSG** is required to be set to 0.

- Sample Z3GatewayFreeRTOS CLI functions:
Please be aware that these CLIs are for testing purpose and could be changed without any prior notice, so are not recommended for customers building any product features on top of the CLI commands.

The implementation of the CLI functions is also a demonstration of the usage of standard API, i.e. `silabs-host-cmd.h` (please refers to “API reference Guide on Z3GatewayFreeRTOS”). We make use of the original AmebaD console service(`log_service.c`) and added extra demo codes to implement a series of functions with `silabs-host-cmd.h`. This is by default disabled, to Enable it, change “#define LOGSEV_PROCESS_UP_MSG” to 1, so that the message will diverted to `log_service` rather than the cloud interface layer. the CLI commands are with suffix “z2 <commands> <arg>..<arg>”

```
#z2 ?
print
create
leave
open
close
cmd
zcl-send
del-entry
set-assn
clr-assn
get-nwk
report
read
write
ota-reload
ota-notify
dev-leave
```

ncpupdate

Please refer to user guide document “[ug_Z3GatewayFreeRTOS_app_development_on_AmebaD_SDK](#)” for these sample CLI commands details.

Command examples for basic Zigbee network management and adding Zigbee devices:

- `z2 leave`
gateway leaves Zigbee network and delete all nodes info
- `z2 create`
gateway creates a new Zigbee network with random parameters
- `z2 open`
- `z2 open <allowed device eui> <install code>`
gateway opens Zigbee network for device joining with default global link key (no argument) or install code (with arguments)
- `z2 close`
gateway close network for device joining
- `z2 print`
print all devices endpoints with indexes which are used other sample CLI commands to specify a particular device/endpoint
- `z2 cmd u <devIdx> 0x<2 byte: clusterId> 0x<1 byte: cmdId> <1 byte:dataLen> 0x<data0> .. 0x<data n>`
unicast send a ZCL command to a device endpoint specified by its device index
- `z2 zcl-send u <devIdx> <endpoint> 0x<2 byte: clusterId> <dataLen> 0x<data0> .. 0x<data n>`
unicast send a raw ZCL messages to a device endpoint specified by its device index
- `z3 zcl on-off <on/off>`
`z3 send <nodeId> 1 1`
unicast send a zcl on-off message to a device endpoint specified by its node id using original ZNET gateway CLI functions.

- Original ZNET gateway CLI functions.

The original CLI function is retained in console in through `log_service.c` as well, start with “`z3[sp]`” will perform the znet CLI commands like the following:

```
#z3 ?
No such command
Usage:
<int>: 123 or 0x1ABC
<string>: "foo" or {0A 1B 2C}
bsend - Send using a binding based on the clusterId in the globalApsFrame and ...
changekey... -
debugprint... -
echo - Turns echo on the command line on or off depending on the argument
endpoint... -
events - Print the list of timer events.
help - Prints out the cli command options for the device
info - Gives information about the local node
interpan... -
keys... -
libs - Lists which optional libraries of the stack are implemented on this de ...
network... -
option... -
plugin... -
print... -
raw - Creates a message by specifying the raw byte ...
read - Read an attribute from the local attribute tabl ...
reset - resets the device
security... -
send - Send a pre-buffered message from a given endpoint to an endpoint on a ...
send-using-multicast-binding - When sending using a binding, specify whether a multicast
binding shou ...
send_multicast - Send a pre-buffered multicast message to a given group id from a given ...
timesync - This sends a read attr for the time of the device specifie ...
version - Shows the version of the software
write - Write an attribute value into the local attribute table
zcl... -
zdo... -
custom... - Custom commands defined by the developer
```

6 Document Revision History

Revision 0.1

April 22, 2020

EmberZNet PRO version: 6.6.5.0

- Initial release with LCGW v1.0 RC1 software package

Revision 0.2

May 6, 2020

EmberZNet PRO version: 6.6.5.0

- Release for LCGW internal training
- 2.1 added Ubuntu build environment to install 32-bit version of build packages
- Enriched 1.3 NCP firmware flashing instructions

Revision 1.0

May 31, 2020

EmberZNet PRO version: 6.6.5.0

- Released with LCGW v1.0 GA software package
- Added document revision history
- Updated latest LCGW kit recent photos in 1.1, 1.3 and 3.1
- Enriched 4.2 and 4.3 LCGW and Z3Light operations add more screenshots details
- Added 4.4 Z3Switch operations

Revision 1.0.1

Jun 17, 2020

EmberZNet PRO version: 6.6.5.0

- Revised 2.1 LCGW AmebaD firmware build steps
- Moved 3.2 and 3.3 AmebaD AT and Gateway CLI commands into new 5.1 and 5.2

Revision 1.0.2

Jun. 23, 2020

- Added 3.1 App Configuration
- Revised 4.5 Important Notice
- Minor formatting and revise

- End of Document -