



# Zigbee 3.0 RTOS 网关培训课程

[DEVELOPER SERVICES] | [JUN 2020]

R0.6



# 课程内容（第一天）

1. 简介：给客户带来的好处 [10分钟]
2. Z3 LCGW硬件设计 [15分钟]
  - LCGWv2 PCB、原理图，Wi-Fi和Zigbee模块
3. RTOS网关软件设计 [30分钟]
  - 软件功能、软件架构、LCGW软件实现
  - LCGW示例：代码、内存、任务和闪存摘要
  - HAL/OS移植摘要
  - Z3GatewayFreeRTOS库、iotkit-embedded库和上层网关应用
4. 设备OTA和NCP更新概述 [15分钟]
5. [休息：10分钟]
6. 编译和烧写LCGW固件 [30分钟]
  - 安装Realtek AmebaD FreeRTOS SDK
  - 安装Silicon Labs Simplicity Studio和EmberZNet SDK
  - 生成Z3GatewayFreeRTOS示例项目
  - 编译阿里云网关应用示例和刷新固件
7. [问答：10分钟]

Silicon Labs  
LCGWv2 套件



# 课程内容（第二天）

1. Wi-Fi和Zigbee共存 [30分钟]
  - 共存概述 · Wi-Fi+Zigbee共存策略
  - LCGW Wi-Fi、Zigbee频道管理
  - LCGW NCP固件PTA配置、LCGWv2 PTA信号探测
  - LCGW-PTA共存测试及测试结果
2. RTOS网关集成云 [40分钟]
  - Wi-Fi设备配网概述
  - 阿里云iotkit-embedded及Cloud Gateway应用概述
  - 阿里云上为LCGW、Zigbee灯和开关创建自己的物联网产品
  - 云智能手机应用：LCGW、灯和开关的智能自动化例子
  - 天猫精灵控制LCGW的Zigbee子设备
3. LCGW软件质量保证 [10分钟]
  - LCGW SQA测试
  - Mesh测试网及GW相关测试设施
4. [休息：10分钟]
5. 演示 [30分钟]
  - 生活物联网平台演示 (手机应用)
6. [问答：10分钟]

Silicon Labs  
LCGWv2 套件



# Zigbee 3.0 RTOS网关介绍

WWW.SILABS.COM



# 给客户带来的好处

## ■ 久经考验的Zigbee网关解决方案

- Silicon Labs在Linux的NCP模式网关架构是久经验证的Zigbee网关解决方案。
- 相同的NCP模式软件移植到RTOS主机上，以确保客户开发体验一致。

## ■ 设计简单和低成本硬件

- 适用于使用Wi-Fi MCU + Zigbee NCP的Zigbee集线器或网桥应用。
- 体积小、硬件设计简单，无散热问题。

## ■ 预集成PTA共存

- 在Realtek RTL872xCS/DN主机上实现了Wi-Fi、BLE和Silicon Labs ER32 Zigbee的3线PTA共存。

## ■ 更完整的Zigbee网关软件解决方案

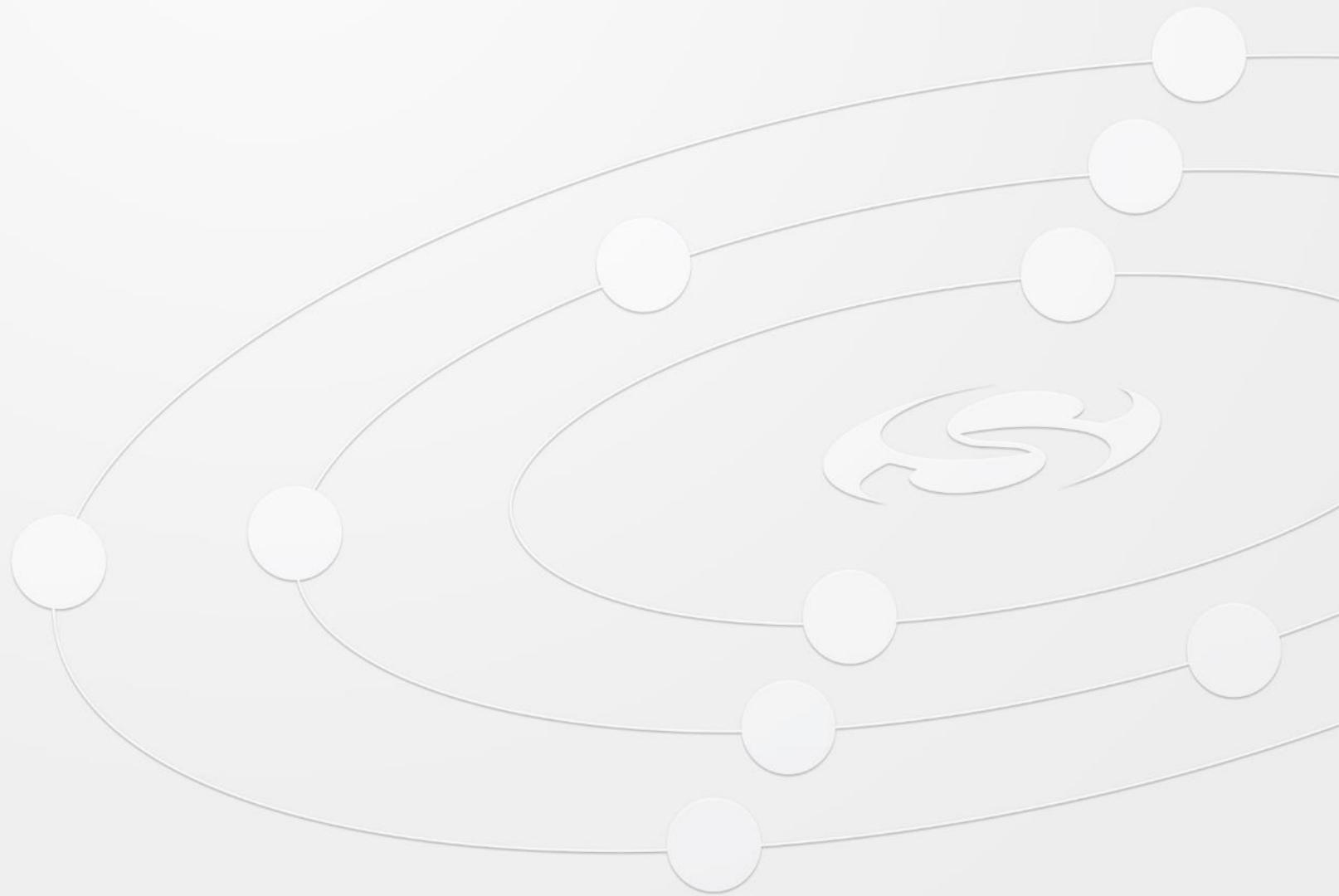
- EmberZNet SDK上的Z3GatewayFreeRTOS网关应用示例与阿里云集成示例。
- 分层网关软件结构允许轻松替代主机RTOS内核和集成云SDK。

## ■ 加快客户产品上市时间

- Z3GatewayFreeRTOS软件在LCGW上已通过SQA功能和大型网络测试，软件质量得到充分验证。
- 客户可以专注于他们核心的上层应用用例开发。
- Silicon Labs提供完整而深入的LCGW网关应用用户指南文档。
- 为中国重要客户提供专业的本地开发服务和技术支持。

# LCGW硬件设计

WWW.SILABS.COM



# Z3 LCGWv1(第1代)硬件设计 (2019.7-12)

## ■ Wi-Fi MCU主机

- 99物联AFM203T (RTL8720CM) Wi-Fi/BT4组合模块
- 瑞昱AmebaZII RTL8720CM (100MHz KM4, 4MB闪存, 256KB+4MB内存)

## ■ Zigbee NCP

- Silicon Labs基于EFR32MG21的MGM210P模块 (+19.5dBm输出功率)
- [EFR32MG21A020F1024IM32](#) (80MHz Cortex-M33, 1MB闪存, 96KB内存)

## ■ 硬件设计亮点

- 紧凑的插头式网关工业外型设计
- 配有Silicon Labs现成的塑料外壳
- 内置通用交直流电源模块 (5V/700mA输出)
- PWM驱动的红/绿和蓝两个LED指示灯
- 配置按钮，位于外壳内的固件更新和复位按钮
- Wi-Fi/BLE模块J-Link和日志串口的6针2.54mm连接器
- Zigbee NCP模块10针Mini-simplicity 1.27mm连接器(SWD, PTI)
- 串口转USB适配器：用于通过USB更新Wi-Fi应用固件

**注：LCGWv1未有实现PTA共存**



# Z3 LCGWv2(第2代)硬件设计 (2020.1-5)

## ■ Wi-Fi MCU主机

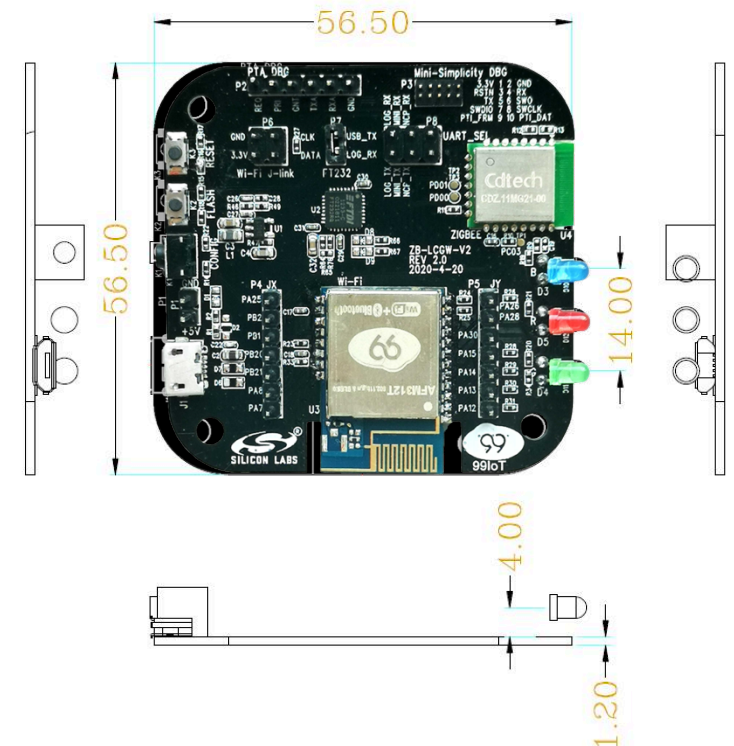
- 99物联AFM312T(RTL8720CS)/AMF412T(RTL8720DN) Wi-Fi/BT5组合模块
- 瑞昱AmebaCS/D (200MHz KM4/KM0双核, 512KB内存, 4MB闪存)

## ■ Zigbee NCP

- 中龙通CDZ-N7EFR32-00 (EFR32MG21) Zigbee模块(+19.5dBm输出功率)
- [EFR32MG21A020F1024IM32](#) (80MHz Cortex-M33, 1MB闪存, 96KB内存)

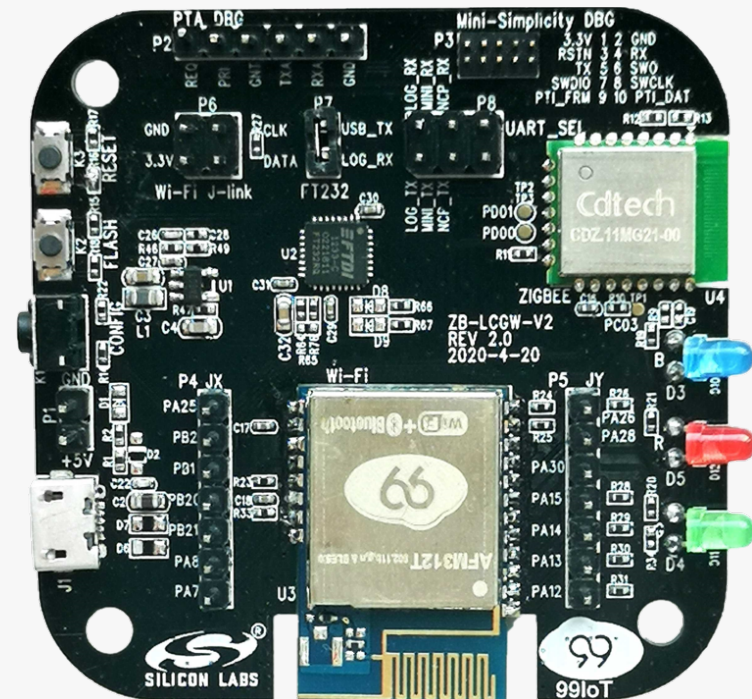
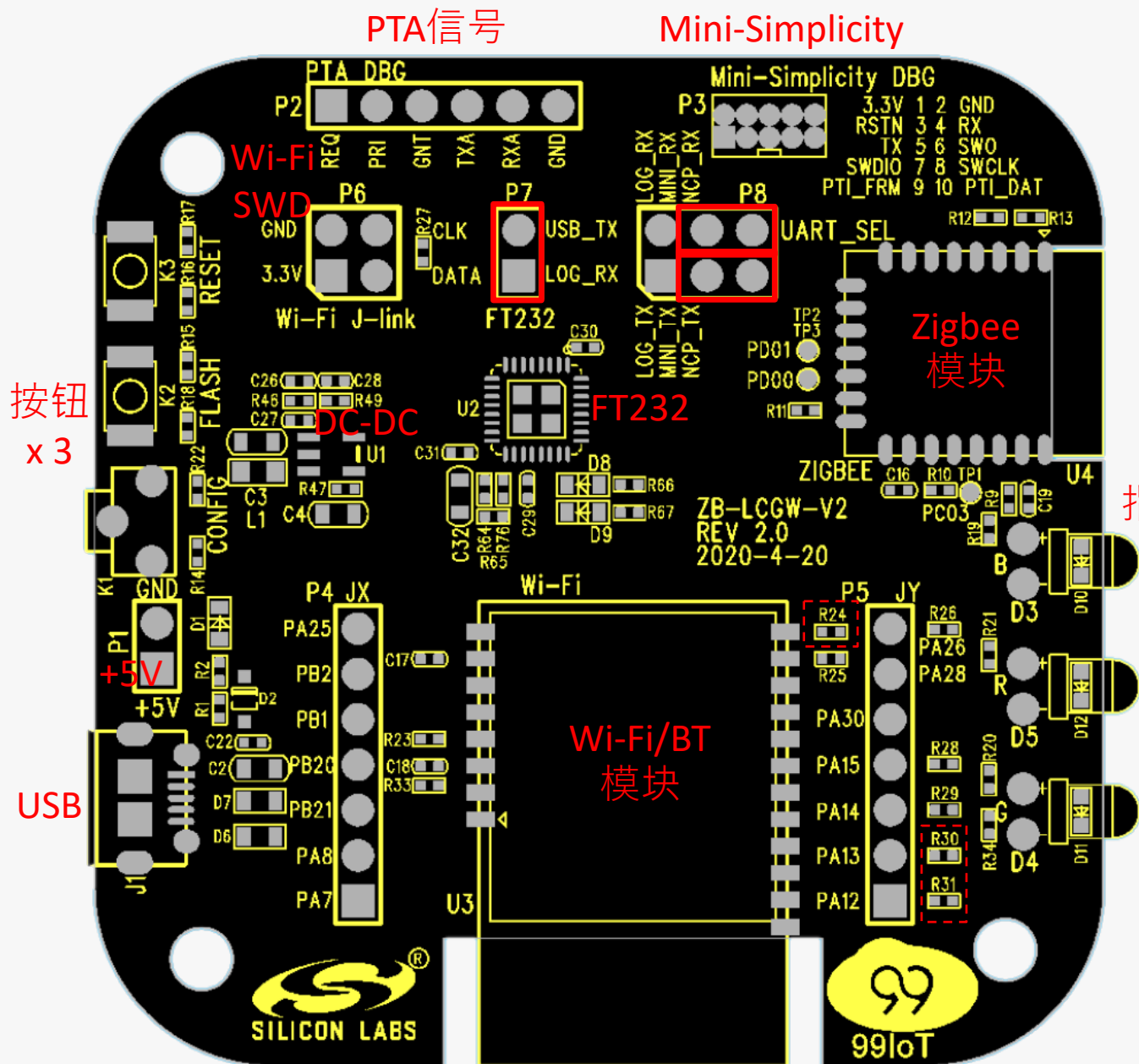
## ■ 硬件设计亮点

- 紧凑的外形並可用于演示、评估和开发等多用途的套件
- 配有公模塑料外壳，用于演示或可作为DIY网关使用
- 支持Wi-Fi/BLE和Zigbee PTA共存
- PWM驱动的红/绿/蓝LED指示灯
- 配置按钮，位于外壳内的固件更新和复位按钮
- Wi-Fi/BLE模块J-Link的4针2.54mm连接器
- Zigbee NCP模块10针1.27mm Mini-simplicity连接器 (SWD, PTI, US0)
- PTA调试6针2.54mm连接器，用于PTA信号探测和调试
- Micro-USB接口：用于1. USB供电，2. 日志串口转USB，3. Wi-Fi固件更新





# Zigbee 3.0 LCGWv2(第2代)



## 跳帽

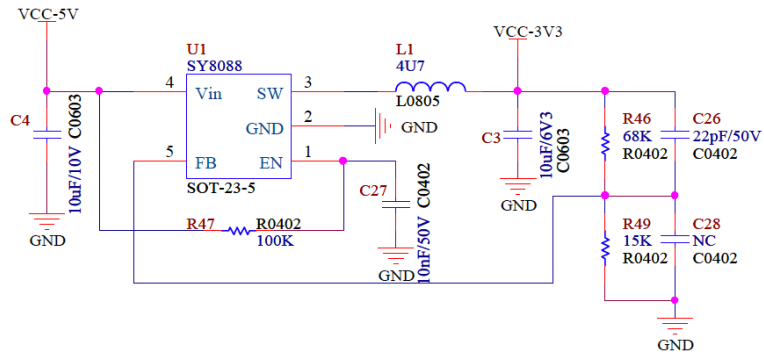
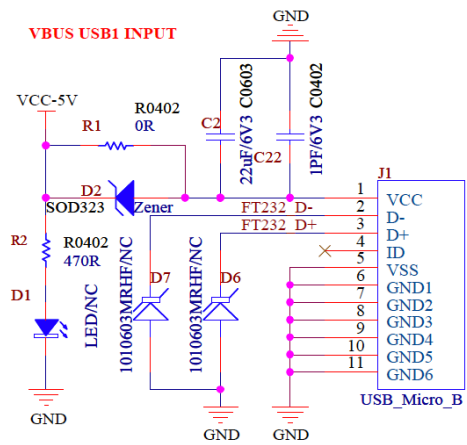
- P7: 默认短接，启用USB上的AmebaD日志串口和Wi-Fi固件更新
- P8: Mini-simplicity TX, RX引脚连接至ZB NCP模块或主机日志串口

## Wi-Fi J-Link

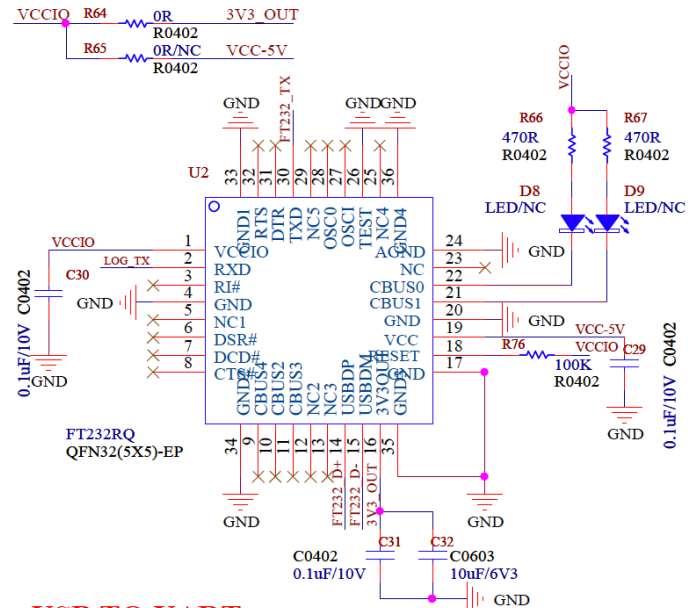
- 移除R24以启用Wi-Fi SWD (断开SWD\_CLK与NCP复位电路的连接)  
通过Mini-Simplicity连接器使用Zigbee CLI命令
- 移除R30, R31以断开从主机到NCP串口的连接

# LCGW v2 原理图

## VBUS USB1 INPUT

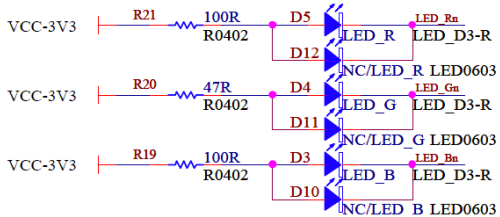


+5V -> 3.3V

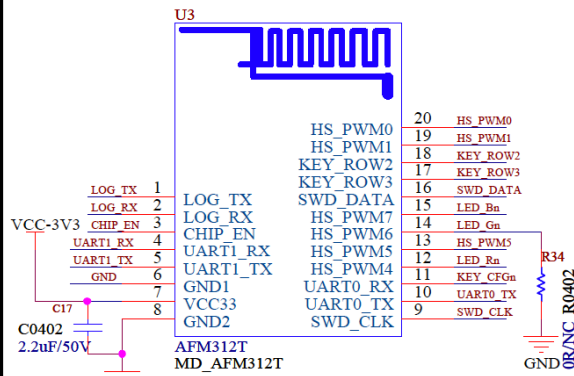
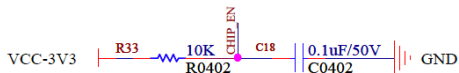
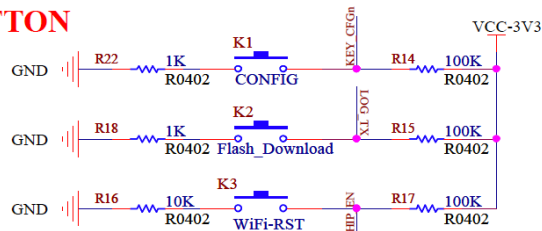


USB TO UART

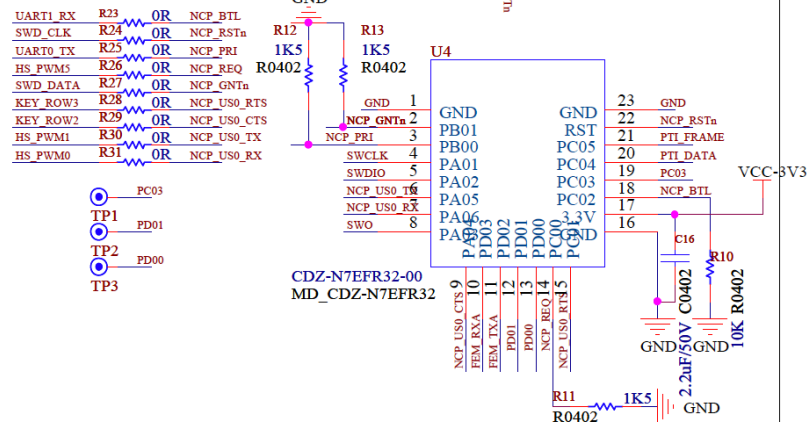
## LED RGB



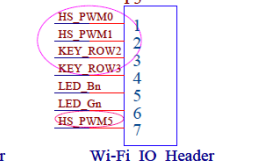
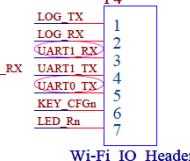
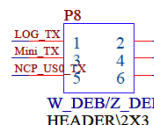
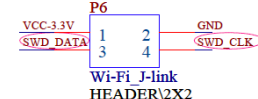
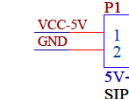
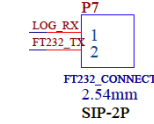
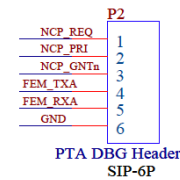
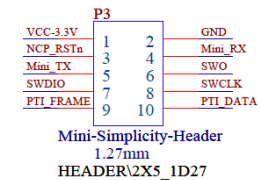
## BUTTON



Wi-Fi



ZIGBEE



# RTL8720CS到EFR32MG21的GPIO映射

硬件块	管脚名称	8720CS GPIO	说明
LOG_UART	LOG_TX	PA7*	AmebaD LOG_TX (参见KEY_FLASH按钮)
	LOG_RX	PA8	AmebaD LOG_RX
NCP_UART	UART1_TX	PA12	NCP US0_RX (MG21 PA06)
	UART1_RX	PA13	NCP US0_TX (MG21 PA05)
	UART1_RTS	PA14	NCP US0_CTS (MG21 PA04)
	UART1_CTS	PA15	NCP US0_RTS (MG21 PC01)
NCP Ctrl	NCP_RSTn	PB3	NCP_RSTn, 低电平有效 (8720的SWD_CLK)
	NCP_BTL	PB21	外部10K下拉电阻, NCP (MG21 PC02) (NCP Bootload模式: 1:NCP固件更新, 0:NCP正常启动)
3W-PTA	NCP_REQ	PA26	外部1.5K下拉电阻, NCP_REQ (MG21 PC00)
	NCP_PRI	PB1	外部1.5K下拉电阻, NCP_PRI (MG21 PB00)
	NCP_GNTn	PA27	NCP_GNTn (MG21 PB01), 低电平有效 (8720的SWD_DIO)
Buttons	KEY_CFGn	PB2	CONFIG按钮, 低电平有效, 外部10K上拉电阻
	KEY_FLASH	PA7	FLASH按钮, 低电平有效 (Boot模式: 1:正常模式, 0:串口下载模式)
	KEY_RESETh	CHIP_EN	RESET按钮(RTL8720CS), 低电平有效
LEDs	LED_Rn	PA25	红LED / PWM4, 低电平有效
	LED_Gn	PA28	绿LED / PWM6, 低电平有效
	LED_Bn	PA30	蓝LED / PWM7, 低电平有效 (8720的SPS选择: 1:SPS模式, 0:LDO模式) 外部10K上拉电阻

# 主机: AmebaD SDK支的AmebaCS 2.4GHz单频WiFi+BT5系列

Items	<del>RTL8720CSM-VA1</del> RTL8720CS-VA1	RTL8721CSM-VA1	RTL8722CSM-VA1
Package (QFN)	48 pins, 6x6	68 pins, 8x8	88 pins, 10x10
wifi	2.4GHz 802.11n 1x1	2.4GHz 802.11n 1x1	2.4GHz 802.11n 1x1
BLE 5.0	Y	Y	Y
BLE mesh/gateway	Y	Y	Y
CPU (KM4)	200MHz	200MHz	200MHz
SRAM + PSRAM	<del>512KB + 4MB /</del> 512KB	512KB + 4MB	512KB + 4MB
Flash (XIP)	ext.	ext.	ext.
UART (max.)	x2	x3	x4
SPI (max.)	x3	x3	x3
RTC	N	Y	Y
I2C	x2	x1	x2
SDIO slave	N	N	Y
SD card supported	N	N	Y
PWM (max.)	x12	x17	x23
Audio codec	N	Y	Y
I2S	Y	Y	Y
LCD controller	N	N	Y
USB	Y	Y	Y
ADC + Vbat measure	x3	x6 + 1	x6 + 1
Market	需要资源大(CPU/RAM)的透传应用	8720CS + codec	8721CSM + SDIO, LCD

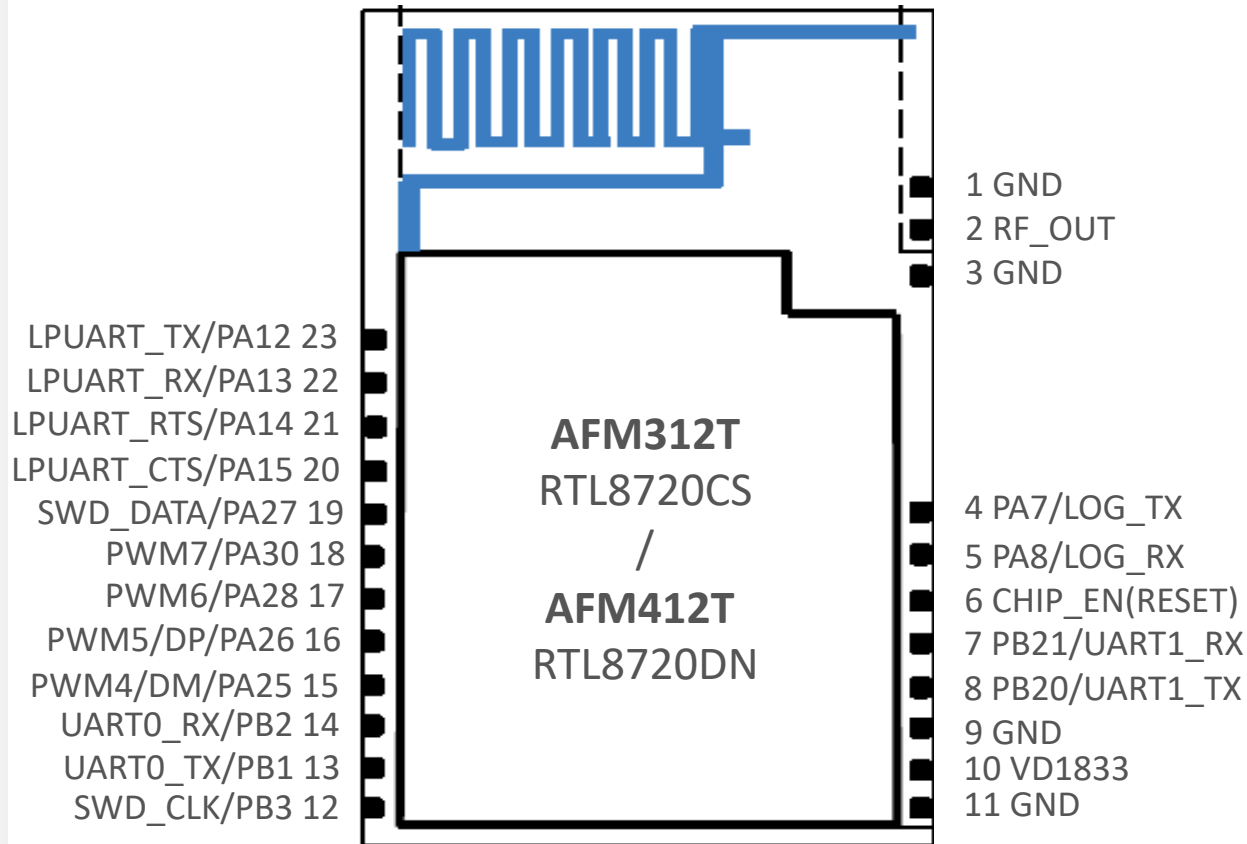
停产

# 主机: AmebaD SDK支持的AmebaD 2.4+5GHz双频WiFi+BT5系列

Items	<del>RTL8720DM-VA1/ RTL8720DN-VA1</del>	RTL8721DM-VA1	RTL8722DM-VA1
Package (QFN)	48 pins, 6x6	68 pins, 8x8	88 pins, 10x10
wifi	2.4G+5G, 11n 1x1	2.4G+5G, 11n 1x1	2.4G+5G, 11n 1x1
BLE 5.0	Y	Y	Y
BLE mesh/gateway	Y	Y	Y
CPU (KM4)	200MHz	200MHz	200MHz
SRAM + PSRAM	<del>512KB + 4MB / 512KB</del>	512KB + 4MB	512KB + 4MB
Flash (XIP)	ext.	ext.	ext.
UART (max.)	x2	x3	x4
SPI (max.)	x2	x3	x3
RTC	N	Y	Y
I2C	x2	x1	x2
SDIO slave	N	N	Y
SD card supported	N	N	Y
PWM (max.)	x10	x15	x18
Audio codec	N	Y	Y
I2S	Y	Y	Y
LCD controller	N	N	Y
USB	Y	Y	Y
Touch keys	N	x4	x4
ADC + Vbat measure	x3	x6 + 1	x6 + 1
Market	需要资源大(CPU/RAM)的 透传应用 + 5G 抗干扰	8720DM + codec	8721DM + SDIO, LCD

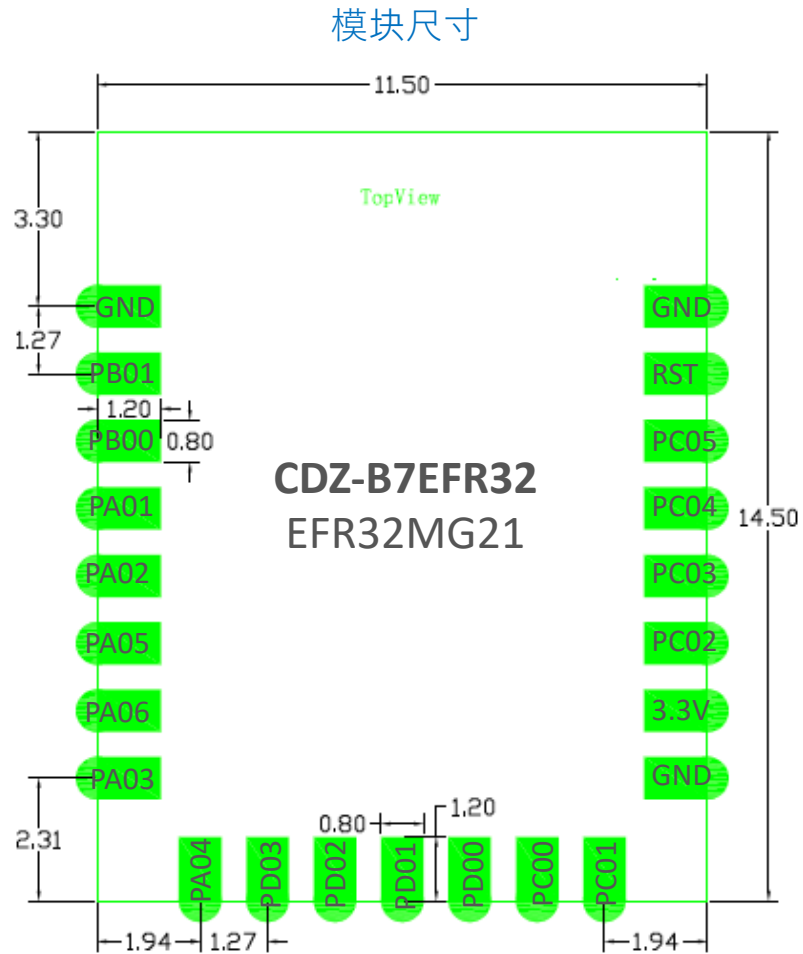
停产

# Wi-Fi/BT5模块：99物联AFM312T(8720CS)/AFM412T(8720DN)



- **MCU**
  - **RTL8720CS**: ARM KM4 200MHz + KM4
- **内存**
  - **512kB SOC SRAM** + 4MB QSPI闪存
- **Wi-Fi**
  - 802.11 b/g/n 1×1 Wi-Fi (HT20)
- **蓝牙**
  - BLE 5.0
- **安全性**
  - Trust-Zone (可选), Secure-boot (可选)
  - Cryptographic加密操作
  - Wi-Fi WEP, WPA, WPA2, WPS
- **接口**
  - 日志串口用于更新Wi-Fi应用固件
  - **所有14个GPIO管脚全引出** (未包括日志串口)
    - 1 LP\_UART(HW FC), 1\_USI\_UART
    - 1 HS\_SPI1 (Master), 1 HS\_USI\_SPI 1 LP\_I2C,
    - IRTX/RX, 1 1W, 8 HS\_PWM, 4 LP\_PWM
- **模块尺寸** : 19mm x 29mm x 3mm

# NCP Zigbee模块：中龙通CDZ-B7EFR32 (EFR32MG21)



## 主要特点

- 部件：[EFR32MG21A020F1024IM32](#) MG21
- 支持Zigbee、蓝牙LE和Open-Thread多种协议
- 内置PCB天线
- **+19.5 dBm 输出功率**
- -104.5 dBm @ 250kbps O-QPSK DSSS 接收灵敏度
- 80MHz的32位ARM Cortex-M33内核
- 1024闪存、96kb内存
- 增强的安全功能
- 单片机外围设备的优化选择
- **引出18个GPIO管脚**
- -40 to +125°C
- 模块尺寸：11.5mm x 14.5mm

# LCGW软件设计

用户指南: ug\_Z3GatewayFreeRTOS\_app\_development\_on\_AmebaD\_SDK

WWW.SILABS.COM





# LCGW软件功能

## ■ 概述

- Simplicity Studio Z3GatewayFreeRTOS应用示例透过ZNet SDK补丁方式提供
- AppBuilder生成的Z3GatewayFreeRTOS项目被构建为lib\_z3ctrl.a库，给AmebaD FreeRTOS gcc SDK使用
- 阿里iotkit-embedded SDK v3.0.1被移植成为lib\_iotkit.a库，给AmebaD SDK使用

## ■ Zigbee功能

- Zigbee 3.0网络创建/管理
- Zigbee 3.0设备安全加入/离开
- 设备发现、设备管理表和其持久性
- ZCL集群命令和属性读/写 (选定的集群和属性)
- ZCL单播/多播/广播发送消息
- 绑定和属性报告
- 设备OTA更新，通过主机串口更新NCP固件
- IAS区域注册
- 主机日志串口支持Z2、Z3 CLI命令 (类似于Z3Gateway的CLI)

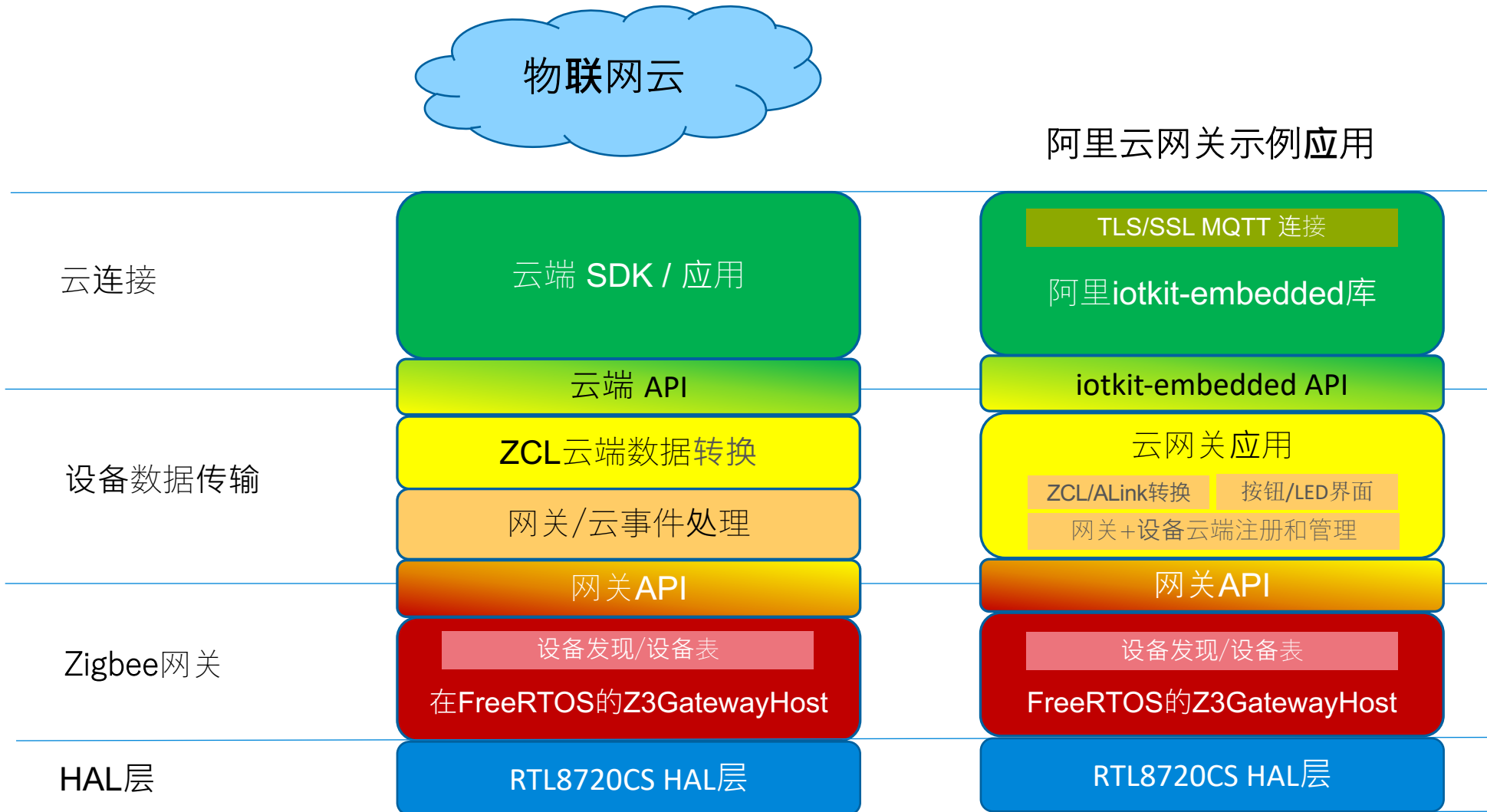
## ■ 云服务集成

- 预集成阿里云iotkit-embedded SDK v3.01作为云网关示例
- TLS安全MQTT连接到阿里云
- ZCL <-> ALink TSL物模型转换 (选定的集群和属性)

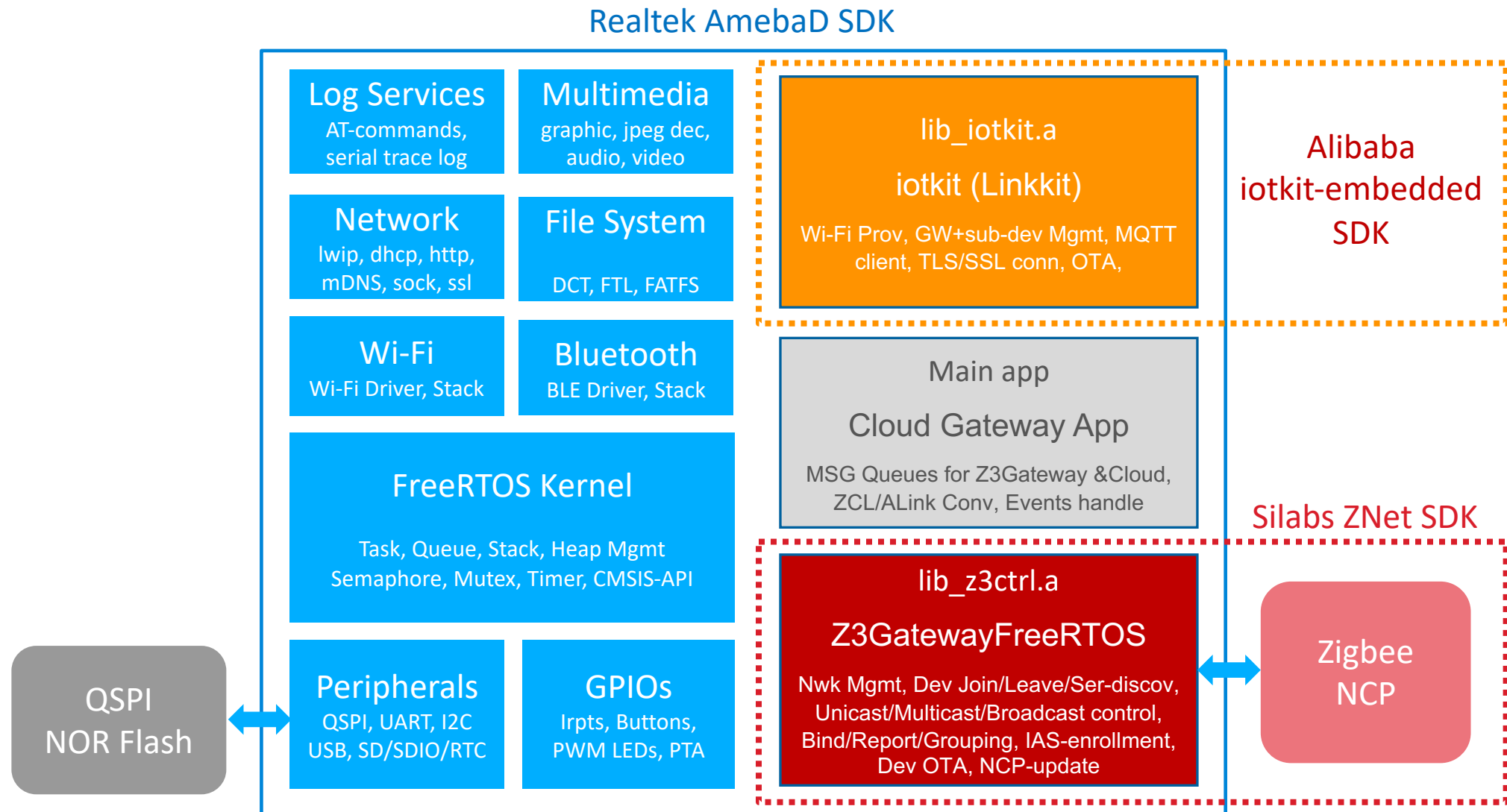
# LCGW软件和Linux Z3Gateway示例的对比

- 一组新的系统HAL/OS函数。
- 为客户添加了一个接口层，以便他们可以使用更简单的消息队列机制来控制网关，而不是原来的ZNet API和回调函数。
- 添加了设备管理功能：它可以管理和跟踪连接的设备，并帮助恢复无响应的节点。
- 添加主机端的NCP固件更新。
- 连接阿里云的示例代码。
- 其他优化和改进，以便于在资源有限的FreeRTOS系统上运行。

# LCGW 软件架构



# LCGW软件实现



# LCGW软件：代码/内存消耗

- 库代码大小：
  - Z3GatewayFreeRTOS库：~250kB
  - 阿里iotkit-embedded库：~120kB
- 总代码大小(外部闪存)：i.e. km0\_km4\_image2.bin
  - 仅Wi-Fi+阿里云：~1024kB
  - Wi-Fi+蓝牙BLE+阿里云：~1190kB
- 内存消耗：
  - Z3GatewayFreeRTOS：
    - 约45kB有Zigbee网络，无设备
    - 每加入20个设备内存增加约1kB
  - 蓝牙BLE功能：
    - 额外消耗约20kB

# LCGW软件：任务

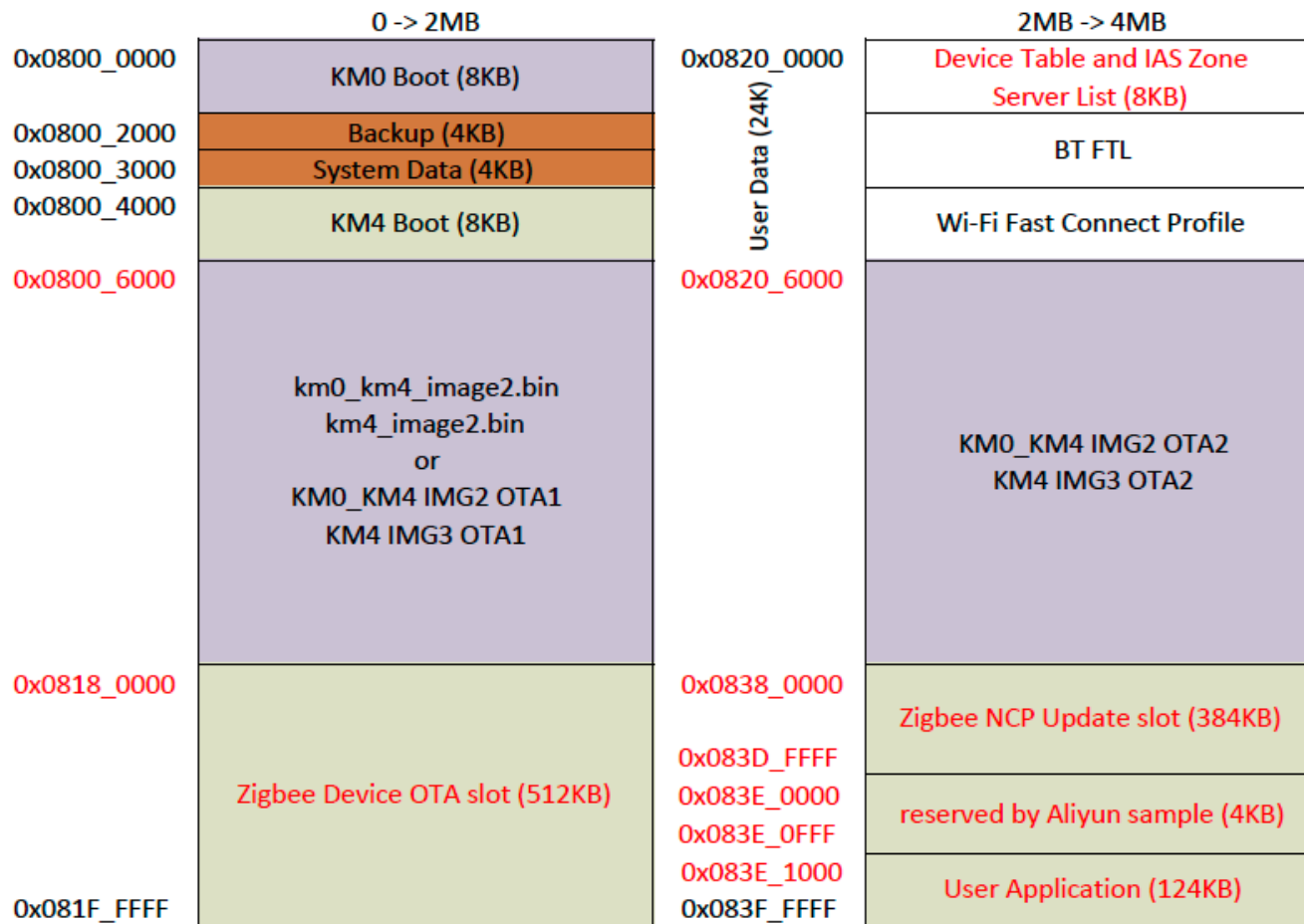
- 阿里云网关示例任务信息：
  - 运行情况：网关连接了阿里云和添加了一个Z3Light设备，在空闲时执行“ATW?”获取任务摘要如下

软件块	任务名称	任务状态	优先级	剩余/堆栈	任务#
main app	button_polling_thread	R	3	222/1024	5
	cloud_msg_center_thread	B	3	472/1024	20
	stack_msg_center_thread	R	3	136/1024	21
lib_iotkit	CoAPServer_yield	R	3	814/4608	18
	cm_yield	B	3	1210/6144	19
	user_dispatch_yield	B	3	672/4096	22
lib_z3ctrl	Z3_MAIN	B	10	1022/1280	16
lwip	TCP_IP	B	9	698/TBD	9
	cmd_thread	B	6	769/TBD	13
	rtw_interrupt_thread	B	6	204/TBD	12
	rtw_rcv_tasklet	B	5	843/TBD	10
	rtw_xmit_tasklet	B	5	214/TBD	11
Log Services	log_servi	X	5	865/1280	1
	LOGUART_TASK	B	5	2006/2048	2
Tmr Services	Tmr Svc	B	1	432/512	8
	IDLE	R	0	478/512	7

状态表示: R=就绪, X=执行/运行, B=堵塞, S=暂停, D=删除

# LCGW软件：外部QSPI闪存映射

- 4MB闪存(LCGW设计): 客户应用程序(使用了约1MB) + OTA分配1.5MBx2、设备OTA 512kB、NCP更新 + 用户储存512KB



- 可扩展8MB闪存: 例如3MBx2用于客户应用程序 + OTA、1MB用于设备OTA、1MB用于NCP更新 + 用户永久储存
  - 参考: Realtek的UM0404 Ameba Flash AVL CN用户手册, 提供兼容的QSPI NOR闪存部件列表和定制

# LCGW的HAL和OS移植

- Z3GatewayFreeRTOS库
  - OS：堆栈malloc/free、互斥、print、信号量、休眠、系统计时器、任务、队列和闪存读写等
  - NCP串口连接：TX、RX、CTS、RTS，使用硬件流量控制
- iotkit-embedded库
  - OS：堆栈malloc/free、互斥、print、信号量、休眠、随机数、任务、重启、计时任务、tcp、udp、AES128
  - AWSS：获取Wi-Fi MAC地址、AP模式、设置频道、获取AP信息、连接AP
  - mbedtls：SSL客户端
- 主应用程序：board\_io.c
  - NCP复位和Bootloader控制
  - RGB LED指示灯，使用定时器+GPIO或PWM控制闪烁
  - 按钮状态读取和按钮扫描任务
- Wi-Fi
  - SW-PTA示例(由Realtek瑞昱提供)

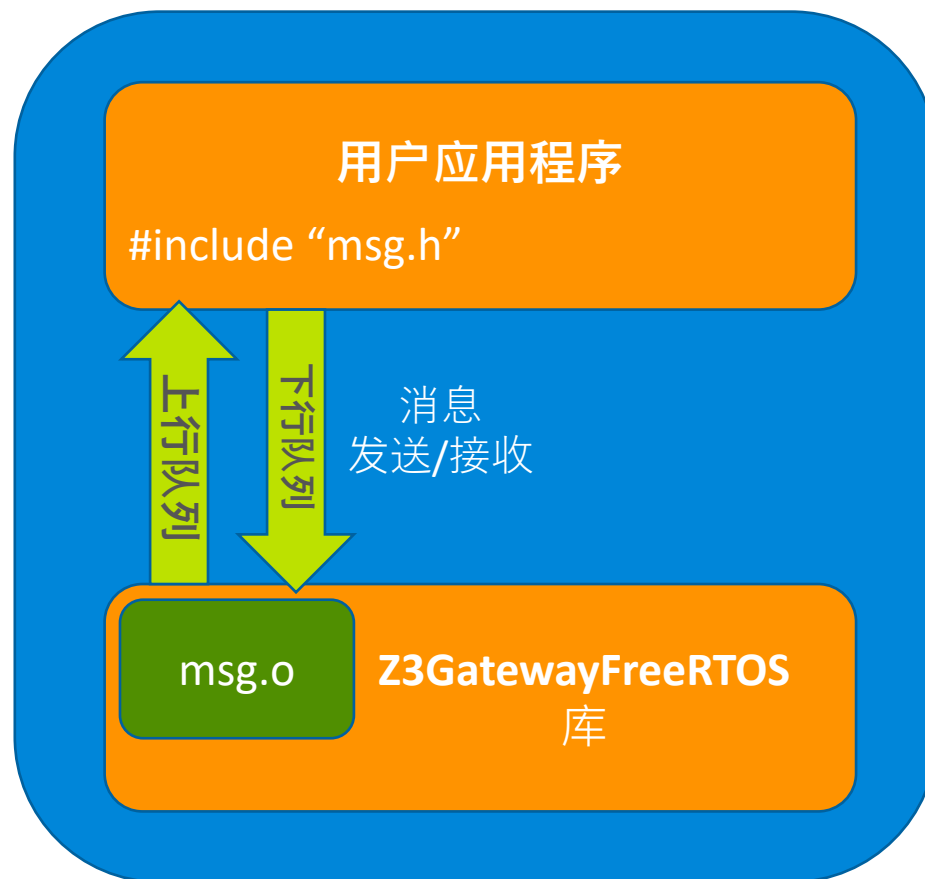


# Z3GatewayFreeRTOS库

- Z3GatewayFreeRTOS库是在EmberZNet SDK 6.6.5上打补丁后生成的。
- 它是构建在AmebaD SDK上的一个库，运行在LCGWv2硬件上。
- 主要功能类别：
  - Zigbee网络管理
  - 远端设备管理
  - 远端设备通信
  - 远端设备的OTA
  - Zigbee NCP固件更新
- Z3GatewayFreeRTOS库专用的闪存空间：设备OTA固件和NCP固件各一个存储空间。
- 提供了基于该库的API的示例CLI “z2”命令行。
- 保留了過往Znet插件的CLI命令行。

# Z3GatewayFreeRTOS库APIs

- Z3GatewayFreeRTOS库和用户应用程序之间的API使用了消息队列。用户应用程序通过下行队列向库发送命令，并通过上行队列从库获取消息。



# Z3GatewayFreeRTOS示例CLI

- LCGW在Ameba日志串口上提供了基于Z3GatewayFreeRTOS库消息队列的演示CLI (Command Line Interface)，其前缀是“z2”。
- 目的是为了本地测试，在默认情况下是禁用的，它不能与iotkit-embedded库同时使用。
- 要启用它，用户需要在platform\_opts.h中更改以下宏定义：
  1. 将“`#define CLOUD_GATEWAY_ENABLE`”定义从“1”更改为“0”，以禁用云操作
  2. 将“`#define LOGSEV_ZIGBEE_CLI_CMD`”定义设为1，使日志服务使能Zigbee CLI命令
  3. 将“`#define LOGSEV_PROCESS_UP_MSG`”定义从“0”更改为“1”，使日志服务能够处理Zigbee堆栈消息
- 提供的CLI:

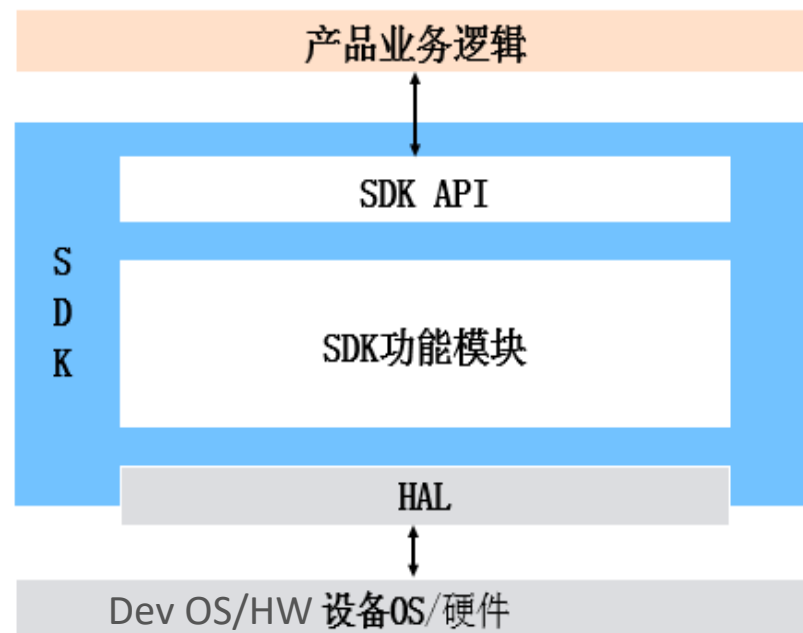
#z2 ?

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

# iotkit-embedded库

- Zigbee设备没有IP地址可以直连到云端，LCGW集成了阿里巴巴的iotkit-embedded SDK v3.0.1 (又名linkkit)，作为一个完整的网关接云的示例以供客户参考，非生产就绪的软件质量。
- iotkit-embedded C-SDK功能:
  - 连接飞燕平台：1 MQTT 2 CoAP 3 HTTPS 或 4 HTTP2
  - 设备标识：1机1密(1设备1密钥)，1型1密(1产品1密钥)
  - 设备数据模型：
    - 属性报告，属性设置
    - 服务呼叫
    - 事件报告
  - 云服务器入口国家/地区配置 (华东-上海)
  - OTA:设备FOTA，检索设备配置文件 (未实现)
  - 子设备管理：添加、删除、验证、数据模型 (ALink)
  - 网关Wi-Fi配网引导方式 (设备AP配网)
  - 使用ALCS服务器/客户端控制本地内部网设备(未实现)
  - 绑定设备并用令牌与云同步(是)
  - 影子设备：设备缓存存储在云中以供查询 (未实现)

## iotkit-embedded SDK



# 用户主程序：cloud\_gateway.c

- main()调用start\_button\_polling()：创建按钮扫描任务
  - Board\_io：调用get\_button\_poll\_event() 扫描按钮并处理按钮事件
- main()调用start\_cloud\_gateway()：
  - 启动看门狗复位定时器(60s超时)，在user\_dispatch\_yield()任务中每~200ms刷新一次
  - 网关在上电/复位时如果按下配置按钮，启动Wi-Fi配网程序
  - 等待Wi-Fi连接成功，然后调用startGateway()，初始化Z3GatewayFreeRTOS库
  - 创建Zigbee网络(仅在Zigbee网络未有被建立时才会调用成功)
  - 启动阿里iotkit-embedded：
    - 网关设备为自己登录阿里云
    - 创建Cloud\_msg处理任务
    - 创建Stack\_msg处理任务
    - 创建user\_dispatch\_yield任务
  - 设备证书/三元组：
    - 预编码了网关自己和3个Z3Light设备共4组三元组
    - 使用一机一密方式提前烧录设备证书(ProductKey、DeviceName和DeviceSecret)，注册设备，上线，然后上报数据。

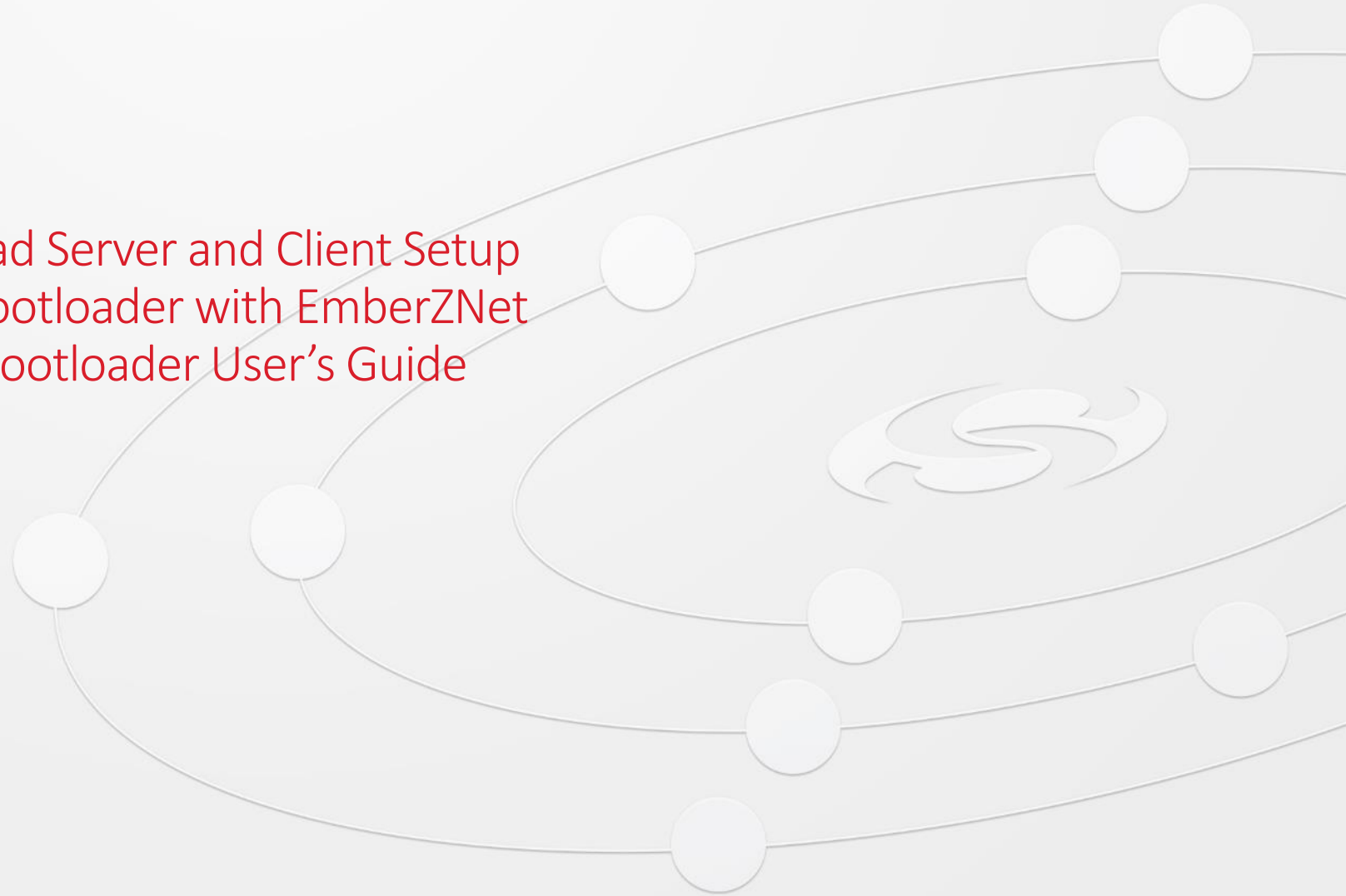
# 用户主程序：cloud\_gateway\_interface.c

- Cloud\_msg任务
  - 实现回调：
    - Connect/Disconnect/PropertySet/ReportRep/TimeStampRep/InitComplete/PermitJoin/ServReq
  - 从Cloud\_msg队列分派消息
  - 调用Z3GatewayFreeRTOS的API (将消息发送到Zigbee堆栈)：
    - CmdCtrl/FromNwk/PermitJoin/CloseJoin/DelDev/SetOnOff/SetBrightness/
- Stack\_msg任务
  - 从Stack\_msg队列分派消息
  - 调用iotkit的API (将MTQQ主题发布到阿里云)
- ZCL <-> Alink JSON格式互转换
- 辅助工具：
  - Board\_io: 控制LED指示灯开/关/闪烁，按键扫描
  - hasing: Z3安装码生成链接密钥
  - wifi\_prov: 网关Wi-Fi配网引导 (取决于iotkit AWSS的DevAP模式)

# 设备OTA和NCP更新

参考: AN728 Over-the-Air Bootload Server and Client Setup  
AN1084 Using the Gecko Bootloader with EmberZNet  
UG266 Silicon Labs Gecko Bootloader User's Guide

WWW.SILABS.COM



# Zigbee设备OTA概述

- 在LCGWv2的闪存中预留了一个专用的OTA固件存储空间：
  - 0x08180000到0x081FFFF，大小为512k字节。
- 用户应用程序负责：
  - 从云端下载并将OTA固件(.ota文件格式)保存到存储空间里。
  - 通过发出“OTA\_FILE\_RELOAD”命令，让库知道OTA文件已准备就绪。
- 当任何远端设备查询OTA升级固件并与LCGWv2闪存中的OTA文件匹配，OTA升级过程将自动启动。
- 当一个OTA固件在LCGWv2的闪存中准备就绪时，用户应用程序还可以发送一个“OTA\_image\_NOTIFY”命令，告诉特定设备OTA固件已经准备就绪，这样设备会去查询OTA固件并触发OTA升级过程开始。
- 为了保持对设备OTA过程的跟踪，用户应用程序将在整个过程中接收“OTA\_STATUS”消息，该消息显示发送到设备的字节数以及OTA是开始、进行中还是完成状态。



# Zigbee设备OTA最佳实践

- 由于设备OTA将产生大量的Zigbee网络流量，因此建议OTA下载过程在网络操作最少的时间段开始，例如午夜。
- 设备查询OTA间隔时间应该设置为很长，我们可以假设设备不会经常需要OTA更新。用户可以将间隔设置为1-2小时甚至更长的时间。
- 使用“OTA\_IMAGE\_NOTIFY”命令
  - 与等待设备查询新OTA固件不一样，OTA的处理对具有相同类型的多个设备使用“OTA\_IMAGE\_NOTIFY”发送到目标设备，这将更易于管理设备的OTA更新。
  - 通知终端设备，让OTA的处理可以一个接一个地进行。(通常终端设备会在收到通知后启动OTA)

# LCGW NCP升级概述 (1)

- 在LCGWv2的闪存中预留了一个专用的NCP固件存储空间：
  - 0x08380000到0x083DFFFF，大小为384k字节。
- 用户应用程序负责：
  - 从云端下载并将NCP固件(.gbl文件格式)保存到存储空间里。
- NCP更新需要NCP中带有Xmodem功能的Bootloader程序。
- NCP映像就绪后，用户应用程序可以通过发出“NCP\_UPDATE”命令启动NCP固件更新。
- 这个过程需要几分钟才能完成。

## LCGW NCP升级概述 (2)

- NCP固件更新后，Z3GatewayFreeRTOS库将向用户应用程序发送“NCP\_UPDATE\_RSP”消息，指示操作是否成功。如果成功，Z3GatewayFreeRTOS库将自动重新启动NCP，因此也将发送“NCP\_RESET\_STATUS”消息。
- 如果NCP更新失败，用户应用程序将无法获得成功的“NCP\_UPDATE\_RSP”。在这种情况下，用户应该发送另一个“NCP\_UPDATE”命令以重新启动NCP更新。
- 在NCP更新过程中，Zigbee网络功能都不可用，因此建议NCP更新应该安排在网络活动最少的时间段内完成，例如在午夜。
- 请注意在NCP更新后，某些设备将进入孤立状态，稍后将尝试重新加入。Zigbee网络性能将不可避免地受到影响。

# 编译和烧写LCGW固件

快速入门指南 : [qsg\\_Z3GatewayFreeRTOS\\_Aliyun\\_gw\\_example\\_build\\_for\\_LCGW](#)

WWW.SILABS.COM



# 固件编译与烧写概述

1. AmebaD FreeRTOS SDK及其编译环境
2. 解压阿里云网关及sdk库源代码
3. Silicon Labs Simplicity Studio安装
4. EmberZNet SDK v6.65及其Z3GatewayFreeRTOS 补丁的安装
5. 创建和生成Z3GatewayFreeRTOS项目
6. 编译带有阿里云网关的LCGW固件
7. AmebaD固件、NCP更新文件和Z3Light的OTA文件烧写
8. 烧写Zigbee NCP固件
9. 固件烧写后的快速验证

# 1. Realtek AmebaD FreeRTOS SDK编译环境

- AmebaD FreeRTOS SDK ( 已包含GCC-ARM工具链 ) :
  - 公开版本 : github [https://github.com/ambiot/ambd\\_sdk](https://github.com/ambiot/ambd_sdk)
  - NDA 版本 : 支持SW-PTA功能 , 请联系Realtek、其分销商或IDH , 签署保密协议后获得。
- 编译环境 :
  - **Ubuntu Linux 18.04** :
    - KM0的project\_lp编译时间约1分钟 (推荐)
    - KM4的project\_hp编译时间约3分钟 (推荐)
    - 在64位系统上安装32位支持包 :

```
$ 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
```
  - Windows Cygwin 32位 : KM4的project\_hp编译时间约15分钟
- AmebaD SDK (NDA版本) 安装程序 :
  - 请参阅Realtek [AN0400](#)应用说明 , 下载链接: <https://www.amebaiot.com/en/sdk-download-manual-8722dm/>
  - **公开版SDK** : `$ git clone https://github.com/ambiot/ambd_sdk.git` (commit: d64c72293cd6d4b75f6b6f37eaa61376e1290e9c)
  - **NDA版SDK** : 提取 00016286-sdk-amebad\_v6.2b-RC.tar.bz2 内文件到 ~/, 提取以下补丁内文件并覆盖在SDK文件夹上 :
    - 00016574-v6.2b\_intergrated\_critical\_patch\_gcc\_IAR\_v39533.zip
    - 0017141-v6.2b\_critical\_patch\_gcc\_IAR\_v39533\_v40238.zip
    - 20200324\_6.2b\_other\_patch\_sw\_pta\_for\_gongniu\_r41263.zip (这是所需的SW-PTA补丁)
  - 编译默认的realtek\_amebaD\_va0\_example示例项目 , 并将生成的二进制文件烧写到LCGWv2上 , 以便先行验证固件。

## 2. 解压阿里云网关及iotkit库源代码並覆盖AmebaD SDK

- 解压amebad\_sdk\_v6.2\_aliyun\_gw\_example\_patch.zip ~/
- 这包括阿里云网关示例和阿里iotkit-embedded SDK库的源代码
- `unzip -oq amebad_sdk_v6.2_aliyun_gw_example_patch.zip -d ~/`
- 将解压生成的aliyun\_gw\_example\_patch文件夹覆盖AmebaD SDK文件夹：
  - `cp -rf ./amebad_sdk_v6.2_aliyun_gw_example_patch/* ./ambd_sdk/`

### 3. 安装Silicon Labs Simplicity Studio

- 从[www.silabs.com](http://www.silabs.com)下载Simplicity Studio v4
  - Linux Ubuntu 18.04 版本构建环境 (或Windows版本Cygwin构建环境)
  - 提取 SimplicityStudio-v4.tgz内文件到 ~/
  - 使用sudoer运行安装程序，安装完成后，将在应用程序列表中创建一个Simplicity Studio图标。

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

- 单击左下角的**应用程序**图标，然后单击**Simplicity Studio**图标以运行studio：
  - 接受许可证，登录\*您的帐户并进行软件更新（注意：如果您还没有帐户，请使用电子邮件地址注册一个新帐户）
  - 跟随：Installation manager：Install by Product Group -> zigbee -> next -> packages (defaults) -> agree all licenses -> finish
  - 这过程可能需时10~30分钟，取决于网速，请耐心等待。



运行Simplicity Studio



## 4. 安装早期EmberZNet SDK v6.6并打上ZNet SDK补丁

### ■ 安装早期ZNet SDK v6.6

- 1. “Launcher” -> 2. “Update Software” -> 3. “Package Manger” -> 4. “SDKs” -> 5. select “Categories” = “EmberZNet SDK”, “Version”=“All” -> 6. pick “EmberZNet SDK – 6.6.5.0” to install

■ 这过程可能需时5~10分钟，取决于网速

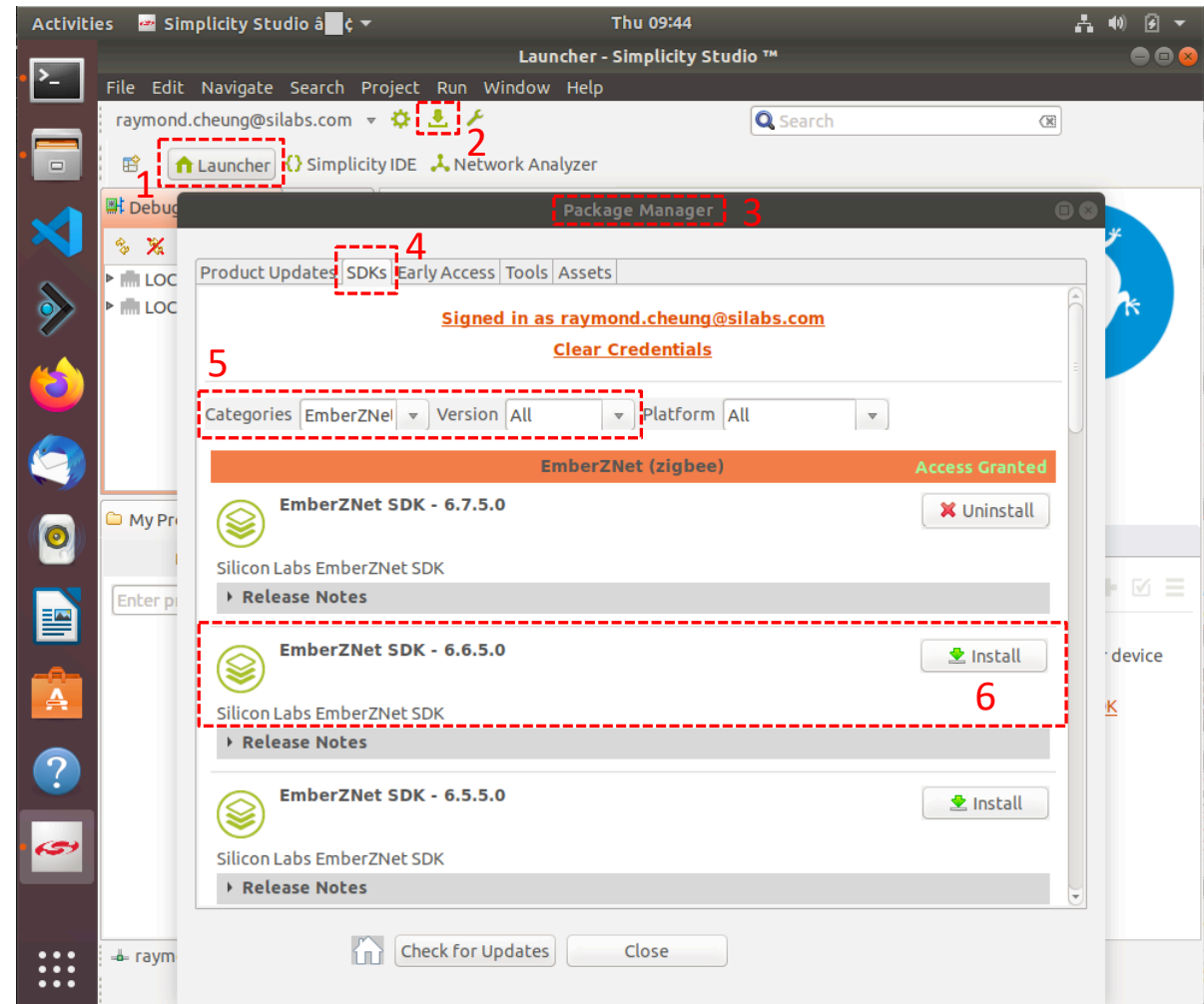
### ■ 打上Znet SDK的Z3GatewayFreeRTOS补丁

- 提取gecko\_sdk\_znet\_v6.6\_patch.zip内文件到~/，请将其复制并覆盖gecko SDK文件夹：

```
$ cp -rf  
./gecko_sdk_znet_v6.6_patch/* ~/SimplicityStudio_  
v4/developer/sdks/gecko_sdk_suite/v2.6/
```

- 导入打过补丁的SDK：

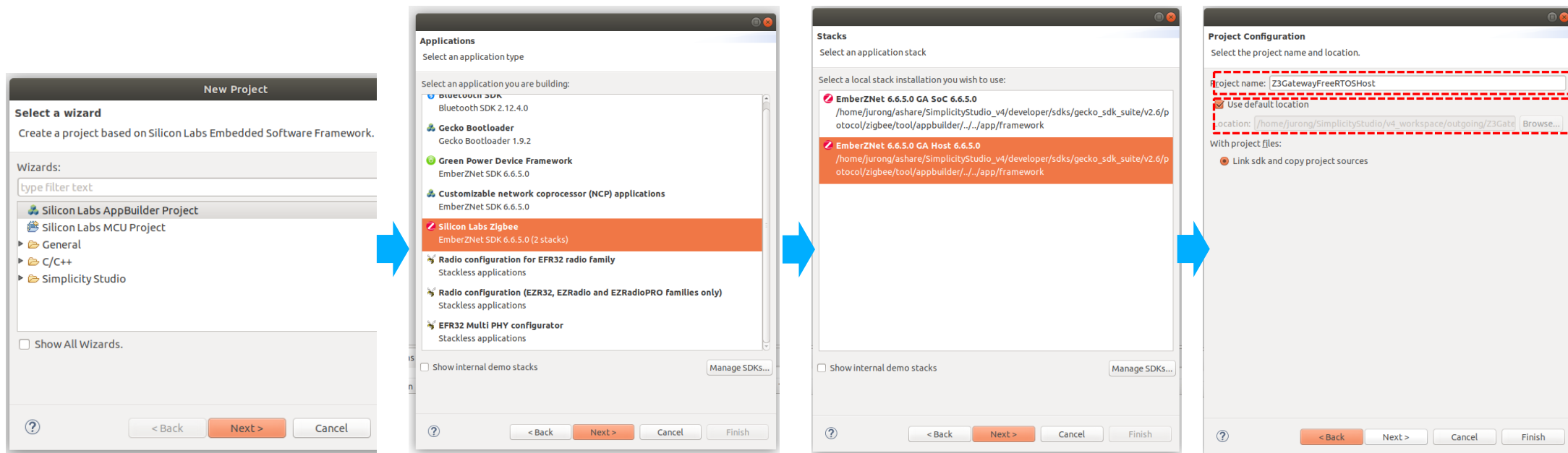
跟随：Windows->Preferences->Simplicity Studio->SDKs，单击添加并浏览到v2.6文件夹



安装早期ZNet SDK 6.6.5

# 5. 创建Z3GatewayFreeRTOS项目到AmebaD SDK步骤 (1)

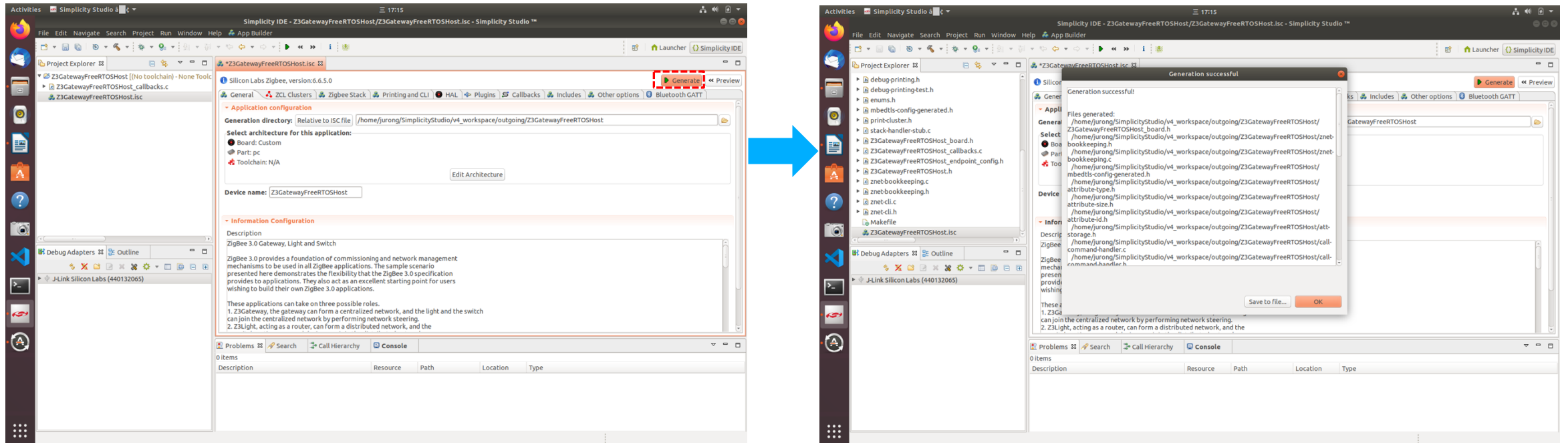
- 在File菜单选择：New a project -> Silicon Labs AppBuilder Project -> Silicon Labs Zigbee -> EmberZnet 6.6.x GA Host -> Z3GatewayFreeRTOS
- 将项目重命名为“lib\_z3ctrl”，并在指定于AmebaD SDK文件夹的相应位置，如：  
~/ambd\_sdk/project/realtek\_amebaD\_va0\_example/GCC-RELEASE/project\_hp/asdk/make/project/
- 单击“Finish”创建新项目



## 5. 创建Z3GatewayFreeRTOS项目到AmebaD SDK步骤 (2)

- 在项目主页面，点击“Generate”按钮，即可在指定的AmebaD SDK目录：

`GCC-RELEASE/project_hp/asdk/make/project/lib_z3ctrl`中生成Zigbee3.0 FreeRTOS gateway的源代码



## 6. 基于AmebaD SDK的LCGWv2固件编译

- 详情请参阅Realtek [AN0400](https://www.amebaiot.com/en/sdk-download-manual-8722dm/)应用说明：<https://www.amebaiot.com/en/sdk-download-manual-8722dm/>
- 编译project\_lp KM0项目：
  - `$ cd ./sdk-amebad_v6.2b/project/realtek_amebaD_va0_example/GCC-RELEASE/project_lp`
  - `$ make clean_all; make all`
- 编译project\_hp KM4项目：
  - `$ cd ./sdk-amebad_v6.2b/project/realtek_amebaD_va0_example/GCC-RELEASE/project_hp`
  - `$ make clean_all;make all`
- 如果“make all”编译不通过，请尝试“make clean”，然后再次“make all”。
- 如果编译通过，将生成以下固件二进制文件：
  1. `~/sdk-amebad_v6.2b/project/realtek_amebaD_va0_example/GCC-RELEASE/project_lp/asdk/image/km0_boot_all.bin`
  2. `~/sdk-amebad_v6.2b/project/realtek_amebaD_va0_example/GCC-RELEASE/project_hp/asdk/image/km4_boot_all.bin`
  3. `~/sdk-amebad_v6.2b/project/realtek_amebaD_va0_example/GCC-RELEASE/project_hp/asdk/image/km0_km4_image2.bin`

# 7. AmebaD固件、NCP更新文件和Z3Light的OTA文件烧写

- 将AmebaD SDK的镜像工具文件夹复制到Window PC上：  
即：~/sdk-amebad\_v6.2b/tools/AmebaD/Image\_Tool/

- 运行RealtekImageTool.exe并设置下列各项：

1. Chip Select：选择“AmebaD(8721D)”

2. COM：选择LCGW对应USB串口COMxx

3. 烧写和下载文件及烧写位置选择

- LCGW固件二进制文件：

1. @0x08000000: km0\_boot\_all.bin
2. @0x08004000: km4\_boot\_all.bin
3. @0x08006000: km0\_km4\_image2.bin

- 设备OTA固件映像：(LCGW支持1个存储空间)

- @0x08180000: 例如Z3LightFw.ota

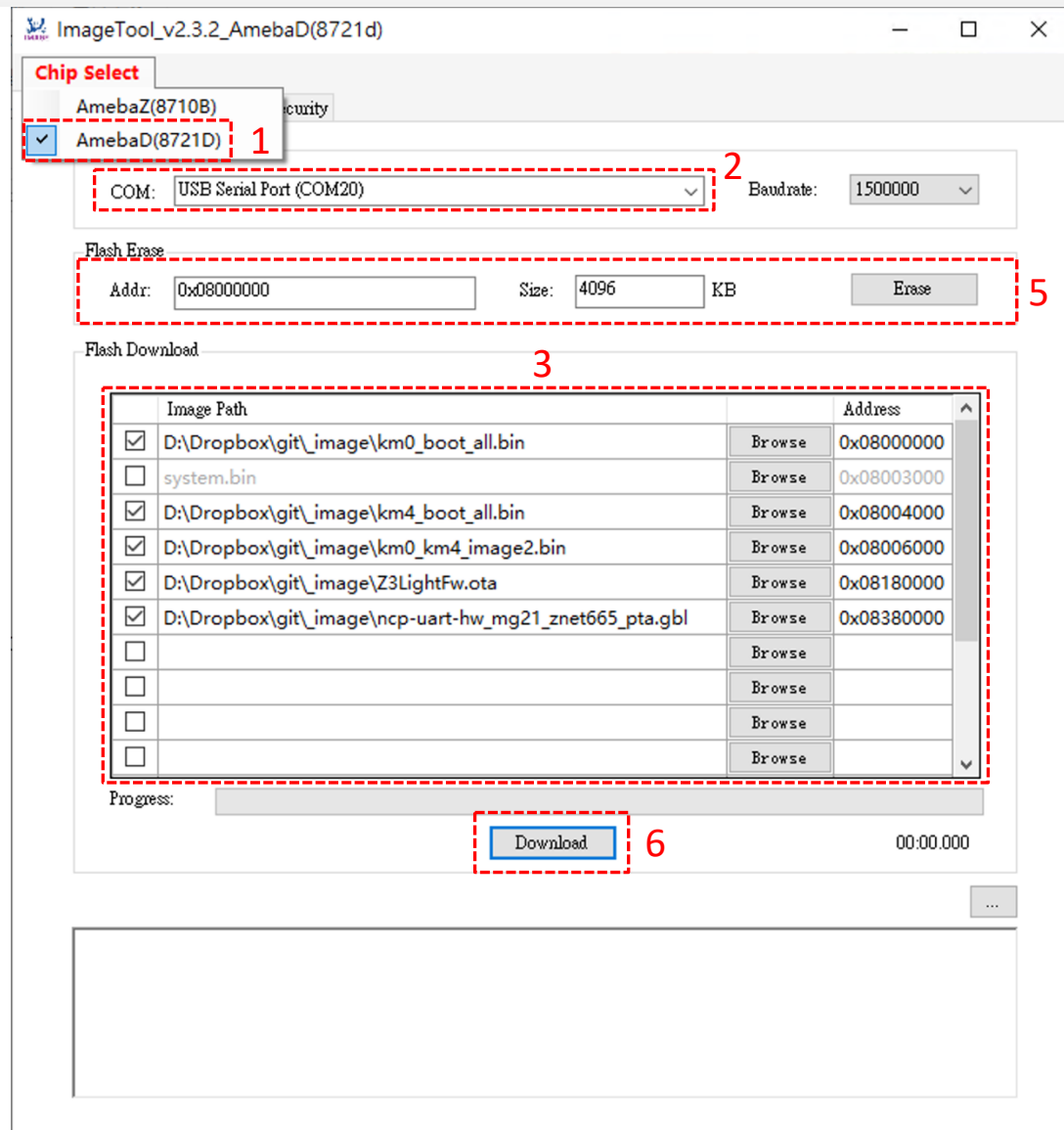
- NCP更新gbl映像：

- @0x08380000: 例如ncp-uart-hw\_mg21\_znet665\_pta.gbl

4. LCGW套件：按住“FLASH”按钮并按“RESET”按钮以启动FLASH模式

5. 可选择设置“Flash Erase”：Address=0x08000000并Size=4096KB，然后单击“Erase”按钮擦除整个4MB外部QSPI闪存

6. 点击“Download”按钮下载并烧写文件...



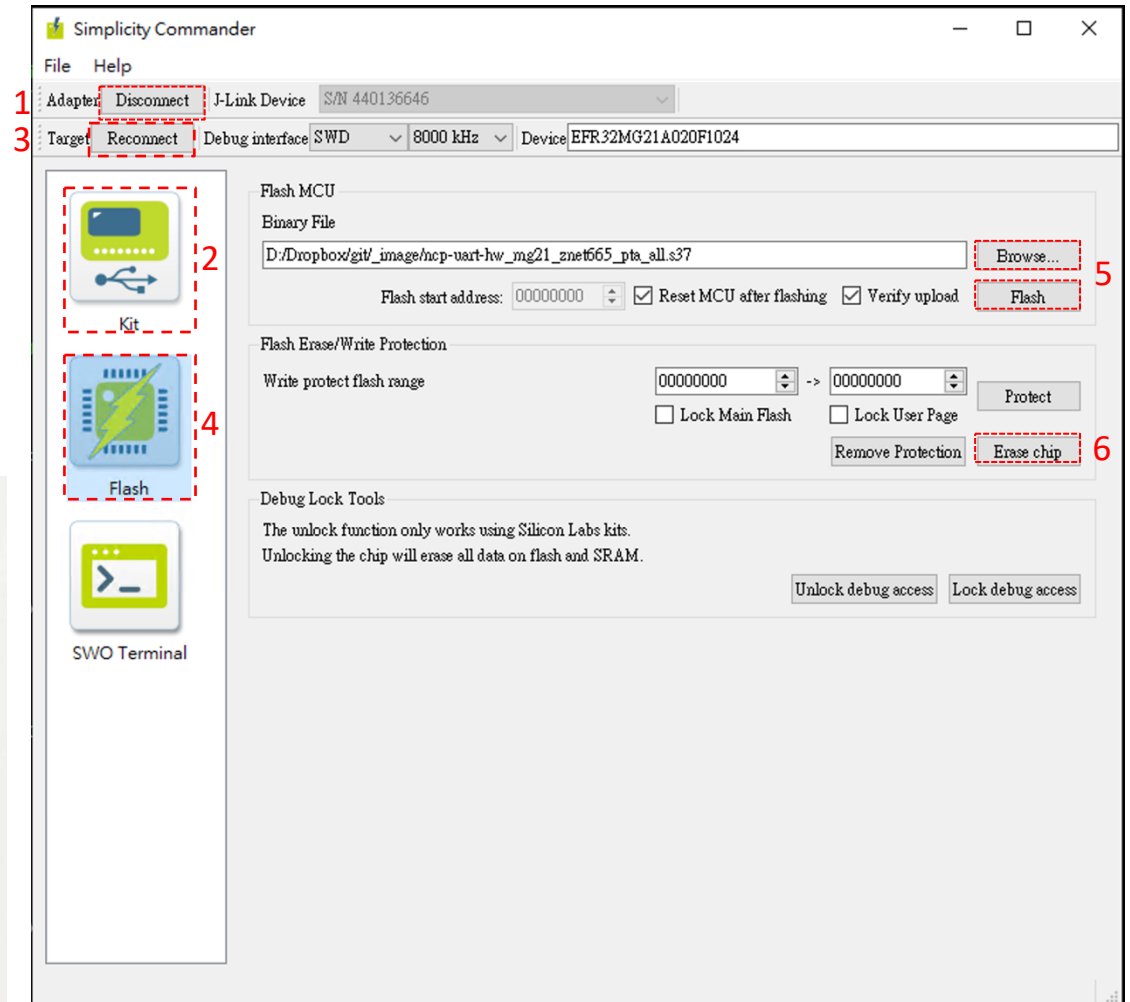
# 8. 烧写Zigbee NCP固件

## ■ Simplicity Commander (在包含在Simplicity Studio中)

1. “Adapter”：选择当前USB连接的WSTK J-Link设备  
→点击“Connect”
2. 点击“Kit”按钮：将debug mode 设置为“OUT”模式
3. “Target”：点击“Connect”，“EFR32MG21A02F1024”会被识别
4. 点击“Flash”按钮...
5. 点击“Browse”按钮，选择NCP和Bootloader的.s37的合并映像文件，即“ncp-hw-znet655\_pta\_all.s37”→点击“Flash”按钮。
6. 提示：可“Erase chip”擦除芯片和单独分别烧写NCP和Bootloader映像文件。



独立版本下载：<https://www.silabs.com/mcu/programming-options>



# 9. 固件烧写后的快速验证

## ■ 使用Teraterm (Windows) 或其他串口调试工具

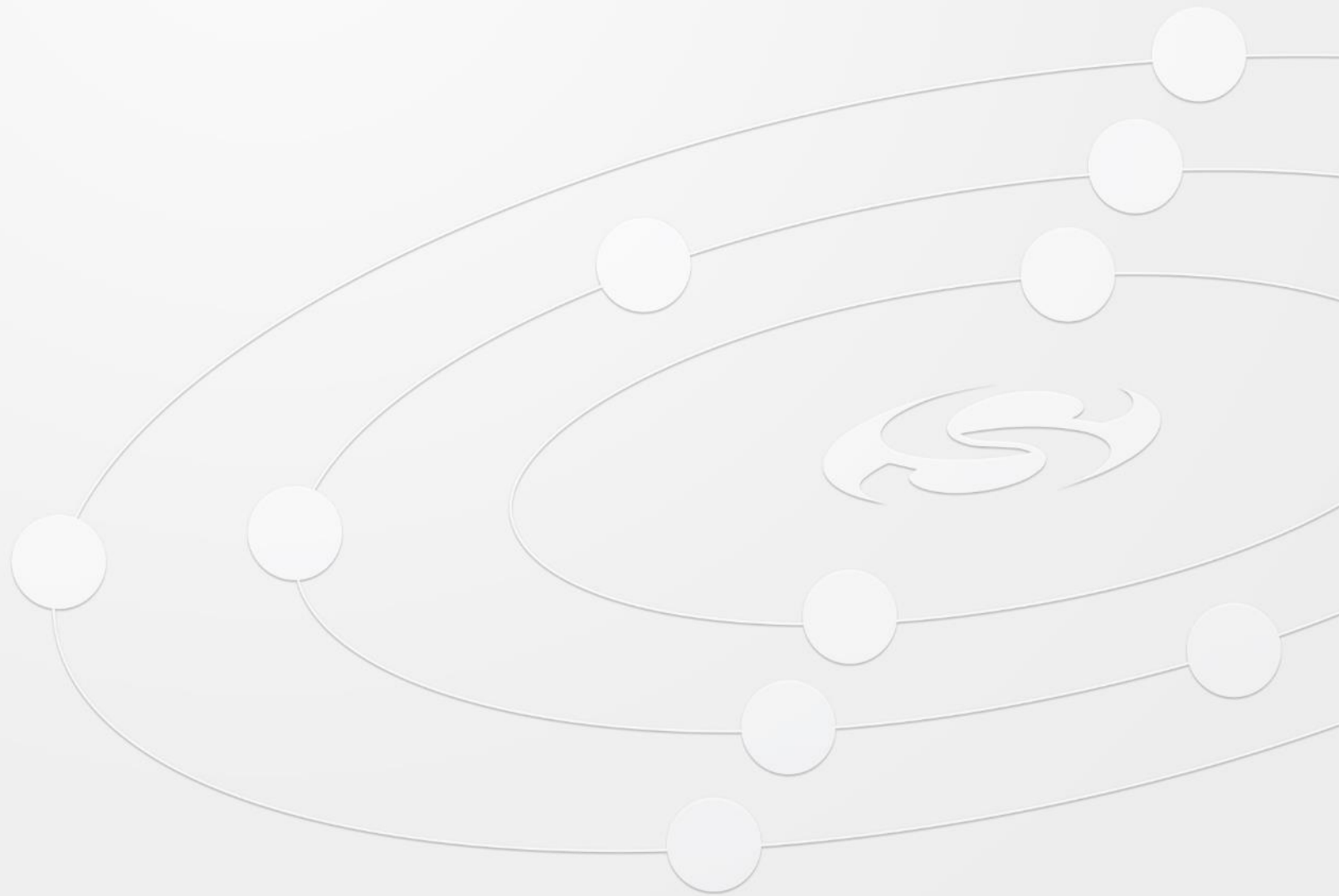
1. 端口设置：**115200bps-8bit-N-1**
2. 按下LCGW套件板上的RESET按钮
3. 命令行中键入以下AT命令：
  - “ATWS” 扫描附近的Wi-Fi AP
  - “ATW0=<ssid>” 需要接入的Wi-Fi网络的SSID
  - “ATW1=<password>” Wi-Fi密码
  - “ATWC” 连接到指定的Wi-Fi网络
4. 如果Wi-Fi连接成功，日志将输出到控制台终端
5. 如右图，输出信息包括：
  - 分配的**IP地址**
  - NCP ezsp启动信息
  - **NCP堆栈版本** (即6.6.5.0)
  - 网络启动成功消息“**network up success**”

```
Interface 0 IP address : 192.168.0.140
WiFi initialized

init_thread(58), Available heap 0x258b0Reset info: 11 (SOFTWARE)
SERIAL INIT-->
ezsp ver 0x07 stack type 0x02 stack ver. [6.6.5 GA build 204]
Ezsp Config: set source route table size to 0x0096: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 0x0004: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 72
Ezsp Config: set end device poll timeout to 0x0008: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: 4, out clusters 19
Ezsp Endpoint 242 added, profile 0xA1E0, in clusters: 0, out clusters 1
numberOfDevices: 0
device table:
gateway interface init
ota file size: 252494
Manufacturer ID: 0x1002
Image Type ID: 0xA001
Version: 0x00000002
Header String: EBL Z3LightSoc
EMBER_NETWORK_UP 0x0000
144 2
ncpStackVer:0x6650
ncpEzspProtocolVer:7
gateway api: network up success

form_nwk.841: To send msg to stack to form a network.
wf_ch=1, zb_ch=24, panId=0x033f, zb_pwr=20
get cmd: CREATE_NWK
ncpStackVer:0x6650
ncpEzspProtocolVer:7
gateway api: network up success
```

# 第一天：问答环节





# 课程内容（第二天）

1. Wi-Fi和Zigbee共存 [30分钟]
  - 共存概述 · Wi-Fi+Zigbee共存策略
  - LCGW Wi-Fi、Zigbee频道管理
  - LCGW NCP固件PTA配置、LCGWv2 PTA信号探测
  - LCGW-PTA共存测试及测试结果
2. RTOS网关集成云 [40分钟]
  - Wi-Fi设备配网概述
  - 阿里云iotkit-embedded及Cloud Gateway应用概述
  - 阿里云上为LCGW、Zigbee灯和开关创建自己的物联网产品
  - 云智能手机应用：LCGW、灯和开关的智能自动化例子
  - 天猫精灵控制LCGW的Zigbee子设备
3. LCGW软件质量保证 [10分钟]
  - LCGW SQA测试
  - Mesh测试网及GW相关测试设施
4. [休息：10分钟]
5. 演示 [30分钟]
  - 生活物联网平台演示 (手机应用)
6. [问答：10分钟]

Silicon Labs  
LCGWv2 套件



# Wi-Fi和Zigbee共存

应用文档 : AN1017 Zigbee and Silicon Labs Thread Coexistence with Wi-Fi  
AN1243 Timing and Test Data for EFR32 CoEx with Wi-Fi

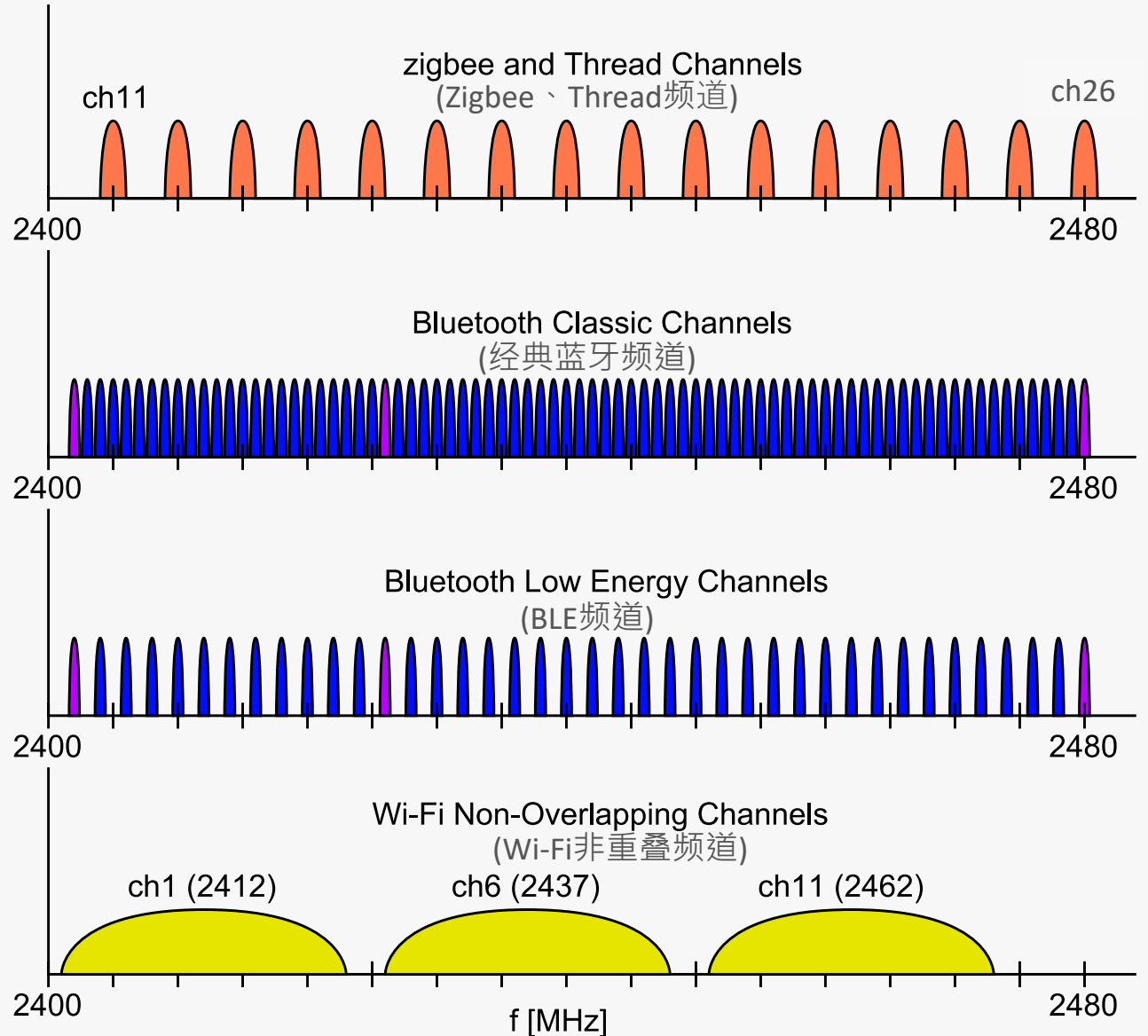
用户指南 : ug\_ncp\_and\_bootloader\_firmware\_customization\_for\_LCGW

WWW.SILABS.COM



# “共存”是什么意思？

- Thread、Zigbee、蓝牙和Wi-Fi共享2.4GHz ISM频段。
- 从一种无线设备发射出的信号对另一种无线接收设备来说就像是无用的噪声。
- 如果所需信号弱于噪声，接收设备将无法正确接收信息。
- 从过往经验来看，即使没有刻意地解决共存的问题，无线设备似乎也能“正常工作”。
- 然而，由于多协议射频部件共存于一个紧凑的网关设备中，“正常工作”不再是理所当然的。





# Wi-Fi和IoT部件射频共存策略

## ■ 共存问题：

- 更高的Wi-Fi发射功率需要更大的天线隔离。
- 更高的Wi-Fi吞吐量导致更高的Wi-Fi占空比。
- 天线隔离通常受到产品尺寸的限制 (仅达到15-20分贝)

## ■ 改善共存：

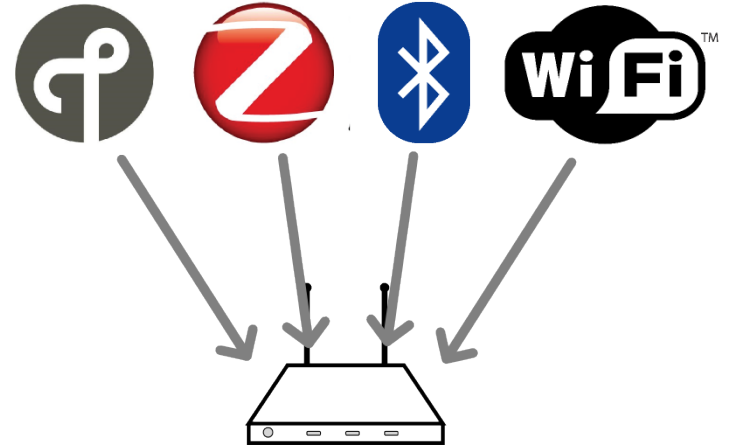
### ■ 非托管共存：

1. 实现分频：尽可能将IoT信道和Wi-Fi信道分隔开
2. Wi-Fi使用20MHz带宽运行：这将为IoT留下半个40MHz的ISM频段
3. 最大化天线间隔：增加天线之间的距离，因为其功率反比于 $R^2$

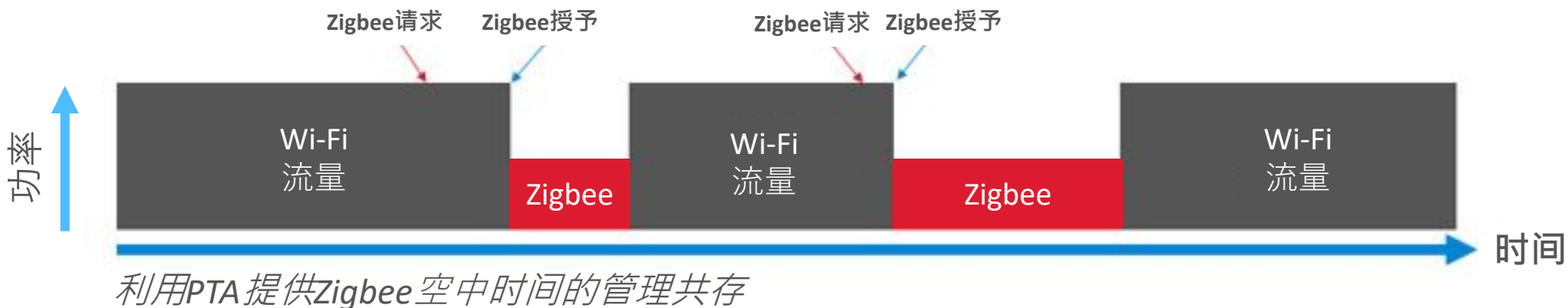
### ■ 托管共存，使用数据包流量仲裁(PTA)：

1. PTA是两种无线之间以标识通信待处理和/或正在进行的发送和接收的信号
2. 在IEEE 802.15.2 (2003) 第6条中描述了PTA，它是一个建议，而不是一个标准
3. Wi-Fi供应商以不同的方式实现PTA，Silicon Labs的PTA是非常可配置的

### ■ **非托管技术是必要的，但还不够！**

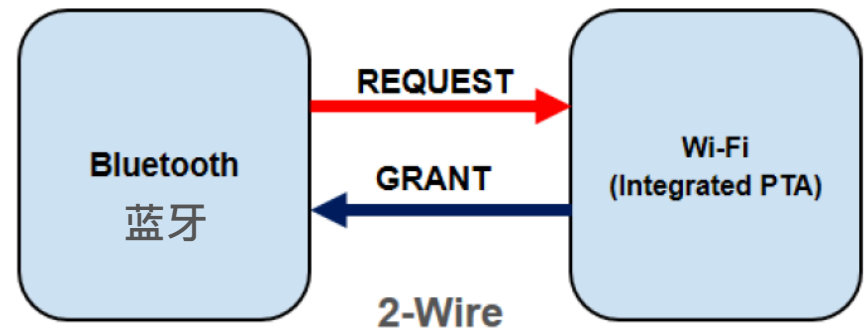
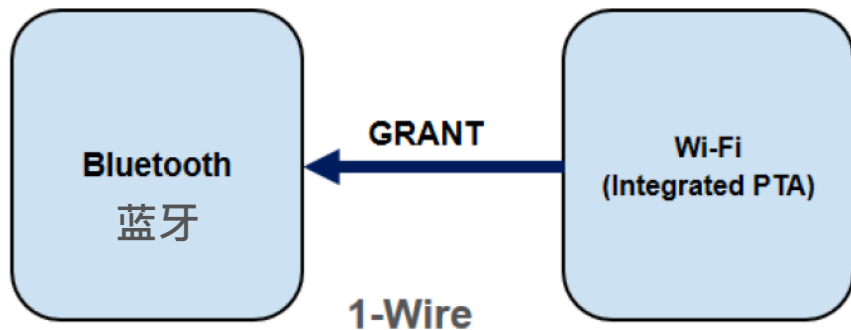


# 共存的概念

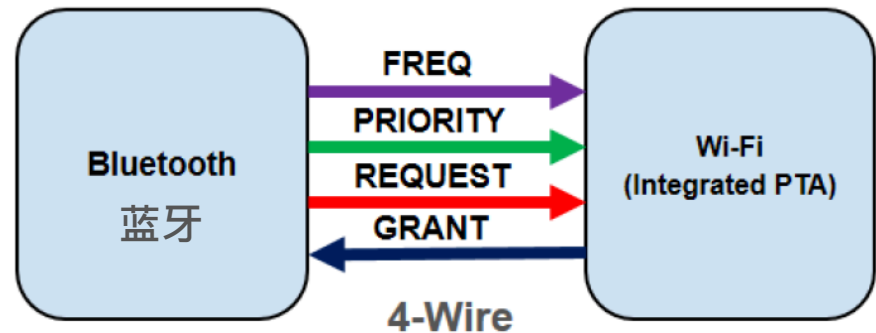
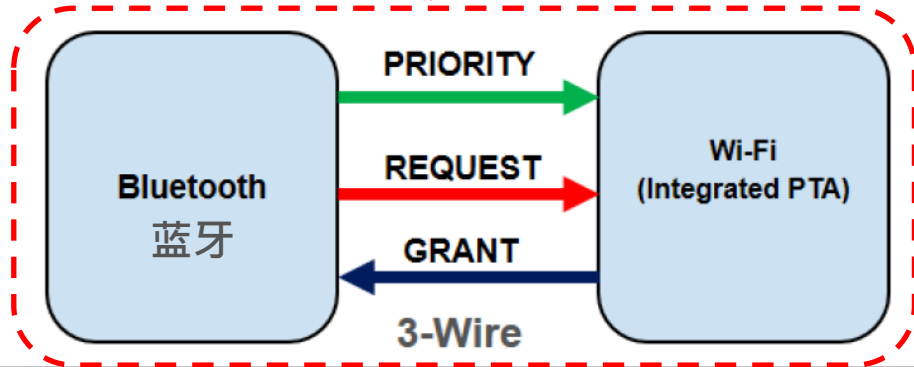


## 802.15.2 PTA布线配置

- 802.15.2建议最初是为蓝牙/Wi-Fi共存而编写的，它包括四种不同的布线配置。
- 添加优先级信号，增加一个状态来增加PTA应用的灵活性。
- Silicon Labs **建议分频**，**不支持频率信号**。
- 三线PTA最为普及并被采用在LCGW中。

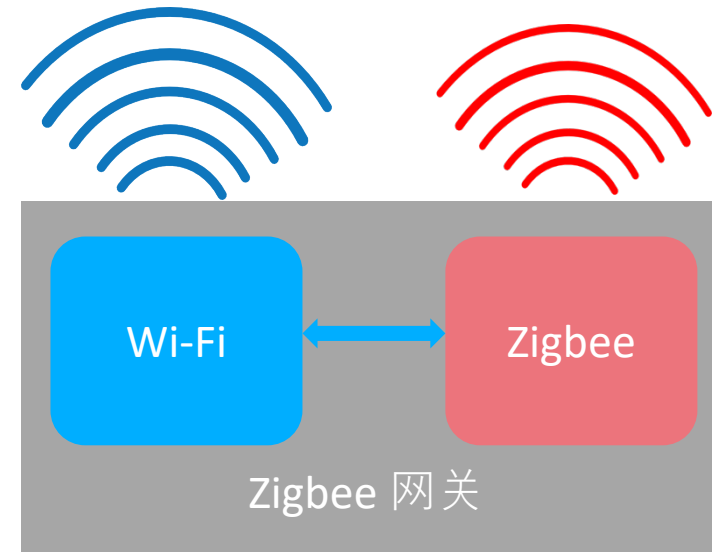


三线PTA最普及并在LCGW实现



# LCGW中Wi-Fi、Zigbee频道的管理

- 在主程序cloud\_gateway\_interface.c中实现：
  - 每当系统启动时...
  - startGateway()，直到GW的Wi-Fi连接到AP
  - 延迟3秒，直到Z3Gateway EZSP初始化完成
  - 调用create\_network\_interface()：
    - 调用wifi\_get\_channel() 获取当前Wi-Fi信道
    - 在faraway\_ch\_list[]中查找远离WiFi信道的Zigbee信道
    - 准备CREATE\_NWK参数：
      - 随机panID
      - 远端Zigbee信道
      - 最大功率20dBm (将被限制为19dBm)
    - 发送CREATE\_NWK消息到下行队列...
    - Z3GatewayFreeRTOS库不会在已有Zigbee网络时重新创建网络
- 动态Zigbee频道管理？
  - 尽管有API更改Zigbee信道，但没有简单的方法让休眠设备立即重新加入Zigbee网络和快速重建源路由表。
  - 当Wi-Fi AP改变其信道时，网关作为STA只能跟随AP而改变信道，信道错开的条件可能被改变。
  - 只依靠PTA：Zigbee在WiFi的远端信道和邻近信道仍可正常工作。在信道重叠情况下，出现一些设备加网和单播消息发送失败。





# 为LCGW准备MG21 NCP+bootloader固件

- 参考：ug\_ncp\_and\_bootloader\_firmware\_customization\_for\_LCGW用户指南
- 导入和编译：(1) ncp-uart-hw\_mg21\_znet665\_pta.sls 和 (2) bootloader-uart-xmodem\_mg21.sls
- 或者从头开始创建项目：ncp\_uart\_hw 示例并使用[EFR32MG21A020F1024IM32](#)部件

- NCP UART接口管脚：启用CTS/RTS硬件流量控制

Name	GPIO	Properties	Module Pin	Remark
USART0 TX	PA05	out, active-low	Pin 6	
USART0 RX	PA06	in, active-low	Pin 7	
USART0 CTS	PA04	in, active-low	Pin 9	
USART0 RTS	PC01	out, active-low	Pin 15	

- PTA接口管脚：3线PTA模式，静态优先级

Name	GPIO	Properties	Module Pin	Remark
REQUEST	PC00	out, active-high, non-shared mode	Pin 14	
RRIORITY	PB00	out, active-high, non-shared mode	Pin 3	
GRANT	PB01	in, active-low	Pin 2	

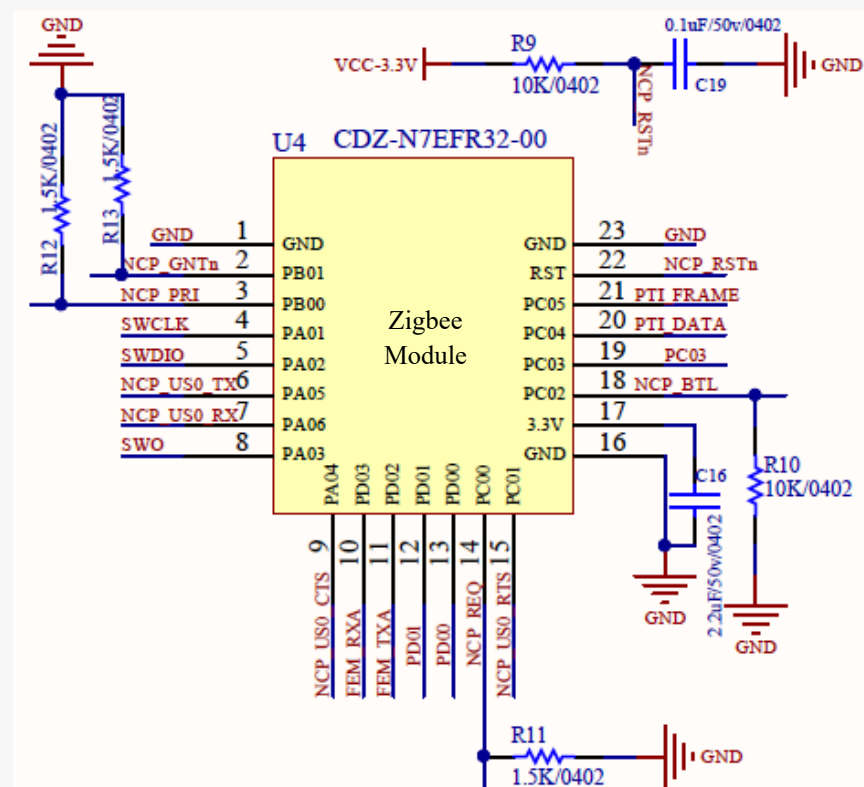
- Bootloader程序管脚: 高电平有效(进入Bootloader程序)

Name	GPIO	Properties	Module Pin	Remark
NCP_BTL	PC02	out, active-high	Pin 18	

- 调试管脚：FEM TXA/RXA，用于网络分析仪PTI 数据捕获

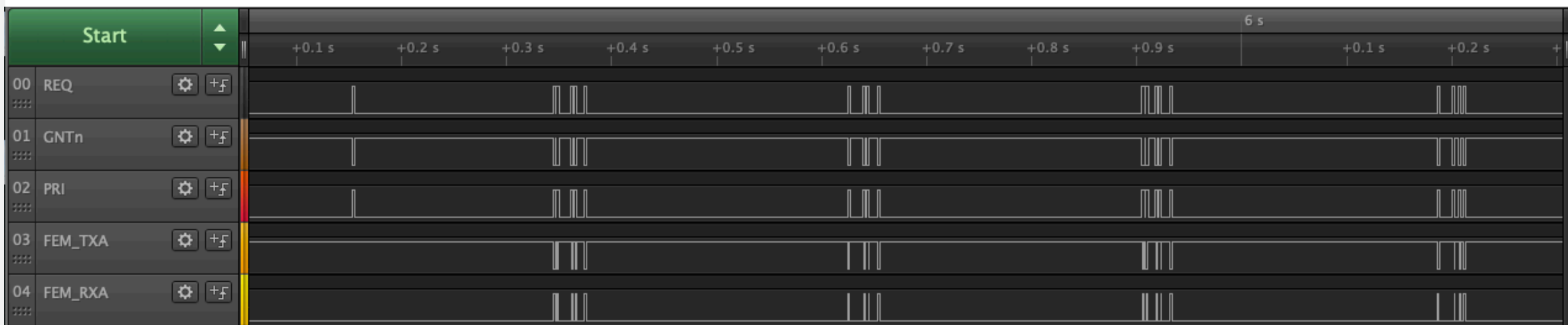
Name	GPIO	Properties	Module Pin	Remark
FEM TXA	PD02	out, active-high	Pin 11	Debug
FEM_RXA	PD03	out, active-high	Pin 10	Debug
PTI_DATA	PC04	out, active-low	Pin 20	Debug
PTI_FRAME	PC05	out, active-low	Pin 21	Debug

- Zigbee PRO堆栈插件：更改Source-Route表大小到150

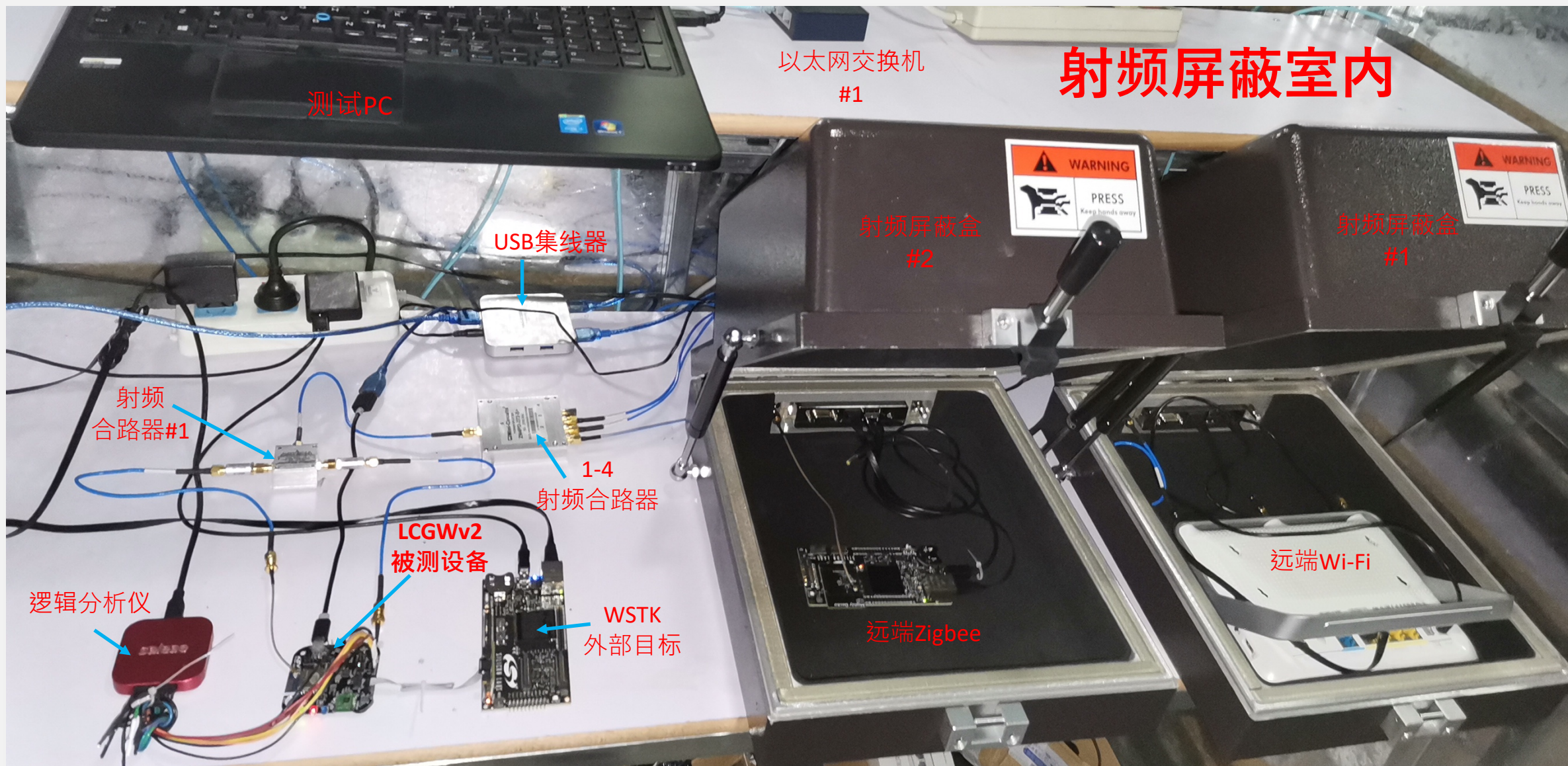


# LCGW PTA及调试管脚信号探测

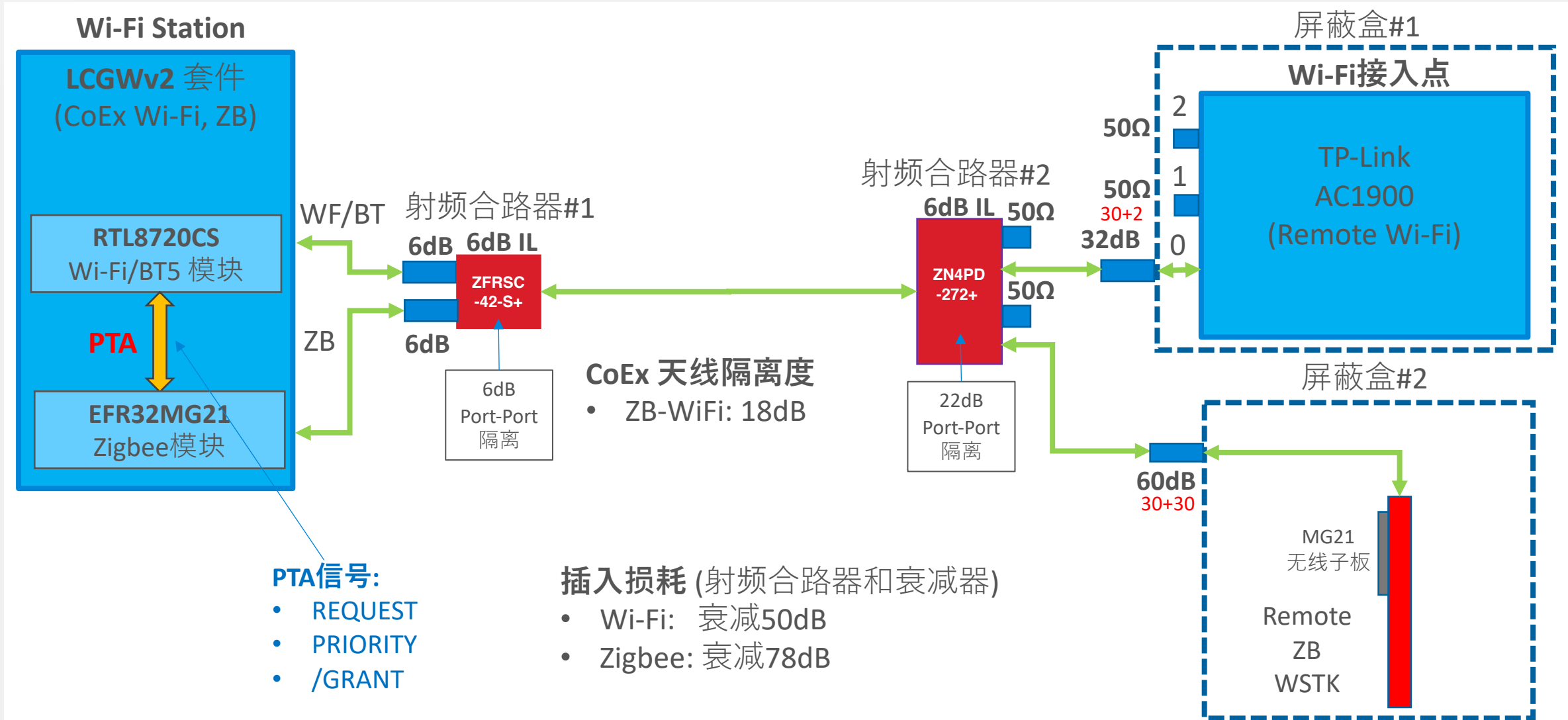
- 将LCGWv2 P2 板子上的 PTA调试引脚连接到逻辑分析器：
  1. REQUEST --- PC02
  2. PRIORITY --- PB00
  3. GRANTn --- PB01
  4. FEM\_TXA --- PD02
  5. FEM\_RXA --- PD03
- 阿里云智能手机应用程序控制Zigbee灯反复亮/灭
  - 捕捉10~20秒的PTA信号
- 测量：
  - 例如：REQUEST -> GRANT响应延迟， REQUEST持续时间



# LCGW PTA共存测试：试验台设置 (1)



# LCGW PTA共存测试：试验台设置 (2)



整个设置，包括笔记本电脑和电源，应在射频屏蔽室内 (约55dB 2.4GHz入口衰减)

# LCGW PTA共存测试: RTL8720CS Wi-Fi, EFR32MG21 Zigbee

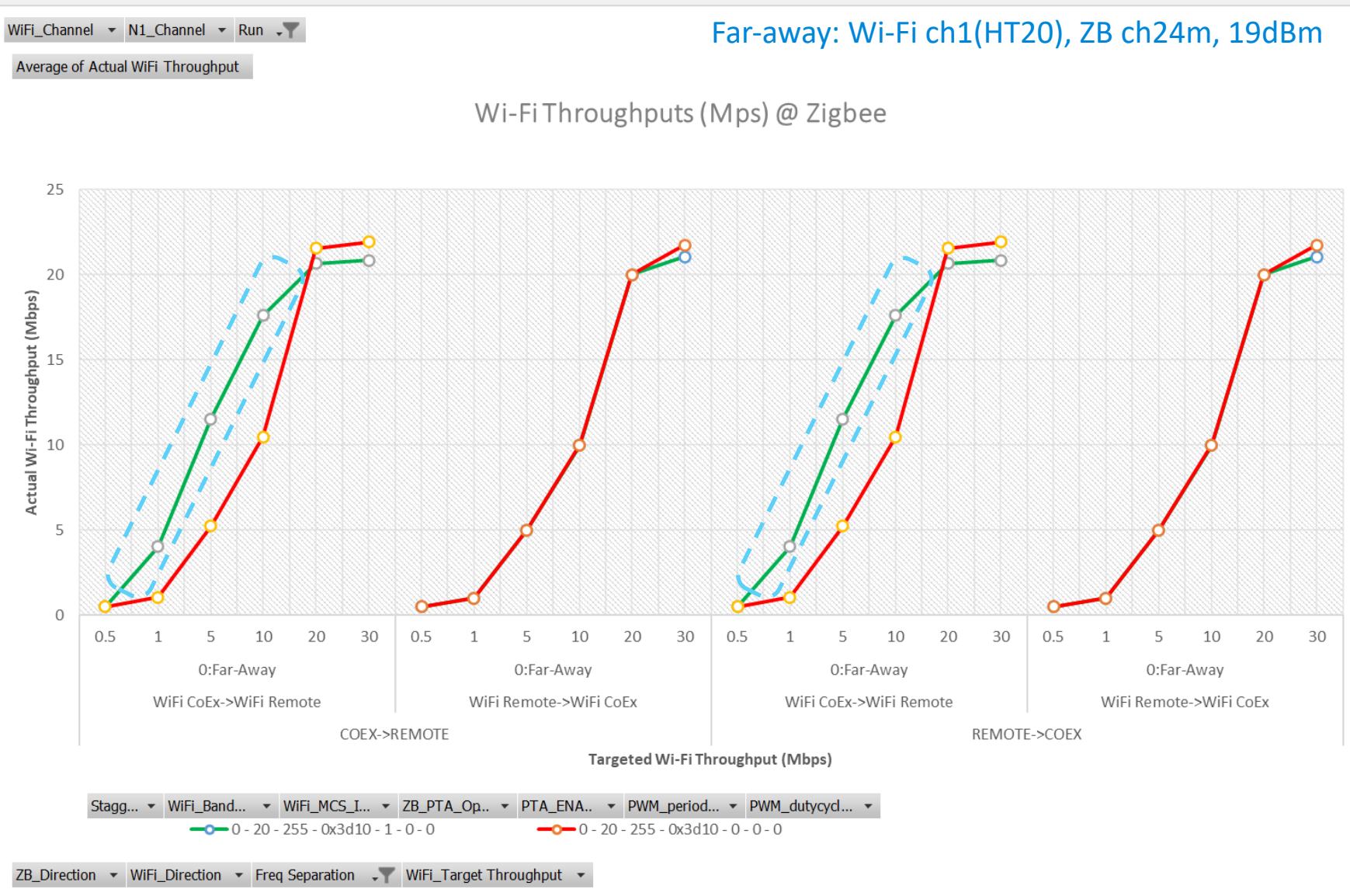
- CoEx Wi-Fi: TX **17.5(Auto)**dBm , Remote Wi-Fi TX **0** dBm , 之间衰减约 : **50**dB
  - 即RTL8720CS 配置 : HT20 , Auto MCS7 , 64QAM rate 5/6
- CoEx Zigbee: TX **19** dBm , Remote Zigbee TX **3** dBm , 之间衰减约: **78**dB
  - 即EFR32MG21 运行特定的SoC PTA测试固件
- 在以下Wi-Fi流量和配置上运行Zigbee coex测试 :
  - Iperf Wi-Fi 流方向 : **Ameba出**(CoEx->Remote) and **Ameba进** (Remote->CoEx)
  - Iperf **TCP**目标吞吐量: **500k, 1M, 5M, 10M, 20M, 30M bits/s**
  - Wi-Fi:
    - 信道: **1, (6, 11 Optional)**, 保护间隔 : 短
    - [带宽, MCS索引]配置: **HT20, Auto MCS**
  - Zigbee:
    - PTA: [启用 , 禁用]
    - CoEx 功率 : **19 (10 可选)** dBm, 远端功率 : **3, (10, 20 可选)** dBm
    - Zigbee流方向 : **CoEx->Remote** 和 **Remote->CoEx**
    - Zigbee信道: **远离**、**邻道**、**重叠**相对于Wi-Fi信道
    - Zigbee**创建网络测试**、**设备加网测试**和**TX/RX吞吐量测试**
    - Zigbee**吞吐量测试配置**: 有效负载**75**字节 , **95**毫秒间隔 , 每次尝试发送**105**个数据包

# LCGWv2(Gen2) Wi-Fi Zigbee PTA共存测试结果

- 测试条件: Wi-Fi (ch1) and Zigbee (ch24) 设置为远离WIFI的信道
  1. Wi-Fi吞吐量 (Mbps)
  2. Zigbee加网成功率 (%)
  3. Zigbee CCA故障率每条消息
  4. Zigbee重试次数每条消息
  5. Zigbee消息失败率 (%)
- 表示方法 :
  - 红色:禁用PTA, 绿色:启用PTA

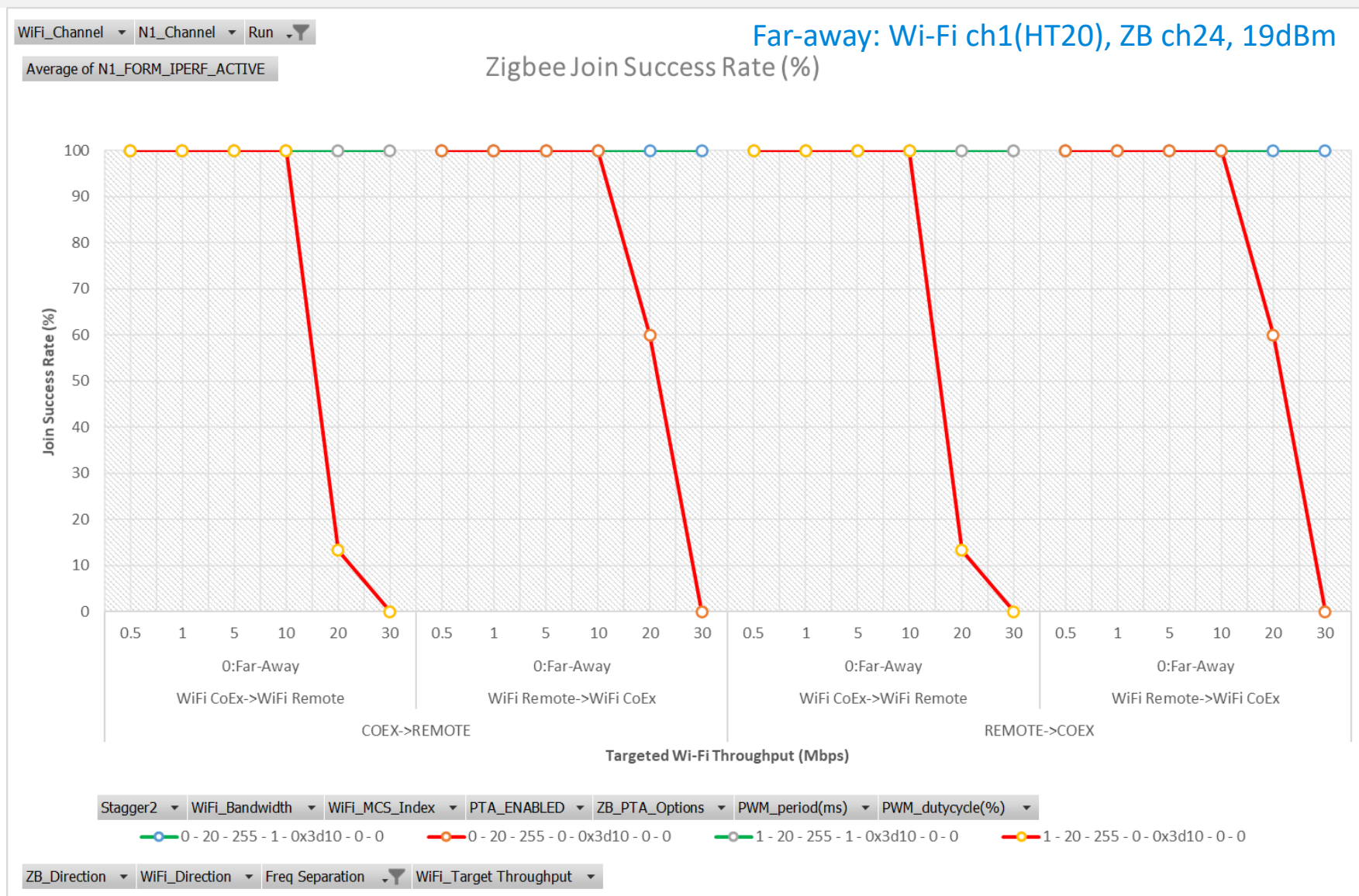
# 1. Wi-Fi吞吐量(Mbps) @Zigbee

Far-away: Wi-Fi ch1(HT20), ZB ch24m, 19dBm



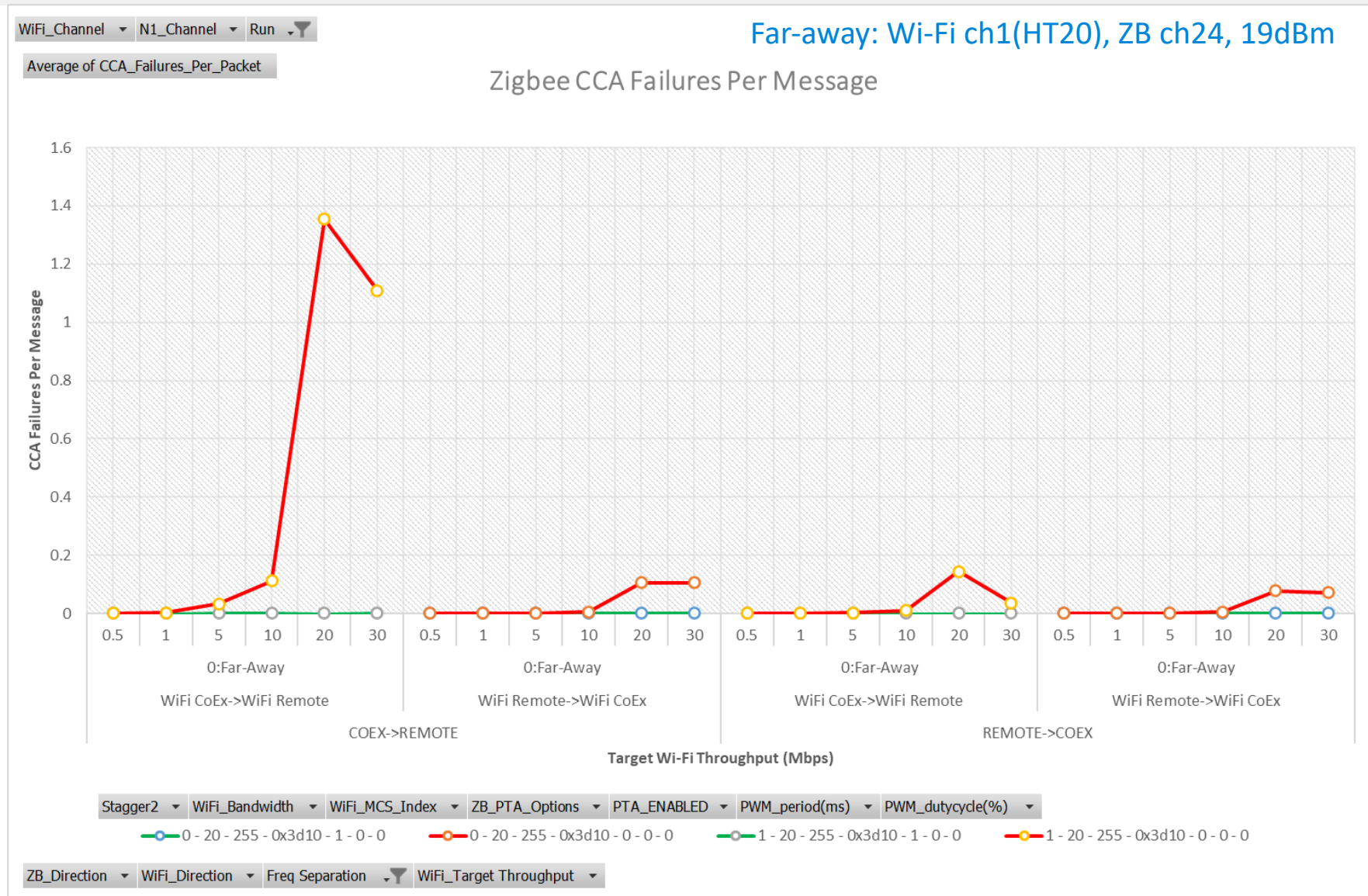
提示: WIFI TCP iperf测试时 Amberd SDK 6.2b 的AT 命令ATWT=-c,-b (客户端), 在PTA 打开的情况下, 有时限制不了WiFi 的带宽。

## 2. Zigbee加网成功率 (%)

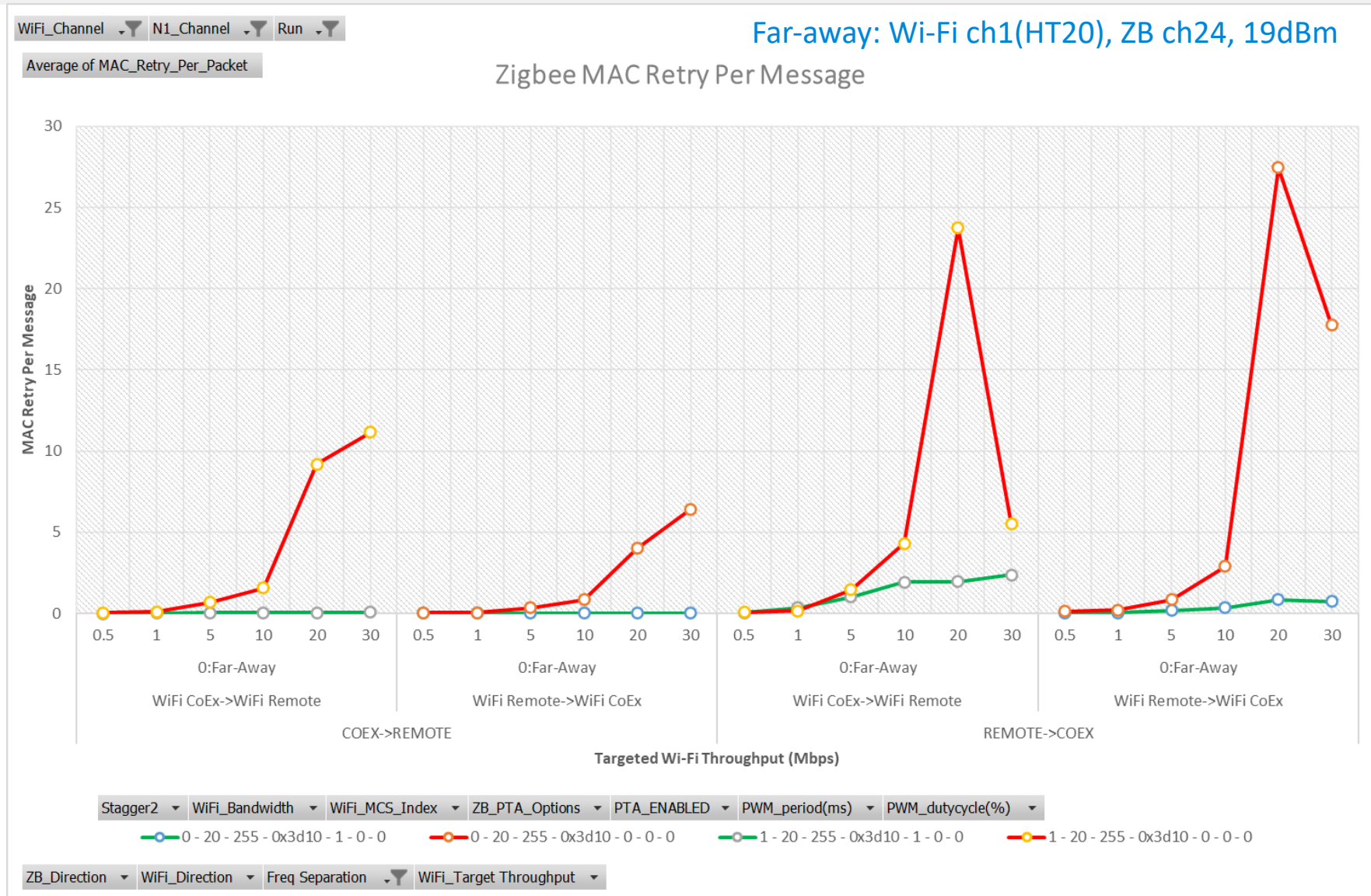




# 3. Zigbee CCA故障率每条消息

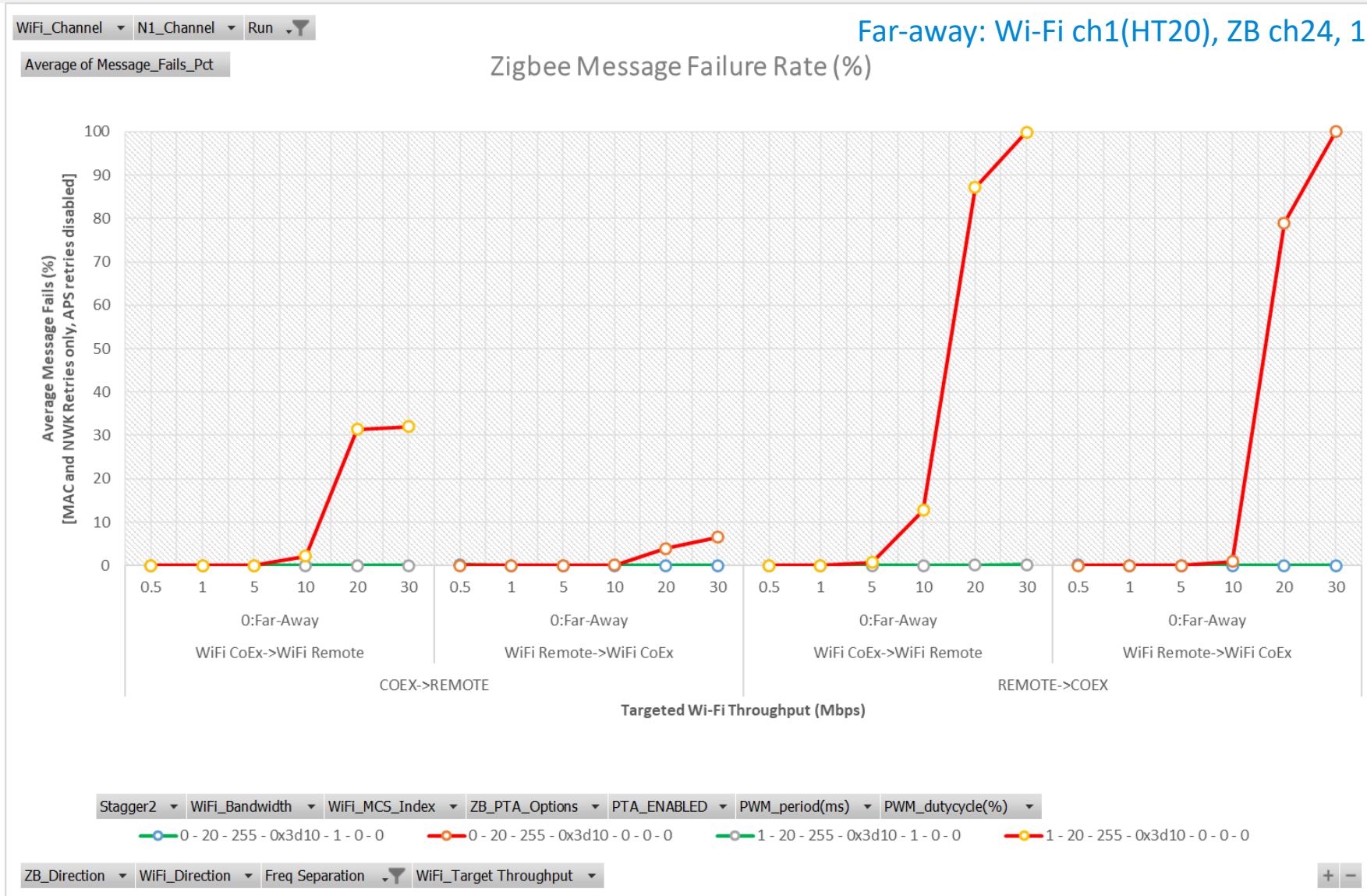


# 4. Zigbee重试次数每条消息



# 5. Zigbee消息失败率 (%)

Far-away: Wi-Fi ch1(HT20), ZB ch24, 19dBm



# 网关云服务集成

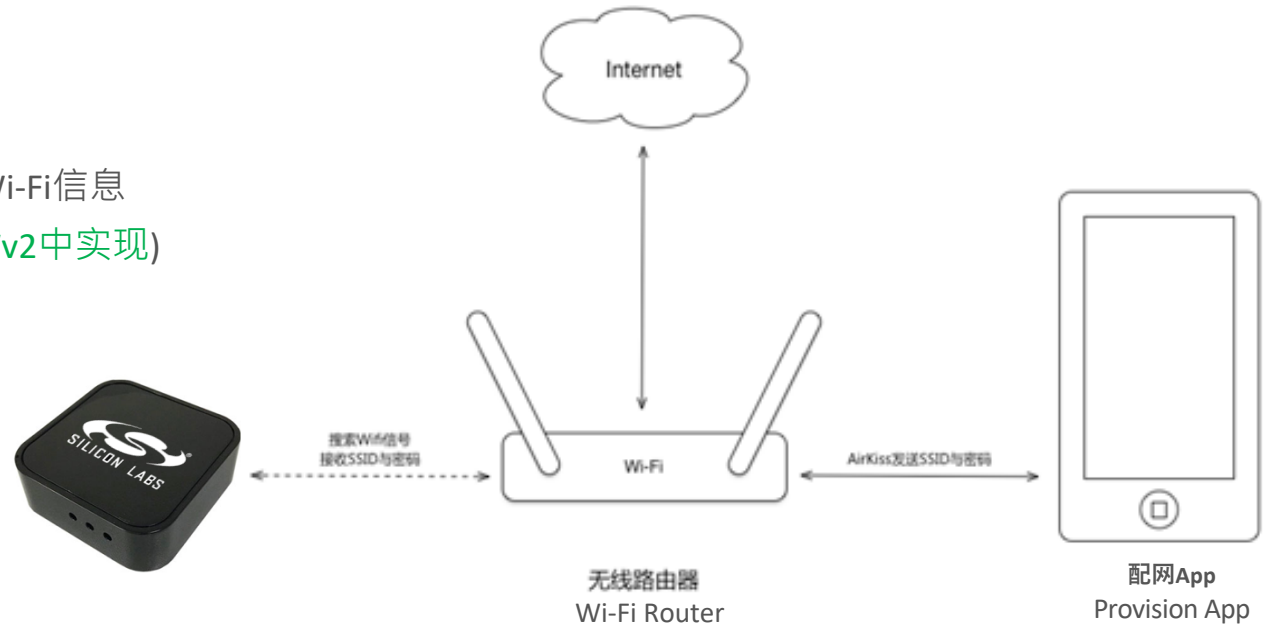
用户指南 : [ug\\_Aliyun\\_iotkit\\_integration\\_of\\_Z3GatewayFreeRTOS\\_for\\_LCGW](#)

WWW.SILABS.COM



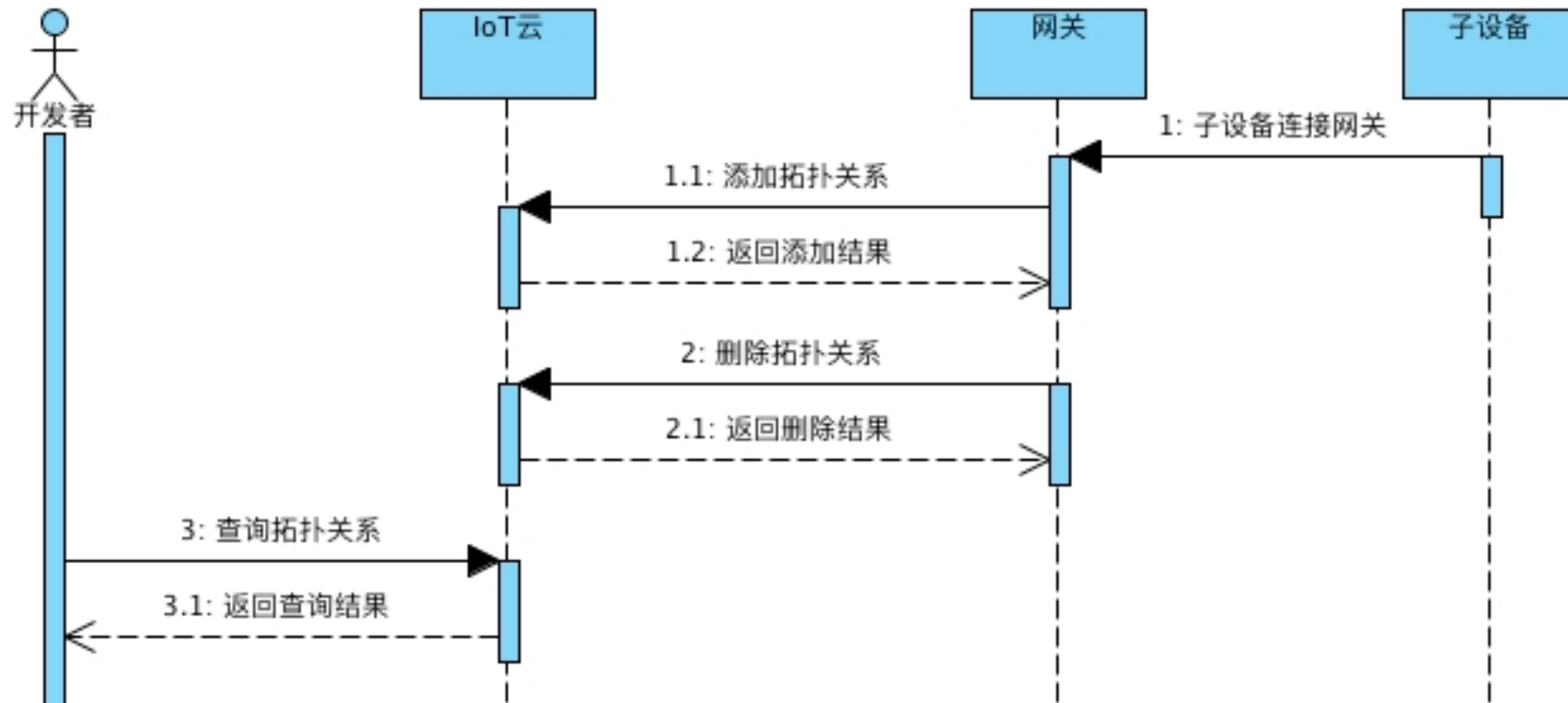
# IoT设备Wi-Fi配网

- **问题**：将IoT设备(无显示屏、无键盘)连接到Wi-Fi网络
- **解决方法**：一般来说，会选择使用移动应用程序为IoT设备进行Wi-Fi配网
  - **阿里巴巴iotkit AWSS Wi-Fi配网方法**：
    1. SmartConfig：通过UDP广播直接给设备提供Wi-Fi信息
    2. 零配：由已配网的设备替新设备来配网
    3. AHA, ADHA：通过手机AP或路由器AP让设备连接获取Wi-Fi信息
    4. **设备AP**：**手机连接设备AP并发送Wi-Fi信息(在LCGWv2中实现)**
  - **Wi-Fi联盟**：
    1. Wi-Fi保护设置 (WPS) (**有安全问题, Wi-Fi路由器上已禁用**)
    2. Wi-Fi Easy Connect：二维码，支持WPA2、WPA3
  - **德州仪器**：SmartConfig
  - **腾讯**：AirKiss 2.0 (类似于TI SmartConfig)
  - **带外配网**：
    - **通过BLE为Wi-Fi配网**：例如 **Realtek RTL872xCS/DN Wi-Fi/BLE combo SoC** (瑞昱SDK示例)
    - **通过NFC为Wi-Fi配网**：例如 TI SimpleLink NFC+SimpleLink NFC
  - **DMZ (Demilitarized Zone)/来宾网络**：例如预定义的ssid/pwd



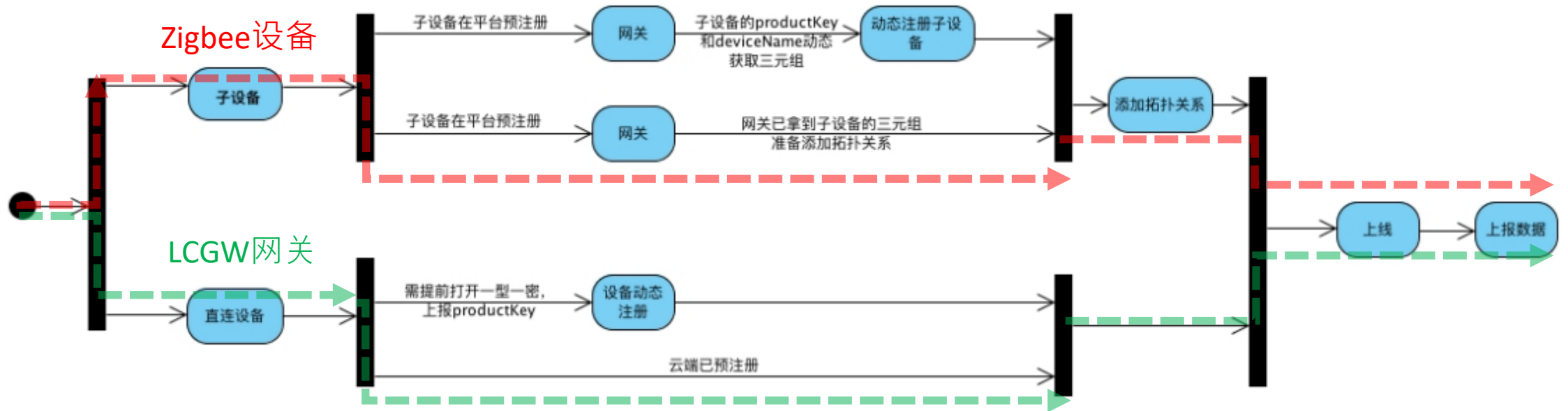
# iotkit-embedded C-SDK : 网关、子设备拓扑关系

1. 子设备连接到网关后，调用add\_subdev()添加拓扑关系，物联网平台返回结果。
2. 子设备离网/网关删除子设备，调用dev\_logout()删除彼此拓扑关系，物联网平台返回结果。
3. 开发者可以调用[GetThingTopo](#)接口来查询网关和子设备的拓扑关系。



# iotkit-embedded C-SDK : 设备上线

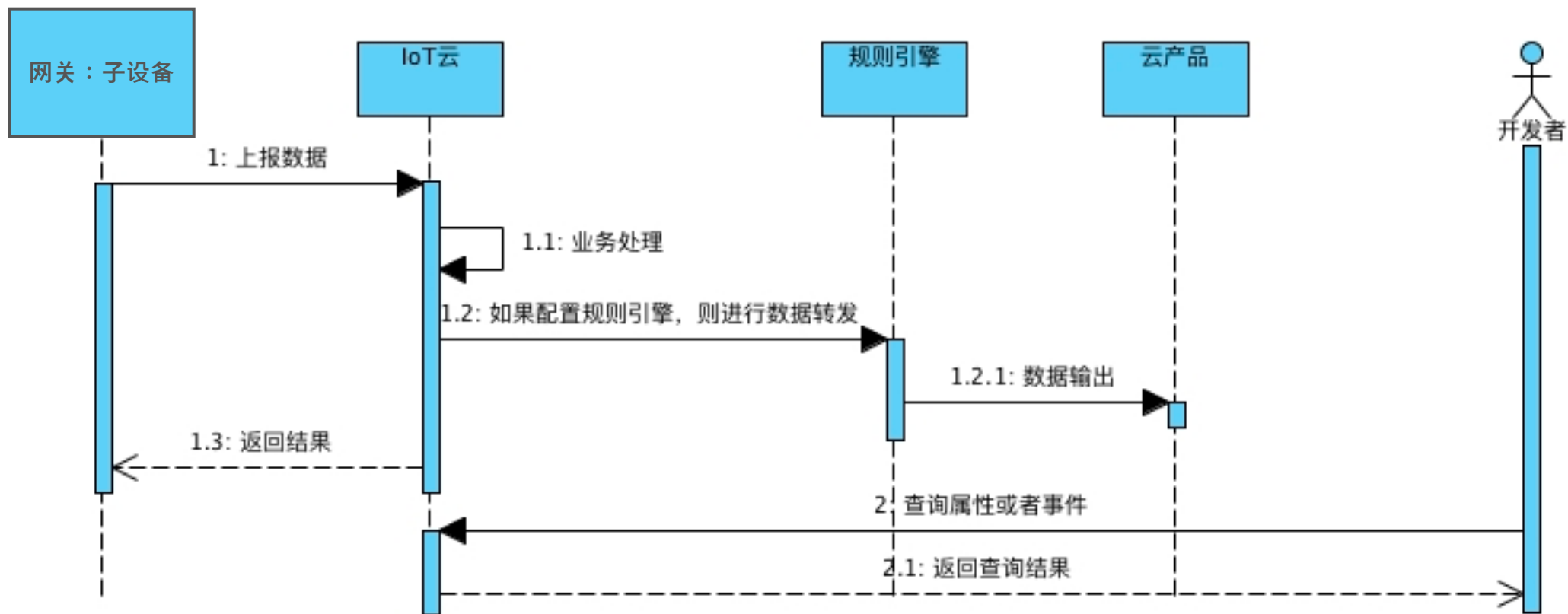
参考: [https://help.aliyun.com/document\\_detail/90459.html](https://help.aliyun.com/document_detail/90459.html)



- 设备上线流程类型：
  - (1) Wi-Fi直连设备接入：例如**LCGW网关**, (2) 子设备接入：例如**Zigbee设备**
- Wi-Fi直连网关:
  1. 通过Alink JSON(ICA数据标准化数据格式)与阿里云 Link平台通信(**LCGW实现**)，或：
  2. 运行自己的透传/自定义数据模型并在云端中实现其与Alink数据的转换服务 (**未实现**)
- 注意：阿里云不关心网关与其子设备之间的底层数据模型。

# iotkit-embedded C-SDK : Zigbee设备上报属性和事件

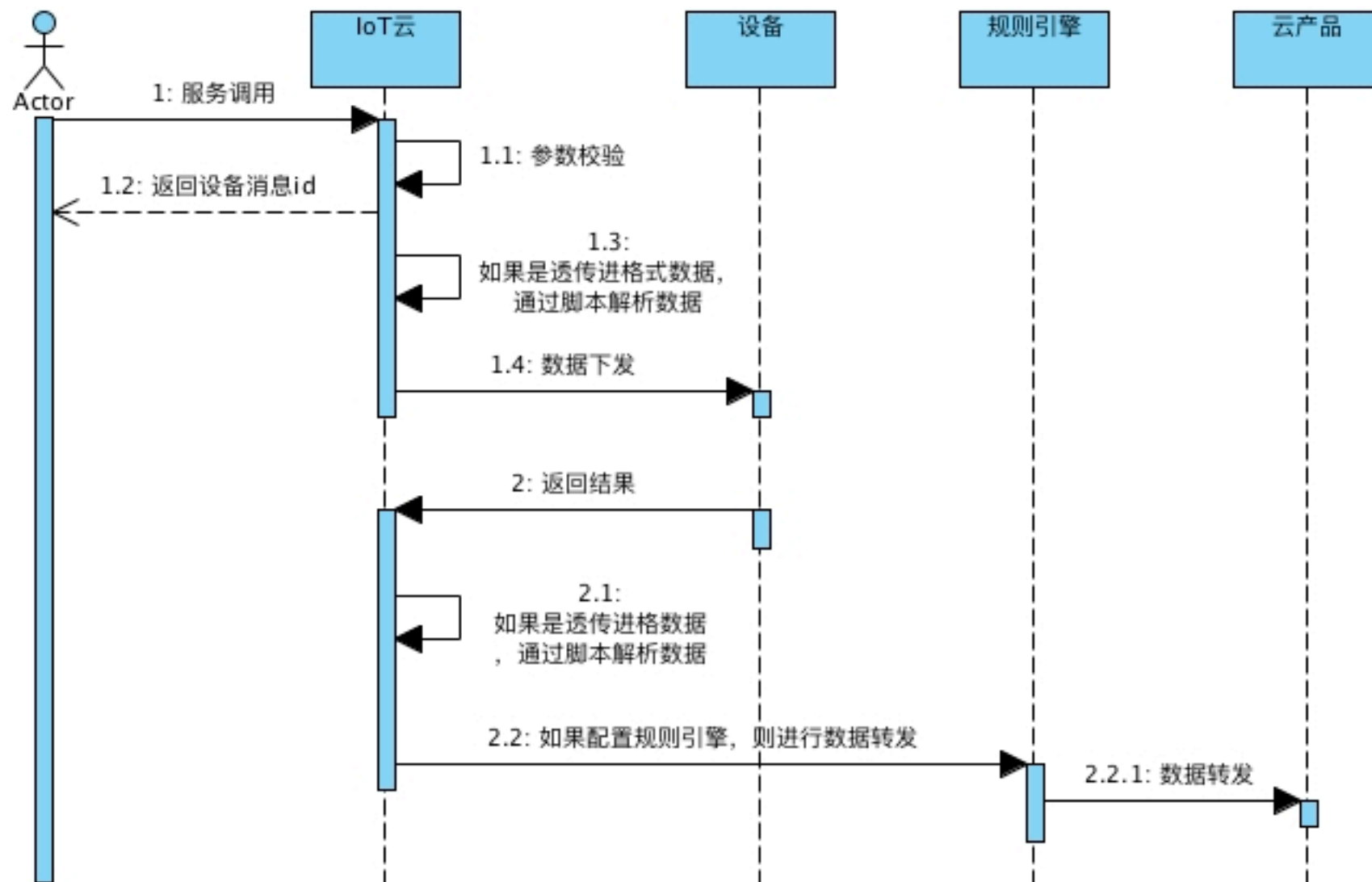
- 通过Wi-Fi网关来代理子设备的属性和事件的上报：
  - 方法一：上传Alink JSON (ICA数据标准化)格式的数据到阿里云物联网平台(LCGW实现)
    - LCGW：实现zcl→Alink的转换、设备的属性和事件的上报
  - 方法2：上传透传/自定义模型数据，即zcl，并在云中实现到Alink数据的转换服务(未实现)





# iotkit-embedded C-SDK : 设置设备属性/调用设备服务

- 异步调用：
  - 设置设备属性
  - 调用设备服务
- LCGW：
  - ALink -> zcl
  - 调用CMD\_CTRL API
  - 发布消息到底层库
- Z3GatewayFreeRTOS库
  - 向ZB设备发送单播消息



# lib\_iotkit、lib\_z3ctrl和cloud gateway主应用程序

- 移植了阿里巴巴的iotkit-embedded C-SDK v3.0.1作为iotkit库，即lib\_iotkit.a，其源代码位于：
  - project/realtek\_amebaD\_va0\_example/GCC-RELEASE/project\_hp/asdk/make/project/lib\_iotkit/
- Simplicity Studio生成的Z3GatewayFreeRTOS库，即lib\_z3ctrl.a，其源代码位于：
  - project/realtek\_amebaD\_va0\_example/GCC-RELEASE/project\_hp/asdk/make/project/lib\_z3ctrl/
- **lib\_iotkit**和**lib\_z3ctrl**之间的交互在**cloud gateway**应用程序中实现，其源代码位于：
  - project/Realtek\_amebad\_va0\_example/

```
├──src_hp  
│   board_io.c  
│   cloud_gateway.c  
│   cloud_gateway_interface.c  
│   main.c  
│   Makefile  
│   wifi_prov.c  
└──inc_hp  
    FreeRTOSConfig.h  
    main.h  
    platform_autoconfig.h  
    platform_opts.h
```

# Cloud Gateway主应用程序 (1)

## ▪ *main.c*

- main() 调用start\_button\_polling()和start\_cloud\_gateway()来启动按钮轮询任务和cloud gateway初始化的任务

## ▪ *cloud\_gateway.c*

- 定义了LCGW和Zigbee设备注册阿里云要使用的三元组 (请替换您自己设备的三元组)
- cloud gateway初始化：初始化看门狗定时器，等待Wi-Fi连接成功后调用startGateway()，延时3秒并创建Zigbee网络，然后替LCGW向云注册，创建cloud\_msg\_center和stack\_msg\_center任务
- 实现lib\_iotkit库的user\_dispatch\_yield()、cloud\_msg\_center()、stack\_msg\_center()等任务来处理云事件和对Zigbee设备管理(如注册到云)和设备控制/报告等。
- 实现button\_evt\_polling\_center()并在cloud\_msg\_center任务中调用button\_evt\_hdl()对按键事件进行处理

## ▪ *cloud\_gateway\_interface.c*

- 实现云和Zigbee协议栈消息/事件队列的分派和处理程序
- 实现zcl<->ALink数据格式转换和对应的接口函数，并调用lib\_z3ctrl库的API进行处理

# Cloud Gateway主应用程序 (2)

- **wifi\_prov.c**
  - 包含了通过手机应用进行Wi-Fi配网的初始化功能：
    - 阿里巴巴提供的设备AP Wi-Fi配网方式 (在LCGWv2中实现)
- **board\_io.c**
  - 定义了RTL8720CS用于Zigbee NCP复位和bootload控制、CONFIG按钮读取和RGB LED指示灯控制的GPIO引脚
  - 实现了按钮轮询和LED闪烁的驱动程序
- **cloud\_gateway.h**
  - 包含了cloud gateway各功能模块的编译选项
- **platform\_opts.h**
  - 包含了AmebaD系统各种Cloud Gateway应用程序相关的特性的编译选项
- **platform\_opts\_bt.h**
  - 包含了AmebaD-BLE角色和启用/禁用BLE功能的编译选项
- **参考** : *ug\_Aliyun\_iotkit\_integration\_of\_Z3GatewayFreeRTOS\_for\_LCGW 1.7 for the compile options details*

# Zigbee ZCL和阿里巴巴/ICA TSL模型转换 (1)

- 阿里云IOT使用Alibaba/ICA定义的TSL(事物规范语言)模型，即ALink协议
- Zigbee设备遵循Zigbee联盟定义的ZCL(Zigbee集群库)作为应用协议
- LCGW选择了使用Alink协议和阿里云交互，并在cloud\_gateway\_interface.c中实现Alink<->ZCL之间的转换
- 示例：下行属性设置

```
/*
 * property setting handler used by cloud msg center
 */
static int16_t lightswitch_property_set(uint16_t value, EUI64_T eui64, uint8_t endpoint)
{
    // send msg to stack
    if (-1 == lightswitch_set_interface(value, eui64, endpoint)) {
        ERROR_TRACE("failed.");
        return -1;
    }
    return 0;
}
```

# Zigbee ZCL和阿里巴巴/ICA TSL模型转换 (2)

- 其中

```
int16_t lightswitch_set_interface(uint16_t value, Eui64_t eui64, uint8_t endpoint)
{
    uint16_t on_off;

    if (0 == value) {
        on_off = CLUSTER_ON_OFF_CMD_ID_OFF;
    } else {
        on_off = CLUSTER_ON_OFF_CMD_ID_ON;
    }

    if (-1 == cmd_ctrl(&eui64, CLUSTER_ID_ON_OFF, on_off, NULL)) {
        ERROR_TRACE("failed.");
        return -1;
    }
    return 0;
}
```

- 参考：[ug\\_Aliyun\\_iotkit\\_integration\\_of\\_Z3GatewayFreeRTOS\\_for\\_LCGW section 2.3](#) for more conversion examples

# 阿里云：1. 创建新项目

- 阿里云有2个物联网平台：(1) [iot.aliyun.com](https://iot.aliyun.com)物联网平台, (2) [living.aliyun.com](https://living.aliyun.com)生活物联网平台
- 免费注册阿里云(两个平台通用)，之后登录 <https://living.aliyun.com> (注：仅有中文用户界面)
- 创建新项目，例如：“Z3\_RTOS\_GW”

The screenshot shows the living.aliyun.com web interface. At the top, there is a navigation bar with '生活物联网平台' and '中国站' on the left, and '账单中心', '文档中心', and '工单' on the right. A green banner below the navigation bar contains an announcement: '【更新公告】国际站的自有App方案全面升级, APP SDK围绕海外统一激活中心提供了账号、消息推送、服务器切换的完整方案 点击查看'. Below the banner, there are three tabs: '全部项目', '自建项目', and '授权项目'. A blue button labeled '创建新项目' is circled in red. Below the tabs, there are two project cards. The first card is for 'si\_app\_deno' with ID 'a124OMsvBLHaKMBe', showing 0 devices, 0 apps, and 1 member, created on 2019-09-23. The second card is for '新手引导项目' with ID 'a1244mk2oTuzd5eb', showing 1 device, 0 apps, and 2 members, created on 2019-09-20. On the right side, there is a '激活码总览' section showing 0 purchased and 0 remaining activation codes.

# 阿里云：2. 为LCGW创建新产品 (1)

- 进入项目，创建新产品： 创建网关产品
  - 所属分类=“网关”，
  - 节点类型=“网关”，
  - 通讯方式=“WIFI”，
  - 数据格式=“ICA标准数据格式(ALink JSON)”，
  - 使用ID<sup>2</sup>认证=“否”

新建产品

产品信息

\* 产品名称  
Gateway

\* 所属分类  
网络设备 / 网关

节点类型

\* 节点类型  
 设备  网关

连网与数据

\* 连网方式  
WiFi

\* 数据格式  
ICA 标准数据格式 (Alink JSON)

\* 使用 ID<sup>2</sup> 认证  
 是  否

完成 取消

- 获取产品的 Product Key和Product Secret

生活物联网平台 (中国站) Z3\_RTOS\_GW

账单中心 文档中心 工单

Z3\_RTOS\_GW 项目设置

产品总数 0 项目激活码 0 已激活设备 0 当前在线设备 0 量产管

产品管理 已在阿里云物联网平台创建过产品? 关联产品 **创建新产品**

App 管理  
平台提供公测 App, 同时支持自有控制产品  
云智能 App

零基础搭建智能产品

生活物联网平台 (中国站) Z3\_RTOS\_GW

账单中心 文档中心 工单

Z3\_RTOS\_GW > Gateway

1 功能定义 2 设备调试 3 人机交互 4 批量投产

功能定义

标准功能

功能类型	功能名称	标识符	数据类型	数据定义	操作
无标准功能					

Gateway

更新时间: 2020-01-07 17:16:39

基本信息 编辑

所属分类: 网关  
节点类型: 网关  
通讯方式: WIFI  
数据格式: ICA标准数据格式 (推荐)  
Product Key: a16MCIWRJQ  
Product Secret: \*\*\*\* **显示**  
Product ID: 3000005  
认证方式: 设备密钥

模组 重选

品牌: 未认证  
型号: 未认证

创建时间: 2020-01-07



# 阿里云：2. 为LCGW创建新产品 (2)

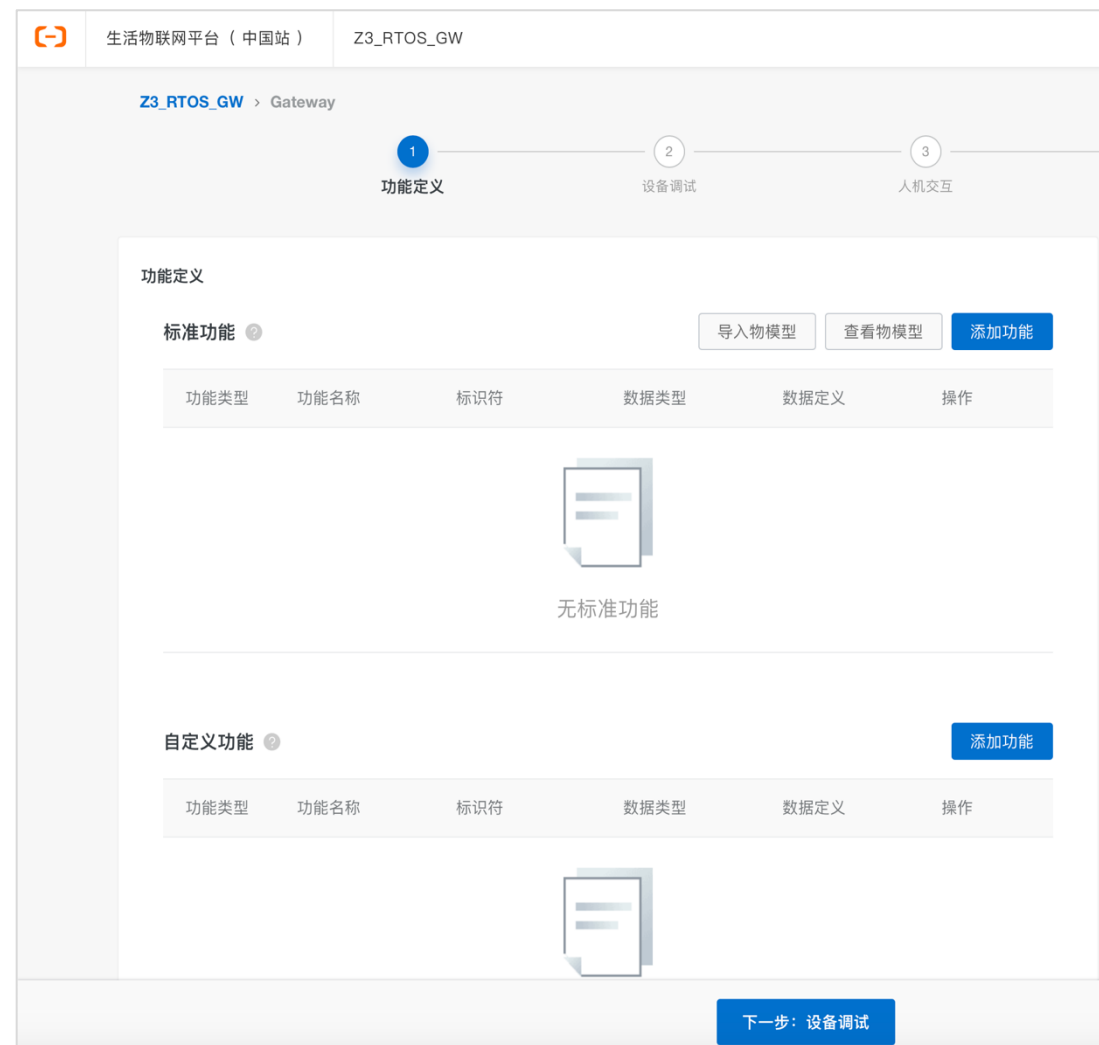
- 进入产品界面，进行以下配置

## 1. 功能定义

- 标准功能 = 无标准功能
- 自定义功能 = 无

## 2. 设备调试

- 模组信息 = 无



# 阿里云：2. 为LCGW创建新产品 (3)

## 3. 人机交互

- 使用公版App控制产品 = 是
- 面板选择：设置产品的控制面板
- 保存配网+App下载二维码，以便后续扫码添加该产品类别的设备
- 分享方式：使用公版APP时，默认“授权式”且不能改
- 多语言管理：设置产品在APP中显示的名称和其功能的名称
- 配网引导：
  - 默认配网方式 = 设备热点配网
  - 备选配网方式 = 无
  - 扫码/本地自动发现：Y 默认
  - 支持手动选择产品列表/搜索：Y
  - 编辑引导页面：设置配网引导页面显示图片和文字
- 天猫精灵接入 = 是



# 阿里云：2. 为LCGW创建新产品 (4)

## 4. 批量投产: 名称修改 (可重新修改产品名称)

Z3\_RTOS\_GW / Gateway

功能定义 设备调试 人机交互 批量投产

确认产品信息 批量生产

**确认产品基本信息**



**基础信息**

所属分类: 网关  
通讯方式: WiFi  
数据格式: ICA标准数据格式  
ProductKey: a16MCIWRJjQ

**模组**

品牌: 未认证  
型号: 未认证

确认云智能APP中需要展示的产品名称已准确填写 [查看规范](#)



中文名称: Z3Gateway0001  
英文名称: Z3 Gateway 0001

[名称修改](#)

扫码预览产品

**1 发布产品**

我充分了解并同意《[开发者须知](#)》和《[授权协议](#)》

[开发完成](#)

**2 购买激活码**

[去购买](#)

**3 少量试产**

建议在名称审核前少量试产，确认审核通过后再进行正式量产及物料制作。

[量产管理](#)

**4 产品名称审核**

至少量产1件设备后进入审核流程

**5 批量生产**

量产烧录麻烦? 推荐使用 [阿里云IoT设备证书分发工具](#)

参考：[https://help.aliyun.com/document\\_detail/126365.html](https://help.aliyun.com/document_detail/126365.html)

# 阿里云：3. 为Zigbee灯创建新产品 (1)

- 创建新产品：为Zigbee灯创建新的产品并获取其**Product Key**和**Product Secret**：
  - 所属分类 = “灯”，节点类型 = “设备”，是否接入网关 = “是”，接入网关协议 = “Zigbee”，数据格式 = “ICA标准数据格式(ALink JSON)”，使用ID<sup>2</sup>认证 = “否”
- 1. 功能定义 -> 标准功能：默认“主灯开关”属性和选取“明暗度”属性

功能定义

标准功能 ? 导入物模型 查看物模型 添加功能

功能类型	功能名称	标识符	数据类型	数据定义	操作
属性	主灯开关 <span>必选</span>	LightSwitch	bool (布尔型)	布尔值: 0 - 关闭 1 - 开启	编辑
属性	明暗度 <span>可选</span>	Brightness	int32 (整数型)	取值范围:0 ~ 100	编辑 删除
事件	故障上报 <span>必选</span>	Error	-	事件类型:故障	编辑

**ALinkZigbeeLight**  
更新时间: 2020-04-14 01:19:53

**基本信息**  
所属分类: 灯  
节点类型: 设备  
通讯方式: ZigBee  
数据格式: ICA标准数据格式 (推荐)

Product Key: a1cSTtZ7TXM  
Product Secret: 66HP98S8G6f5GlrB 复制 隐藏  
Product Id: 3028644  
认证方式: 设备密钥

创建时间: 2019-10-26

2. 设备调试 -> 无需设置

# 阿里云：3. 为Zigbee灯创建新产品 (2)

## 3. 人机交互

- 使用公版App控制产品 = 是
- 面板选择：设置产品的控制面板
- 保存配网+App下载二维码，以便后续扫码添加该产品类别的设备
- 分享方式：使用公版APP时，默认“授权式”且不能改
- 多语言管理：设置产品在APP中显示的名称和其功能的名称
- 配网引导：

- 扫码/本地自动发现：Y 默认
- 支持手动选择产品列表/搜索：Y
- 配网引导图：设置引导图片
- 配网文案：设置配网提示信息
- 按钮文案：设置按键显示信息

# 阿里云：3. 为Zigbee灯创建新产品 (3)

- 自动化和定时：选择可用于自动化控制的产品的功能属性，设置是否支持启动产品本身的定时和倒计时功能  
这里选择允许主灯开关和明暗度作为条件和执行

### 自动化与定时

产品联动功能设置 [功能参数](#)

功能名称	智能场景 ?	本地定时 ?	本地倒计时 ?
主灯开关	<input checked="" type="checkbox"/> 作为条件 <input checked="" type="checkbox"/> 作为执行	<input type="checkbox"/> 开启	<input type="checkbox"/> 开启
明暗度	<input checked="" type="checkbox"/> 作为条件 <input checked="" type="checkbox"/> 作为执行	<input type="checkbox"/> 开启	
故障上报	<input type="checkbox"/> 作为条件 <input type="checkbox"/> 作为执行		

- 天猫精灵接入 = 是

## 4. 批量投产 -> 名称修改(可重新修改产品名称)

# 阿里云：4. 为Zigbee开关创建新产品 (1)

- 创新建产品：为Zigbee开关创建新的产品并获取其**Product Key**和**Product Secret**：
  - 所属分类 = “入墙开关”，节点类型 = “设备”，是否接入网关 = “是”，接入网关协议 = “Zigbee”，数据格式 = “ICA标准数据格式 (ALink JSON)”，使用ID<sup>2</sup>认证 = “否”
- 1. 功能定义 -> 标准功能：默认“电源开关\_1”属性

功能定义

标准功能 ? 导入物模型 查看物模型 添加功能

功能类型	功能名称	标识符	数据类型	数据定义	操作
属性	电源开关_1 <span>必选</span>	PowerSwitch_1	bool (布尔型)	布尔值: 0 - 关闭 1 - 开启	编辑
事件	故障上报 <span>必选</span>	Error	-	事件类型:信息	编辑

## AlinkZigbeeWallSwitch

更新时间: 2020-05-31 23:35:28

### 基本信息

所属分类: 入墙开关

节点类型: 设备

通讯方式: ZigBee

数据格式: ICA标准数据格式 (推荐)

Product Key: a1pDuSGqiOf

Product Secret: T0IWt2HYZqfhzFHK [复制](#) [隐藏](#)

Product Id: 4985089

认证方式: 设备密钥

创建时间: 2020-05-30

2. 设备调试 -> 无需设置

# 阿里云：4. 为Zigbee开关创建新产品 (2)

## 3. 人机交互

- 使用公版App控制产品 = 是
- 面板选择：设置产品的控制面板
- 保存配网+App下载二维码，以便后续扫码添加该产品类别的设备
- 分享方式：使用公版APP时，默认“授权式”且不能改
- 多语言管理：设置产品在APP中显示的名称和其功能的名称
- 配网引导：

- 扫码/本地自动发现：Y 默认
- 支持手动选择产品列表/搜索：Y
- 配网引导图：设置引导图片
- 配网文案：设置配网提示信息
- 按钮文案：设置按键显示信息

RtosGateway / AlinkZigbeeWallSwitch / 人机交互 / 配网引导

**配网入口**

扫码/本地自动发现

支持手动选择产品列表/搜索 勾选后，产品名称需审核，审核通过后，用户可在配网列表中看到您的设备

**配网图文**

标准配网是阿里云IoT提供的配网方案，会根据当前环境自动选择最佳的配网方案。

中文 英文 西班牙语 法语 俄语 德语 日语 韩语 印地语 意大利语 葡萄牙语

**引导页**

**配网引导图**

建议尺寸：600px \* 600px 支持png、jpg、gif格式，大小不超过1M

**配网文案：**

先把设备恢复出厂设置，点击下一步，网关的红色指示灯会快速闪烁，并允许设备加网。

**按钮文案：**

**帮助页**

开启后页面上会显示帮助入口，可配置相关帮助信息协助用户配网。



功能定义 设备调试 3 人机交互 4 批量投产

选择交互 配置项默认用于您创建的自有APP，如启用公版APP，相关配置可同时用于自有APP和公版

端 GatewayDemo (默认) 使用公版App控制产品  可以直接从应用市场下载公版App，用于控制智能设备。

**配置App功能**

\* 面板选择 必填 自有APP如何集成产品面板？[立即查看](#)

**界面预览**

使用公版 App 扫码体验产品界面



预览界面专用

未下载公版App? [立即安装](#)

**选择产品图标**



**App资料下载**

\* 分享方式 分享方式包括：抢占式、授权式和共享式。  必填

\* 多语言管理 可以修改在App中展示的产品和功能名称，并可编辑多语言，包括英语、西班牙语、法语、俄语、德语、日语、韩语、印地语、意大利语

\* 配网引导 为该产品选择配网方案，并填写配网引导的图文详情。

**设备告警** 自定义设备的告警条件，当设备出现异常时，第一时间自动通知用户或企业，实时监管设备状态

**自动化和定时** 选择智能场景、自动化、云端定时、本地定时和本地倒计时中可用的设备功能。

**天猫精灵** (适用于国内) 接入后可以通过天猫精灵音箱控制该产品。



# 阿里云：4. 为Zigbee开关创建新产品 (3)

- 自动化和定时：选择可用于自动化控制的产品的功能属性，设置是否支持启动产品本身的定时和倒计时功能  
这里选择允许电源开关\_1作为条件

### 自动化与定时

产品联动功能设置 [功能参数](#)

功能名称	智能场景 ?	本地定时 ?	本地倒计时 ?
电源开关_1	<input checked="" type="checkbox"/> 作为条件 <input type="checkbox"/> 作为执行	<input type="checkbox"/> 开启	<input type="checkbox"/> 开启
故障上报	<input type="checkbox"/> 作为条件 <input type="checkbox"/> 作为执行		

- 天猫精灵接入 = 是

## 4. 批量投产 -> 名称修改(可重新修改产品名称)

# 阿里云：5. 定制阿里云网关示例以连接自己的产品 (1)

- 分别在创建好的网关、灯和开关项目中的“设备调试”界面，添加对应新的测试设备：
  - 阿里云允许每个产品**最多免费添加50个测试设备**
  - 为测试设备输入设备名称，例如alink\_gw\_0004

### 新增测试设备

DeviceName可以是MAC地址、IMEI号或自定义SN等，须确保产品下唯一，为空将由系统自动颁发，您可以烧录到设备中，并上报到云端进行鉴权认证。

DeviceName ?

请输入DeviceName，为空将由系统自动颁发

- 在测试设备列表中，点击“新增测试设备”按钮查看网关、灯和开关的测试设备的三元组：

### 查看设备激活凭证

设备激活凭证，请烧录到设备中

ProductKey:	a1VtEtCqVsx	复制
DeviceName:	alink_gw_0004	复制
DeviceSecret:	O29fUQif5JXIKUxCLo3ToxfEKLuthcxa	复制

The screenshot shows the 'Z3\_RTOS\_GW' gateway configuration page in the IoT Platform console. The '测试设备' (Test Devices) section is visible, showing a table with columns for DeviceName, Status, Last Online Time, and Actions. A blue button labeled '新增测试设备' (Add Test Device) is circled in red. Below the table, a message indicates '暂无测试设备' (No test devices). At the bottom, there are buttons for '上一步：功能定义' (Previous Step: Function Definition) and '下一步：人机交互' (Next Step: Human-Machine Interaction).

## 阿里云：5. 定制阿里云网关示例以连接自己的产品 (2)

- LCGW网关示例使用一机一密，且为了演示方便，不需要Zigbee子设备具备三元组信息，网关里会预先存储所支持的产品类别的多组设备的三元组信息，由网关动态地把三元组分配给入网的子设备，并帮其登录到云端。
- 将自己创建的产品的设备三元组替换cloud\_gateway.c中的默认三元组：
  - Zigbee 设备的产品类型: `char *product_type_cloud_define[PRODUCT_TYPE_MAX] = {product_key,... }`
  - 网关设备： `_product_key[], _product_secret[], _device_name[], _device_secret[]`
  - Zigbee设备： `const CLOUD_DEV_META_T subdevArr[EXAMPLE_SUBDEV_MAX_NUM] = { { product_key, product_secret, device_name, device_secret },... }`  
代码中EXAMPLE\_SUBDEV\_MAX\_NUM默认为4，数组中存储了3个灯和1个开关的三元组信息。
- 重新编译AmebaD的网关示例并生成固件，并将其烧写到LCGWv2硬件中...

# 阿里云：6. 云智能手机应用 - 添加LCGW设备 (1)

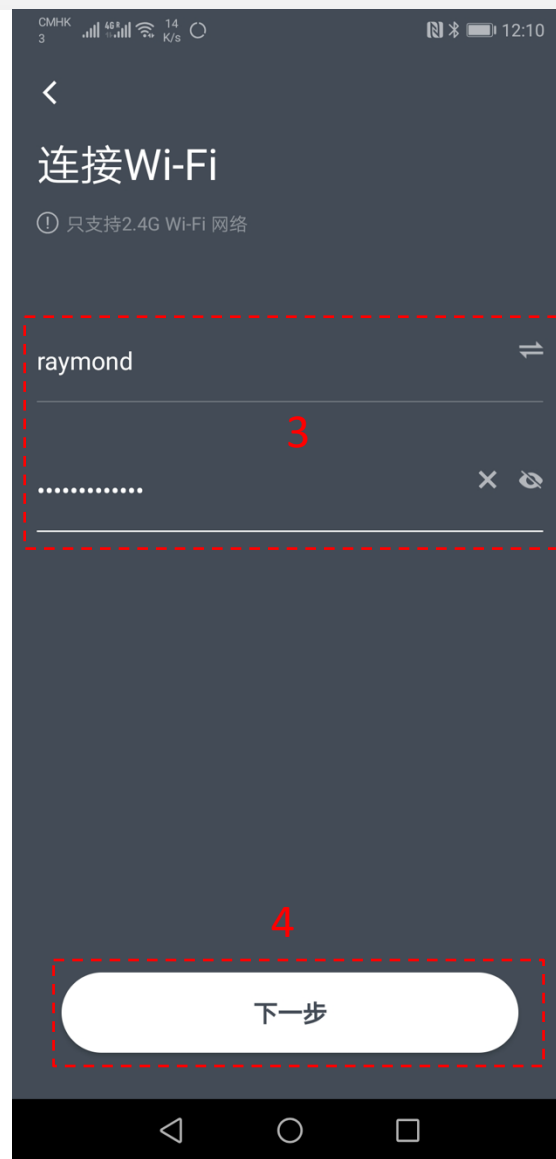
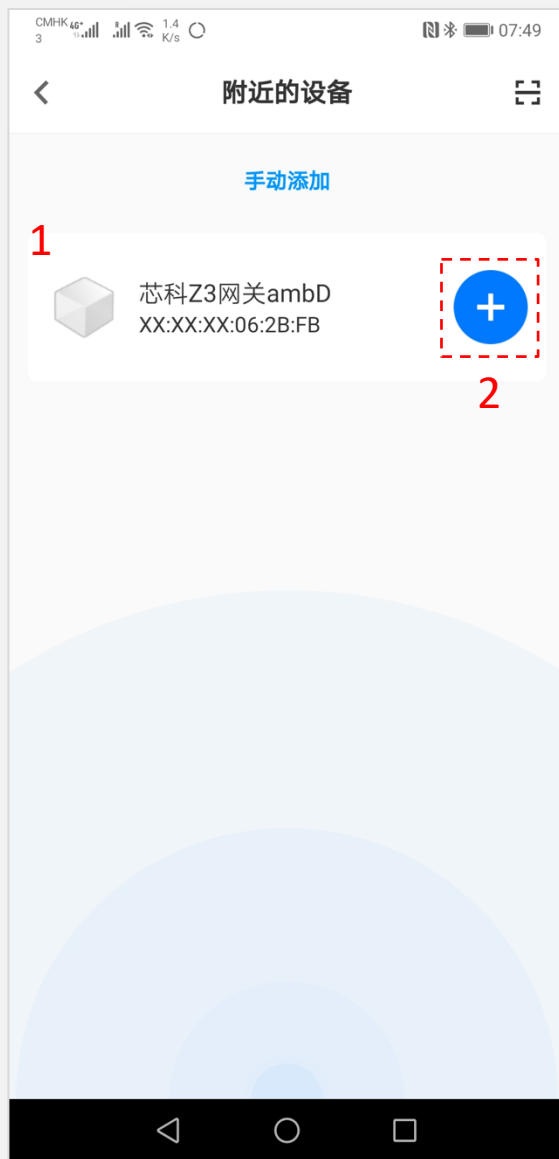


The image shows a mobile application interface for '我的家' (My Home). At the top, there are tabs for '用户版' (User Version) and '开发版' (Developer Version). Below the tabs, there is a QR code for downloading the app. The main content area shows the '我的家' section with a '+' icon in the top right corner. Below this, there are options for '设备' (Devices), '房间' (Rooms), and '分组' (Groups). A large blue circle with a white '+' sign is highlighted, with the text '暂无设备, 请添加' (No devices, please add) below it. At the bottom, there are navigation icons for home, search, and profile.



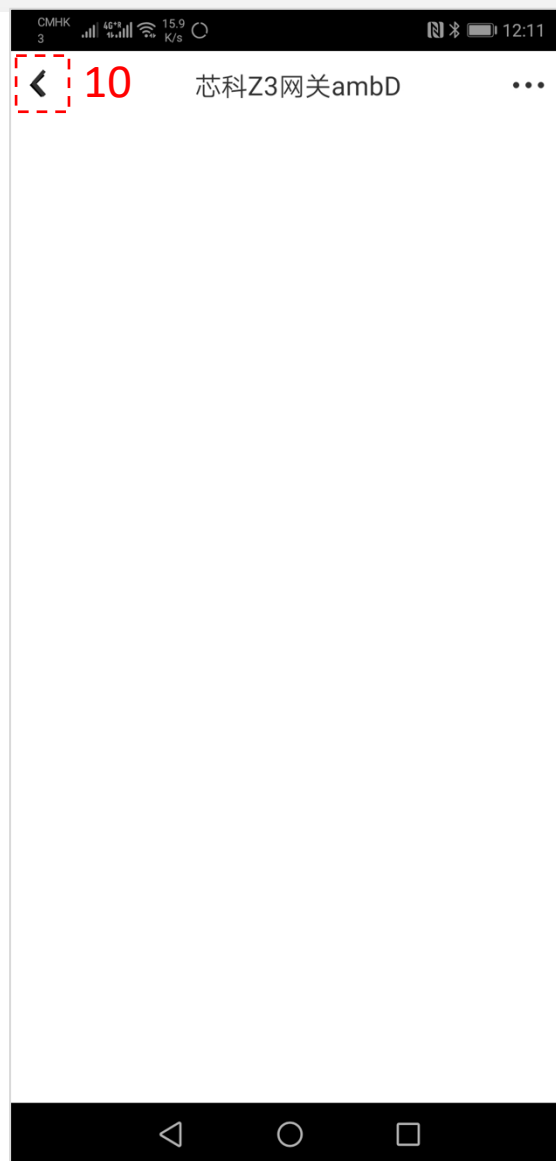
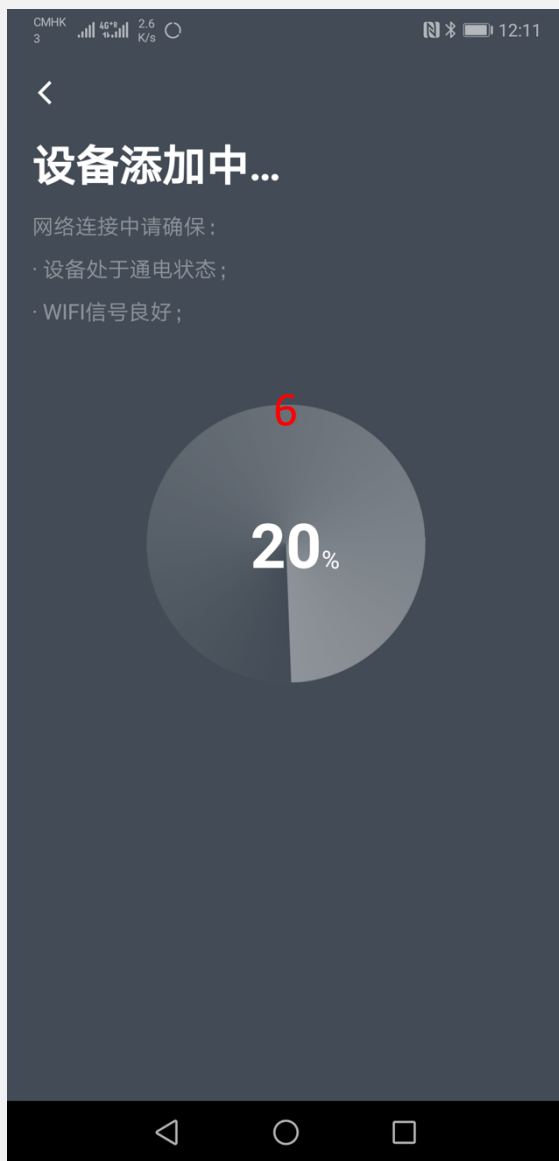
- 扫描左方二维码，安装阿里云智能手机应用程序。
  - 两个版本: 用户版和开发版 (两者并不相通)
- 使用你的帐户或下面分享帐户登录移动应用程序：
  - 用户名：[silabs\\_tester@sina.com](mailto:silabs_tester@sina.com)
  - 密码：[Silabs2014](#)
- LCGW网关Wi-Fi配网引导：
  - 按住COFIG按钮，插入USB电源线
  - Wi-Fi蓝色指示灯快速闪烁(100ms开/关)
- 云智能手机应用：
  - 将手机Wi-Fi连接到目标2.4GHz Wi-Fi路由器
  - 轻触右上角“+”图标或中间“(+)”图标，扫描附近的Wi-Fi直连设备，即LCGW套件。

# 阿里云：6. 云智能手机应用 - 添加LCGW设备 (2)



1. 在Android手机应用上，LCGW设备将被自动发现...
2. 当“芯科Z3网关ambD”出现，轻触右侧蓝色“(+)”图标
3. 输入Wi-Fi SSID和密码
4. 轻触“下一步”
5. 确认您看到LCGW蓝色指示灯快速闪烁，然后轻触“下一步”按钮继续...

# 阿里云：6. 云智能手机应用 - 添加LCGW设备 (3)

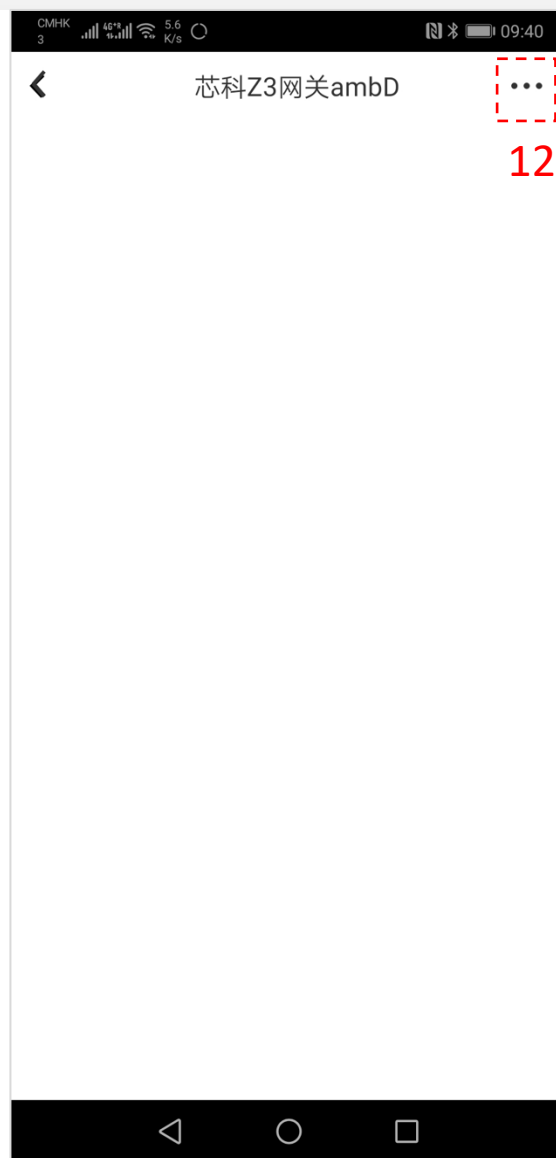


6. 你会看到“设备添加中”的进度...
7. 等待LCGW设备成功添加(等候少于1分钟)，可选择重命名设备
8. 可选择重命名房子并选择房间
9. 轻触“完成”按钮
10. 在设备控制面板中，轻触左上方“<”图标返回主页面

LED指示灯：

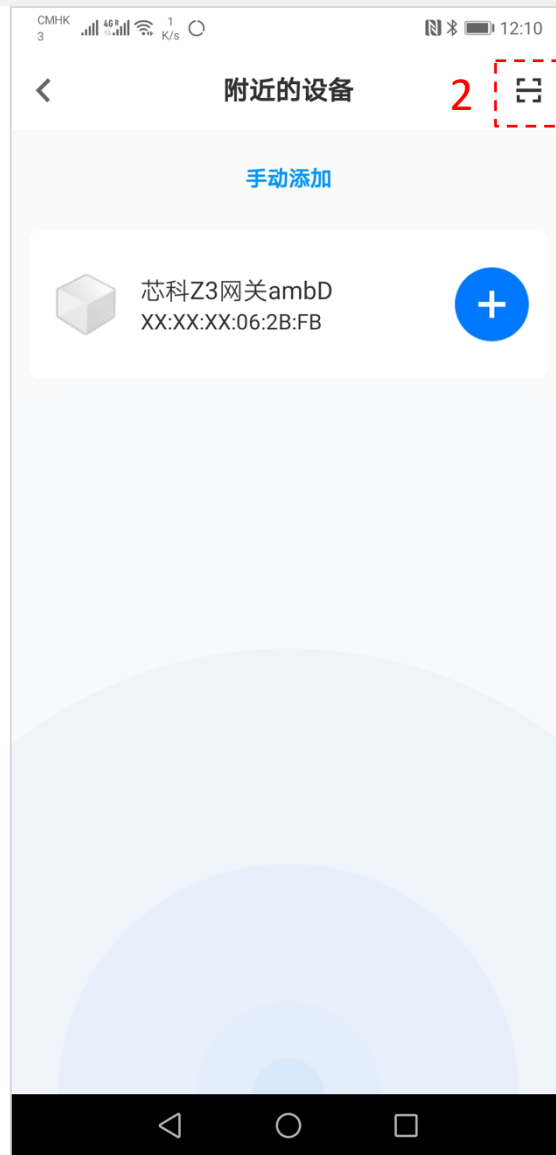
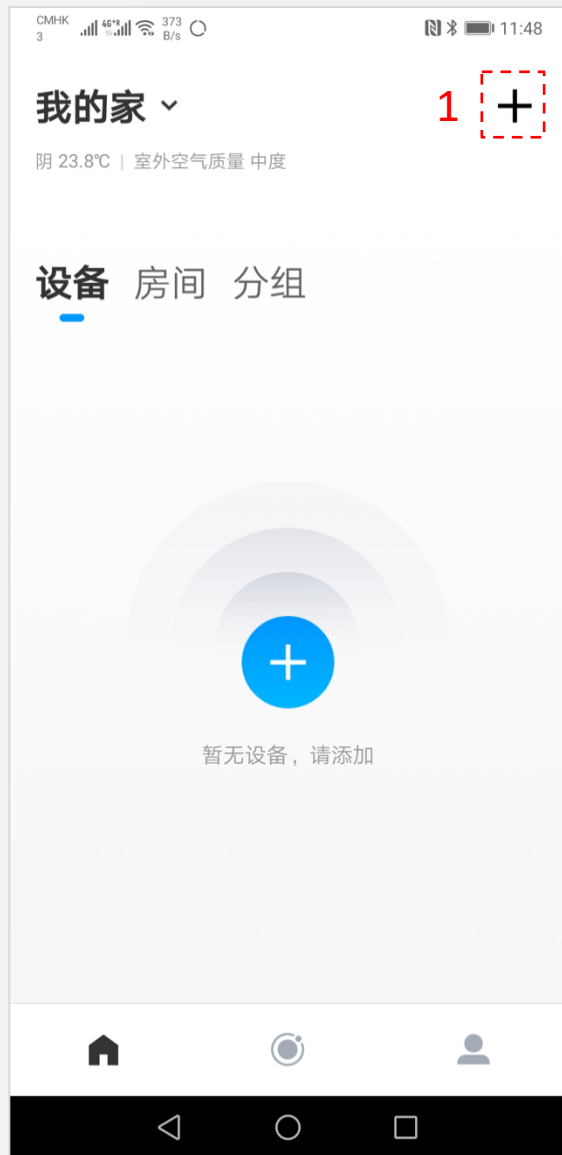
- 如果网关成功连接云，云绿色指示灯会亮起
- Wi-Fi蓝色和Zigbee红色指示灯恢复慢速闪烁

# 阿里云：6. 云智能手机应用 - 添加LCGW设备 (4)



11. “芯科Z3网关ambD”出现在设备列表中，轻触网关本身图标
12. 网关控制面板是空白的，因为LCGW并无提供相关控制。轻触右上角“...”图标查看网关的设置页面
13. 检查网关设备ID、IP地址、绑定时间等信息

# 阿里云：6. 云智能手机应用 - 添加LCGW设备替代方法一



## 添加LCGW设备的替代方法一：

1. 轻触右上“+”图标
2. 轻触“[-]”扫描图标
3. 扫描你自己或下面的网关二维码

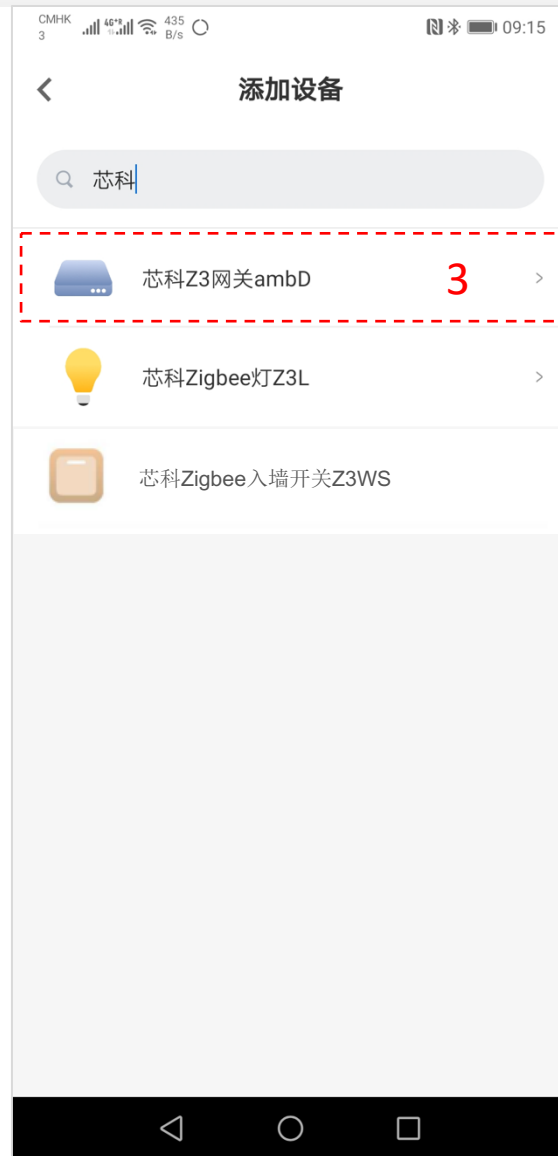
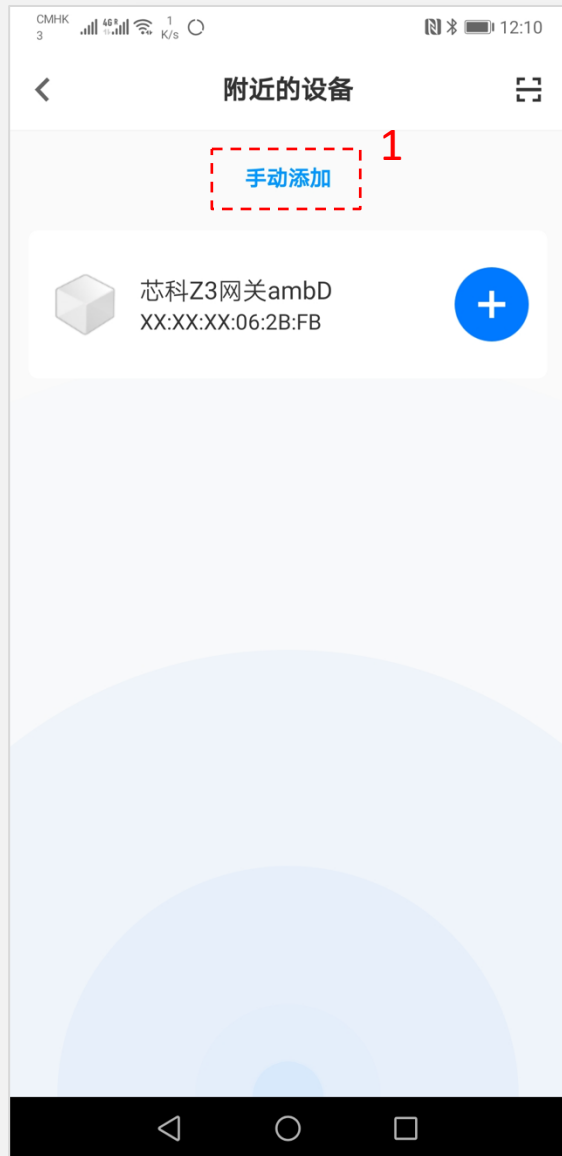
LCGW二维码



4. 按照前面的幻灯片(2)至(3)的步骤添加网关设备



# 阿里云：6. 云智能手机应用 - 添加LCGW设备替代方法二



1. 轻触上方中间的“手动添加”图标
2. 在“添加设备”面板搜索框中输入“芯科”(或你自己已发布的设备名称)以筛选设备
3. 从结果列表中轻触“芯科Z3网关ambD”(或你的设备名称)以添加设备

按照前面的幻灯片(2)至(3)的步骤添加网关设备

# 阿里云：7. 云智能手机应用 - 添加Zigbee子设备

## ■ LCGWv2配置按钮使用提示：

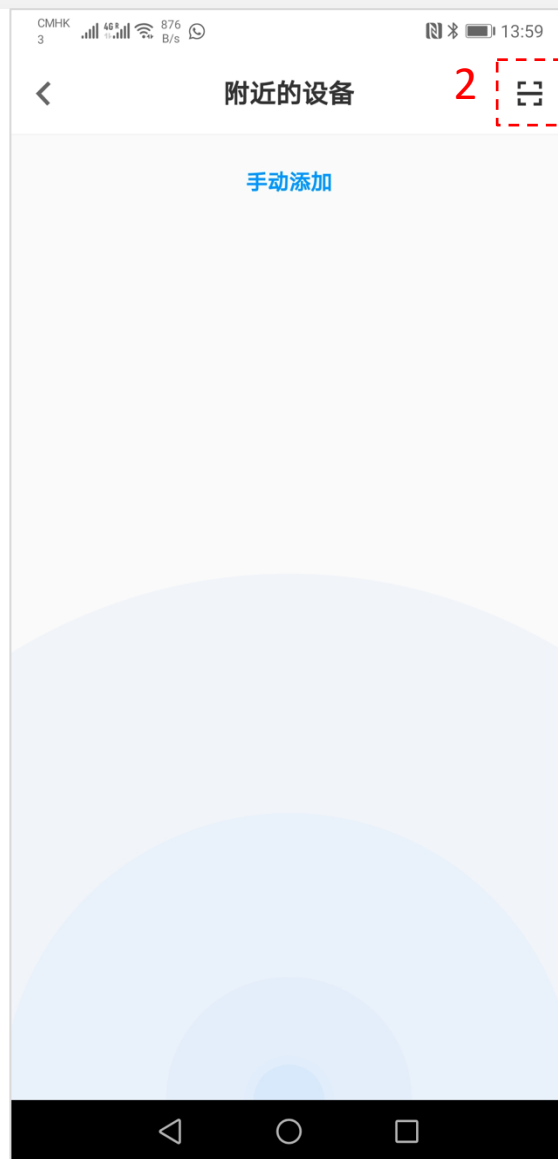
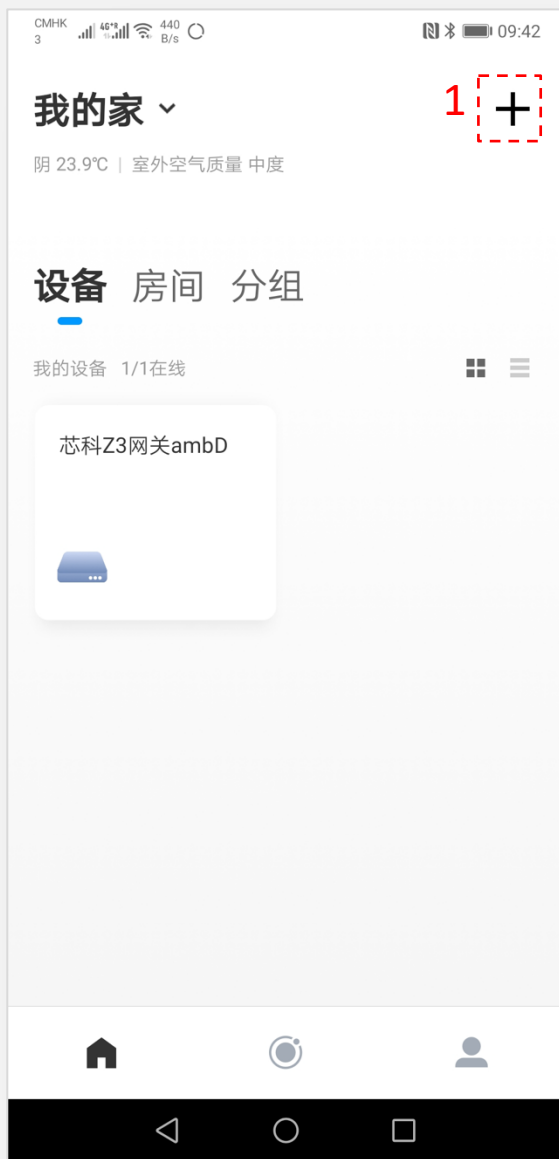
- 按一下CONFIG按钮(小于500毫秒)，网关允许Zigbee设备在180s内使用默认全局链接密钥加网，Zigbee(红色)指示灯快速闪烁(100毫秒亮100毫秒灭)
- 再按一下CONFIG按钮关闭设备加网，红色指示灯恢复缓慢闪烁(100毫秒亮1.9秒灭)
- 按住CONFIG按钮3秒钟以上，直到蓝色和绿色指示灯都亮起，然后松开按钮。LCGW将使所有连接的设备离网，并创建一个新的Zigbee网络。在复位过程中，蓝色和绿色指示灯将同时闪烁(100毫秒亮100毫秒灭)
- 按住CONFIG按钮8秒以上，直到三个指示灯都亮起，然后松开按钮，将LCGW重置为出厂默认值，即注销云并离开网络。在复位过程中，三个指示器将同时闪烁(100毫秒亮100毫秒灭)。复位过程完成后，三个指示灯保持亮起，LCGW不再工作。用户需要按住CONFIG按钮并重新上电，LCGW启动Wi-Fi配网。

## ■ 添加Zigbee子设备的三种不同方法：

1. 手机扫描设备的产品二维码(由生活物联网平台生成)
2. 在手机应用中从完整产品列表中手动选择设备产品(前提是产品已发布并名称通过审核)
3. LCGW CONFIG按钮一键配置，网关允许Zigbee设备加网并连接阿里云

注：方法3一键配置方法要求产品的第一个设备先已通过二维码方法成功添加后，才能正常使用。

# 阿里云：7. 云智能手机应用 - 添加Zigbee灯设备方法一 (1)



## 添加Zigbee灯方法一：扫码

1. 轻触右上“+”图标
2. 轻触“[-]”扫描图标
3. 扫描你自己或下面的Z3Light二维码

Z3Light 二维码



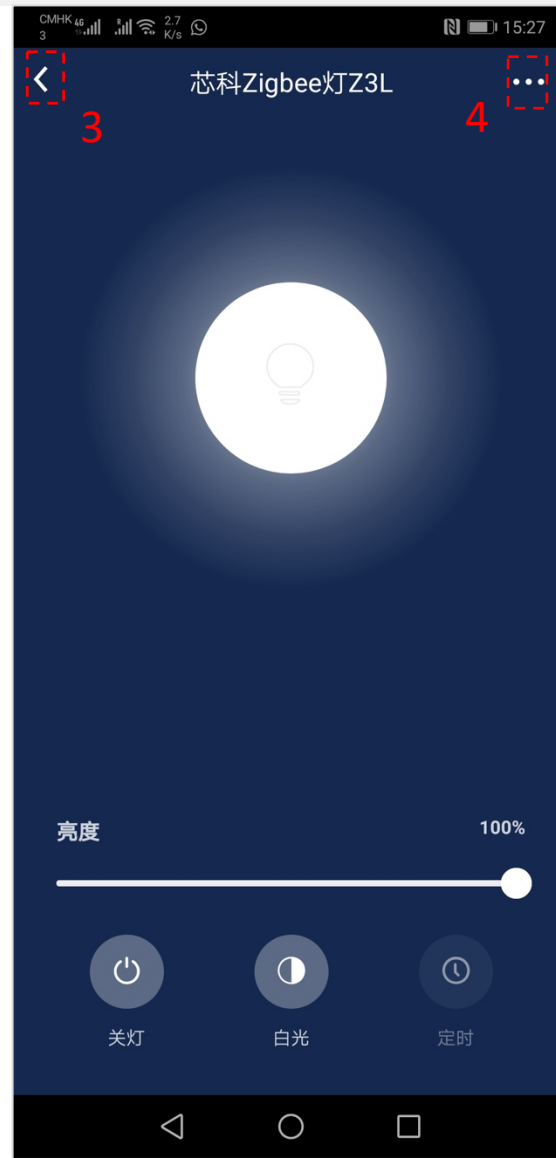
# 阿里云：7. 云智能手机应用 - 添加Zigbee灯设备方法一 (2)

## ■ 运行Z3LightSoC固件的WSTK加网：

1. 将已烧Z3LightSoC固件的BRD4161/4162A无线子板(MG12)的WSTK连接到PC的USB端口
2. 在Simplicity Studio的Debug Adapters窗口中，右键单击Z3Light的调试适配器并选择“launch Console”
3. 进入“串行1”窗口，在终端中按“回车”键，直到看到“Z3LightSoC>”提示出现。
4. 键入“**network leave**”，然后按“回车”键强制设备离网
5. 按下WSTK上的目标RESET按钮，Z3Light重启后搜索网络（可以在终端显示的日志信息中确认设备是否加网成功）



# 阿里云：7. 云智能手机应用 - 添加Zigbee灯设备方法一 (3)



1. 点击“下一步”按钮容许设备加网
2. 你会看到“设备添加中”的进度...而网关上的Zigbee红色指示灯正在快速闪烁(100毫秒亮100毫秒灭)，等待设备成功添加(等候少于1分钟)，Zigbee红色指示灯恢复慢闪(100毫秒亮1.9秒灭)
3. 轻触左上方“<”图标返回主页面，或
4. 轻触右上方“...”图标查看灯设备的设置页面

# 阿里云：7. 云智能手机应用 - 添加Zigbee灯设备方法一 (4)



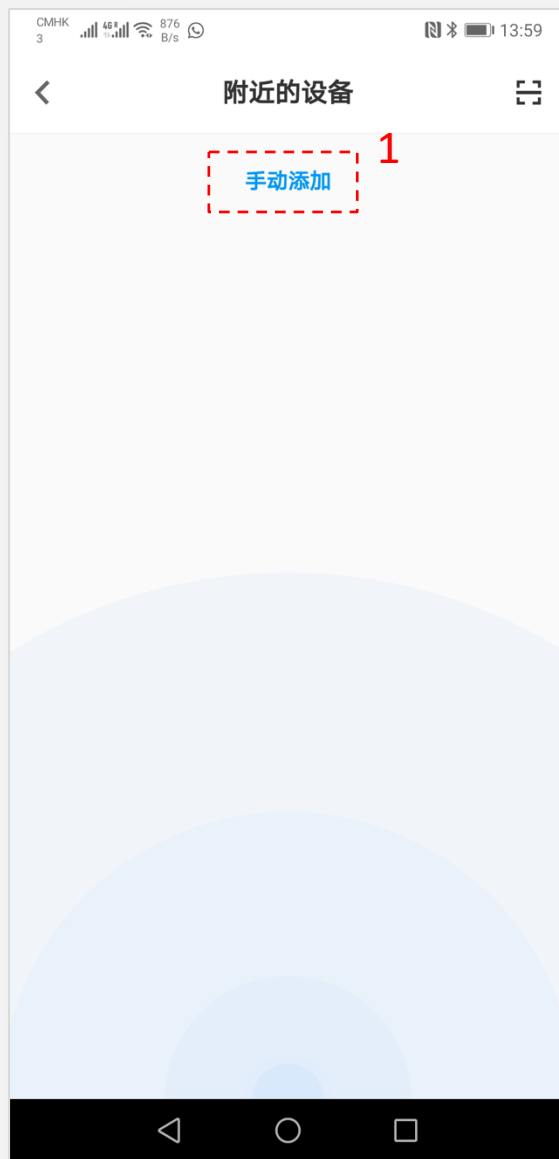
1. 回到主页面，“芯科 Zigbee灯 Z3L”出现在设备列表中(若没有，则需要刷新该页面)
2. 轻触“开/关”按钮，打开/关闭灯光
3. 轻触“芯科 Zigbee灯 Z3L”图标本身开启灯光控制面板
4. 轻触中间“灯泡”图标或底部的“开/关”按钮以打开/关闭灯光
5. 拖动亮度条可将亮度从0%调整到100% (WSTK上的LED不支持调节亮度)
6. 轻触右上角“...”图标查看灯设备的设置页面

# 阿里云：7. 云智能手机应用 - 添加Zigbee灯设备方法一 (5)



7. 在灯设备设置页面中，检查设备ID、IP地址、绑定时间等信息

# 阿里云：7. 云智能手机应用 - 添加Zigbee灯设备方法二



## 添加Zigbee灯方法二：手动添加

1. 轻触上方中间的“手动添加”图标
2. 在“添加设备”面板搜索框中输入“芯科”(或你已发布的设备名称)以筛选设备
3. 从结果列表中轻触“芯科Zigbee灯Z3L”(或你的设备名称)以添加设备
4. 按照方法一(2)至(4)的步骤添加灯设备和对灯进行控制

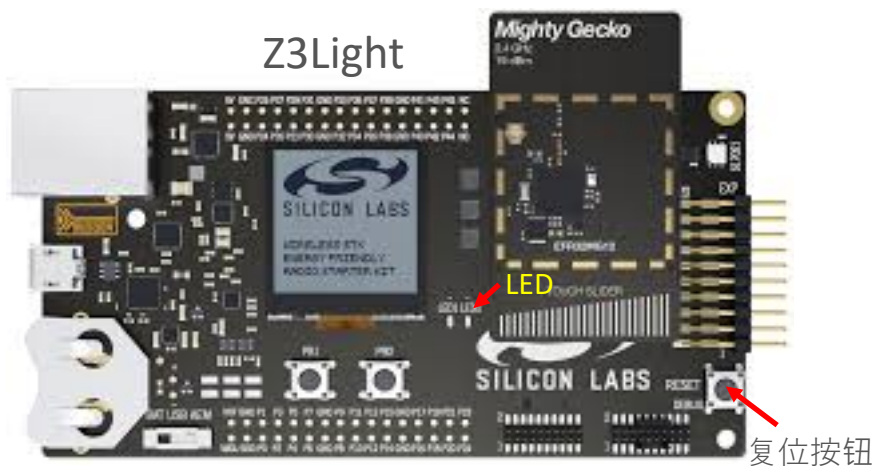


# 阿里云：7. 云智能手机应用 - 添加Zigbee灯设备方法三

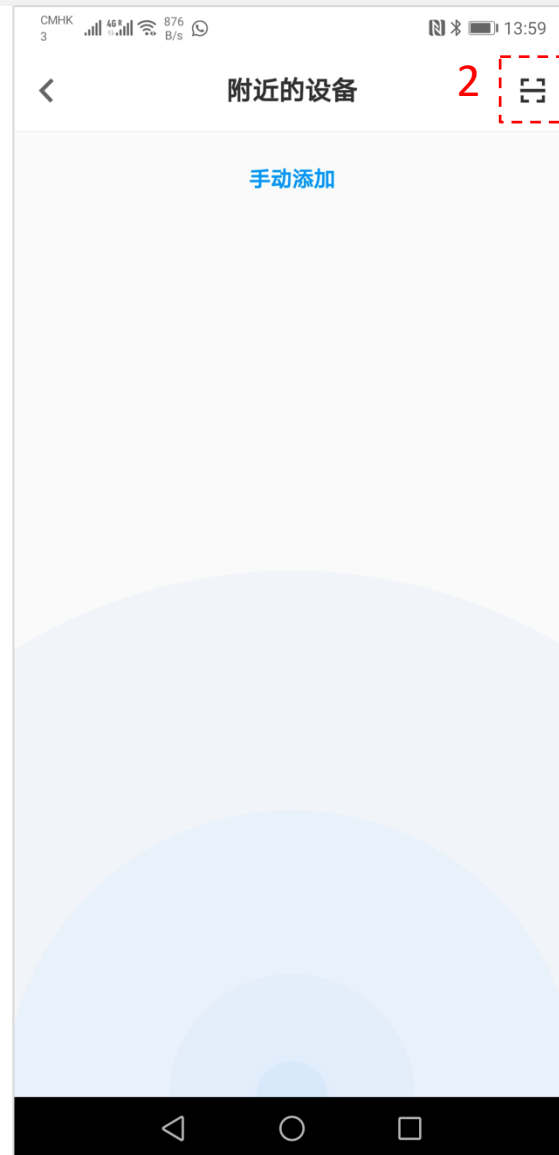
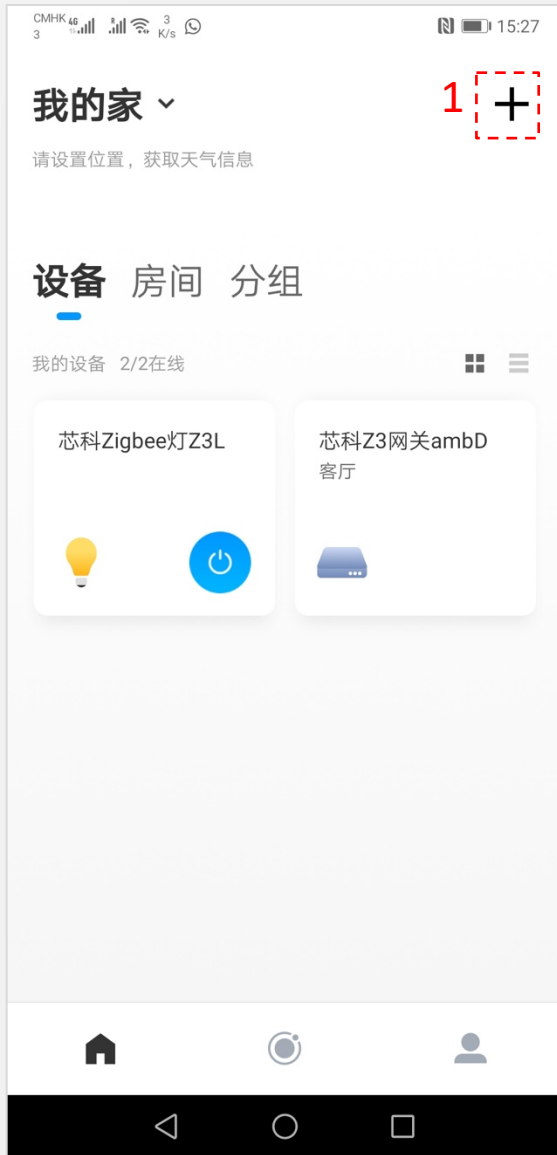


## 添加Zigbee灯方法三：使用CONFIG按键

- 按照方法一(2)强制设备离网并开始搜索Zigbee网络
- 让网关打开网络允许设备加网
  - 用笔短按一下隐藏的CONFIG按钮（小于500毫秒）
  - Zigbee红色指示灯快速闪烁(100毫秒亮100毫秒灭)
  - 网关允许设备加网180秒，使用默认全局链接密钥
- 一旦设备加网成功，或者加网超时，Zigbee红色指示灯将恢复慢速闪烁(100毫秒亮1.9秒灭)
- 如果设备加网成功，按照方法一(4)对设备进行控制



# 阿里云：8. 云智能手机应用 - 添加Zigbee开关设备方法一 (1)



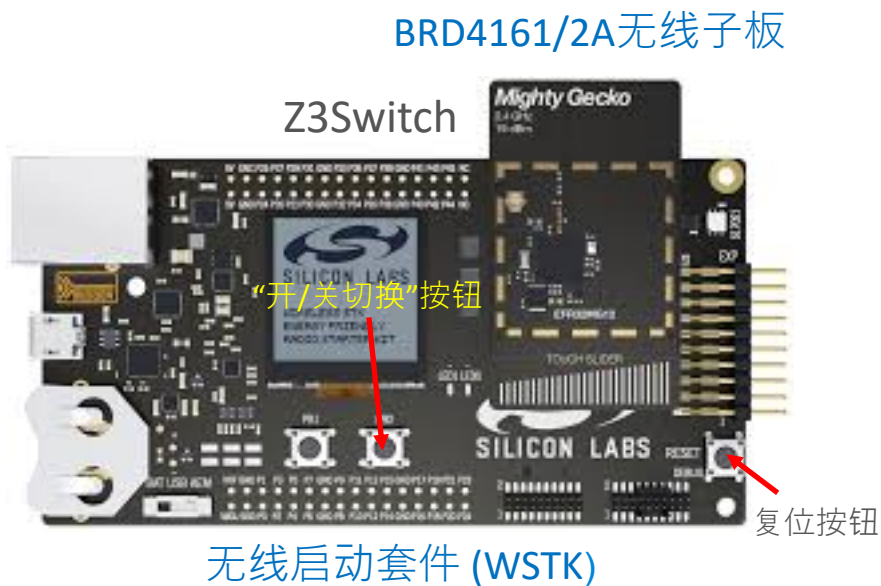
## 添加Zigbee开关方法一： 扫码

1. 轻触右上“+”图标
2. 轻触“[-]”扫描图标
3. 扫描你自己或下面的 Z3Switch 二维码

Z3Switch 二维码



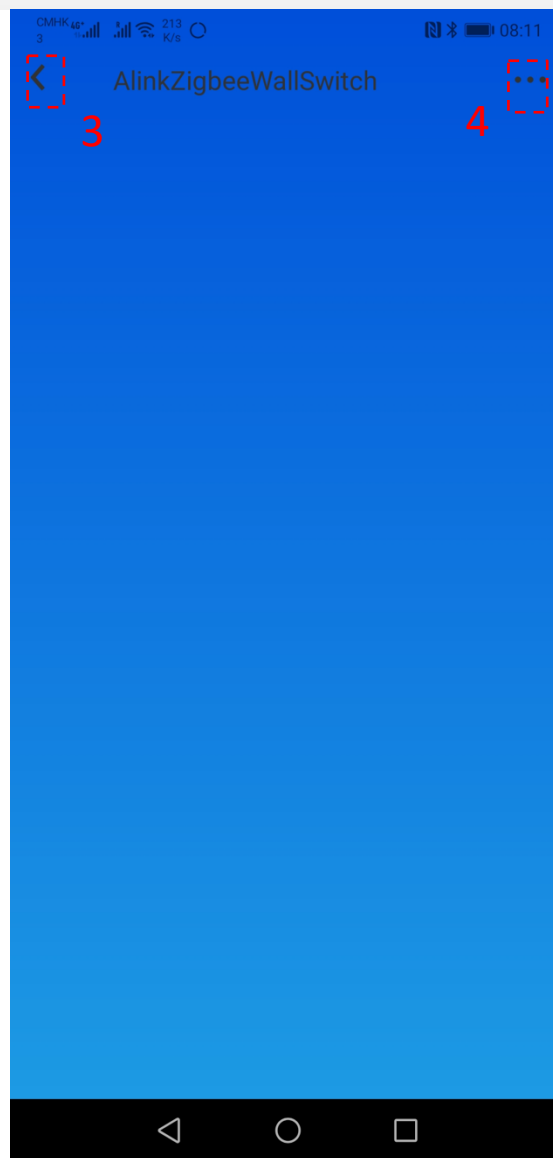
# 阿里云：8. 云智能手机应用 - 添加Zigbee开关设备方法一 (2)



## ■ 运行Z3SwitchSoC固件的WSTK加网：

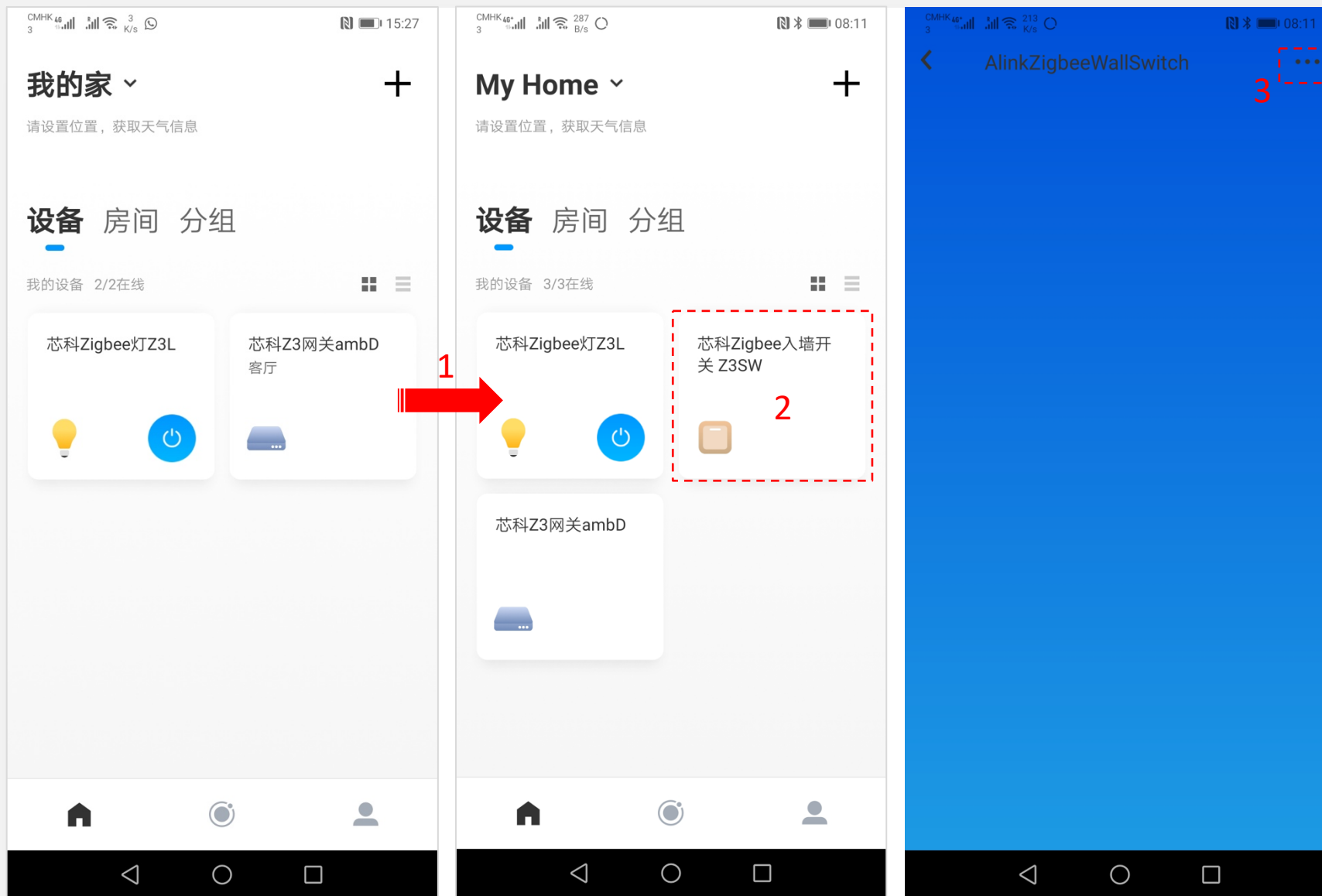
1. 将已烧Z3SwitchSoC固件的BRD4161/4162A无线子板(MG12)的WSTK连接到PC的USB端口
2. 在Simplicity Studio的Debug Adapters窗口中，右键单击Z3Switch的调试适配器并选择“launch Console”
3. 进入“串行1”窗口，在终端中按“回车”键，直到看到“Z3SwitchSoC>”提示出现。
4. 键入“**network leave**”，然后按“回车”键强制设备离网
5. 键入“**plugin network-steering start 60**”，然后按“回车”键命令设备搜网并加入网络  
(可以在终端显示的日志信息中确认设备是否加网成功)

# 阿里云：8. 云智能手机应用 - 添加Zigbee开关设备方法一 (3)



1. 点击“下一步”按钮容许设备加网
2. 你会看到“设备添加中”的进度...而网关上的Zigbee红色指示灯正在快速闪烁(100毫秒亮100毫秒灭)，等待设备成功添加(等候少于1分钟)，Zigbee红色指示灯恢复慢闪(100毫秒亮1.9秒灭)
3. 轻触左上方“<”图标返回主页面，或
4. 轻触右上方“...”图标查看开关设备的设置页面

# 阿里云：8. 云智能手机应用 - 添加Zigbee开关设备方法一 (4)



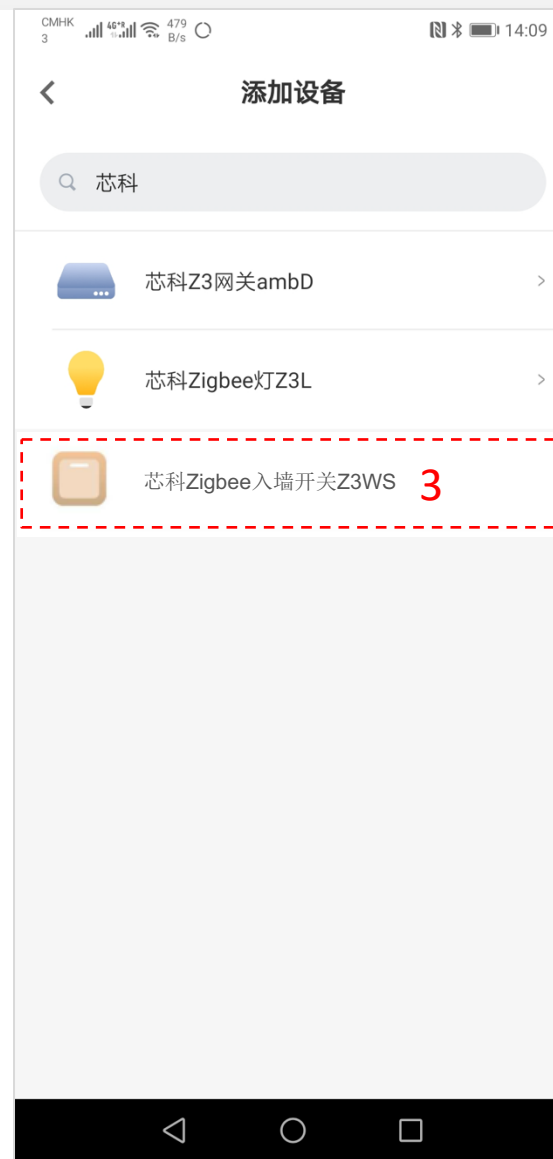
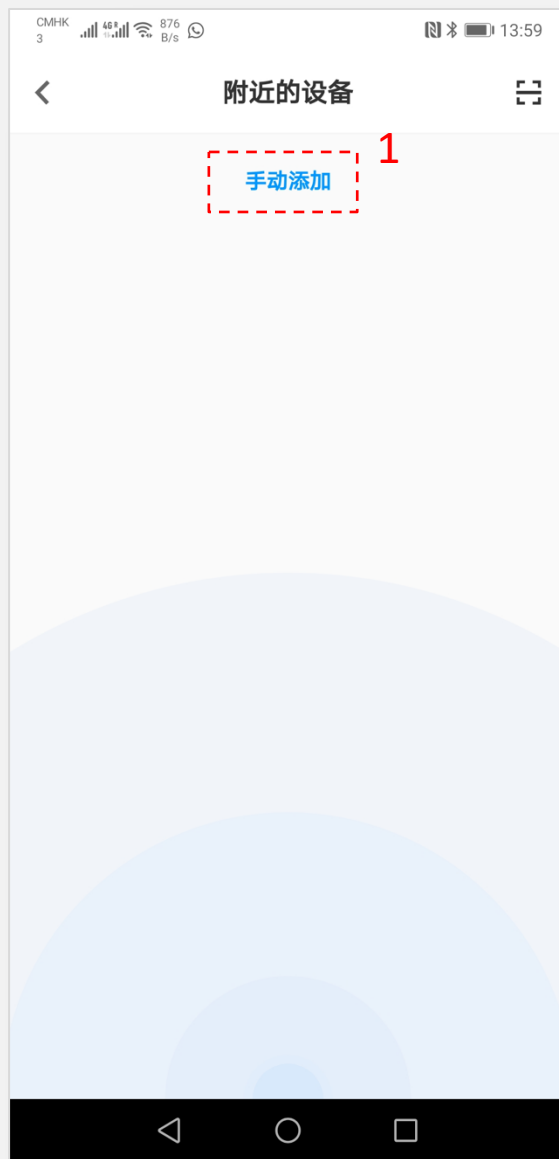
1. 回到主页面，“芯科 Zigbee入墙开关Z3SW”出现在设备列表中(若没有，则需要刷新该页面)
2. 轻触开关设备本身图标
3. 开关控制面板是空白的，因为开关并无提供相关控制。轻触右上角“...”图标查看开关的设置页面

# 阿里云：8. 云智能手机应用 - 添加Zigbee开关设备方法一 (5)



4. 在开关设备设置页面中，检查设备ID、IP地址、绑定时间等信息

# 阿里云：8. 云智能手机应用 - 添加Zigbee开关设备方法二



## 添加Zigbee开关方法二： 手动添加

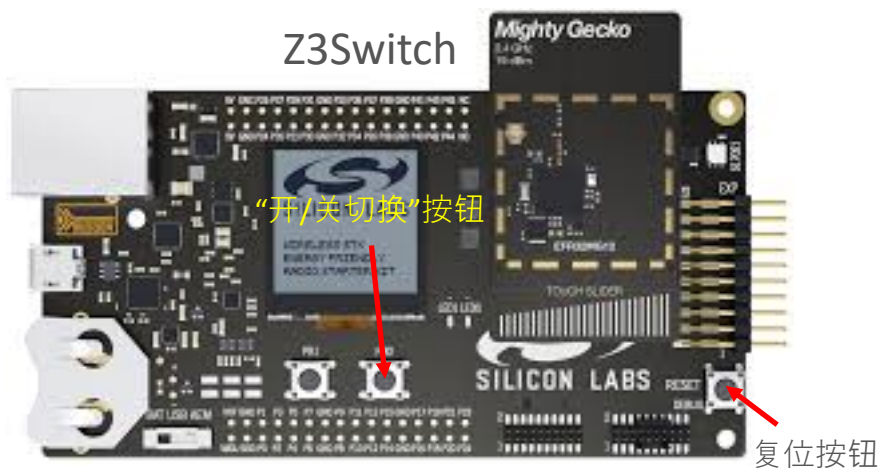
1. 轻触上方中间的“手动添加”图标
2. 在“添加设备”面板搜索框中输入“芯科”(或你已发布的设备名称)以筛选设备
3. 从结果列表中轻触“芯科Zigbee入墙开关Z3SW”(或你的设备名称)以添加设备

按照方法一(2)至(4)添加开关设备和进行设备控制

# 阿里云：8. 云智能手机应用 - 添加Zigbee开关设备方法三



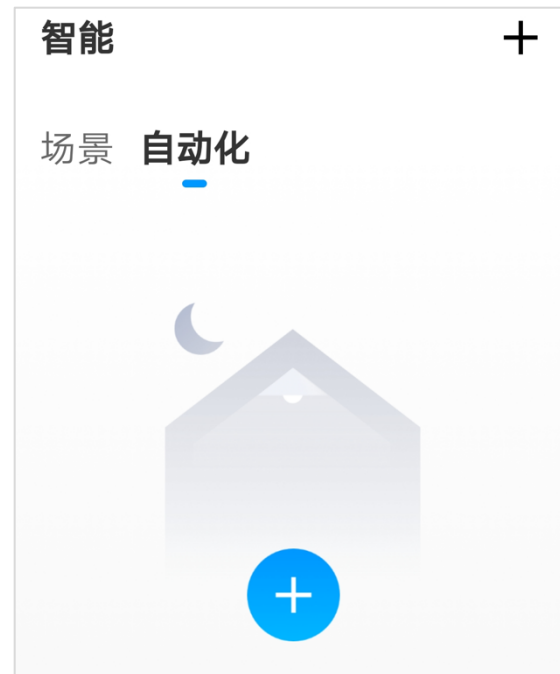
- 添加Zigbee开关方法三：使用CONFIG按键
- 按照方法一(2)强制设备离网并开始搜索网络
- 让网关打开网络允许设备加网
  - 用笔短按一下隐藏的CONFIG按钮（小于500毫秒）
  - Zigbee红色指示灯快速闪烁(100毫秒亮100毫秒灭)
  - 网关允许设备加网180秒，使用默认全局链接密钥
- 一旦设备加网成功，或者加网超时，Zigbee红色指示灯将恢复慢速闪烁(100毫秒亮1.9秒灭)
- 如果设备加网成功，按照方法一(4)对设备进行控制



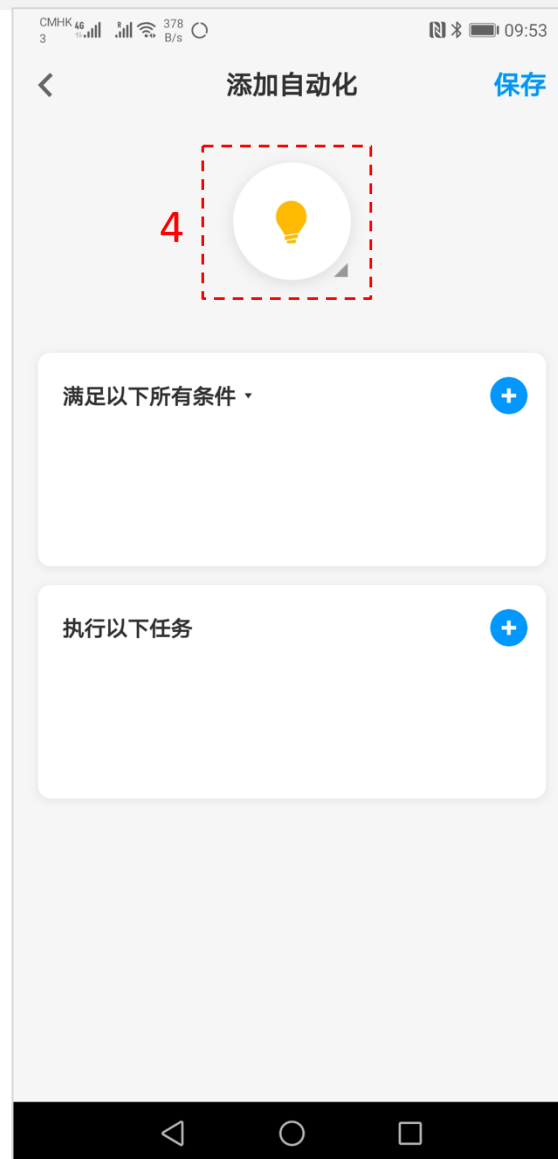
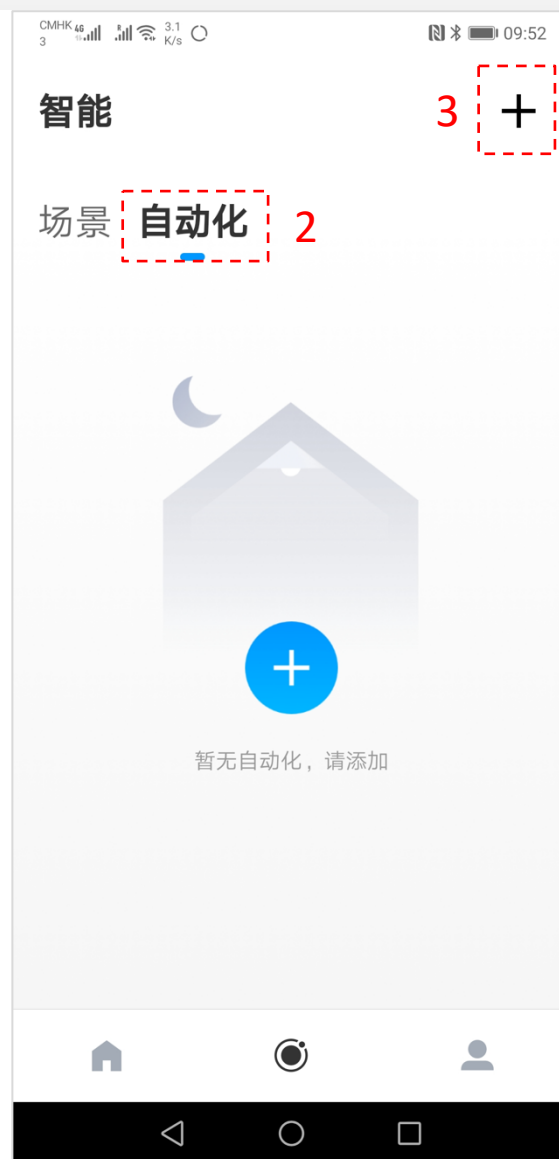
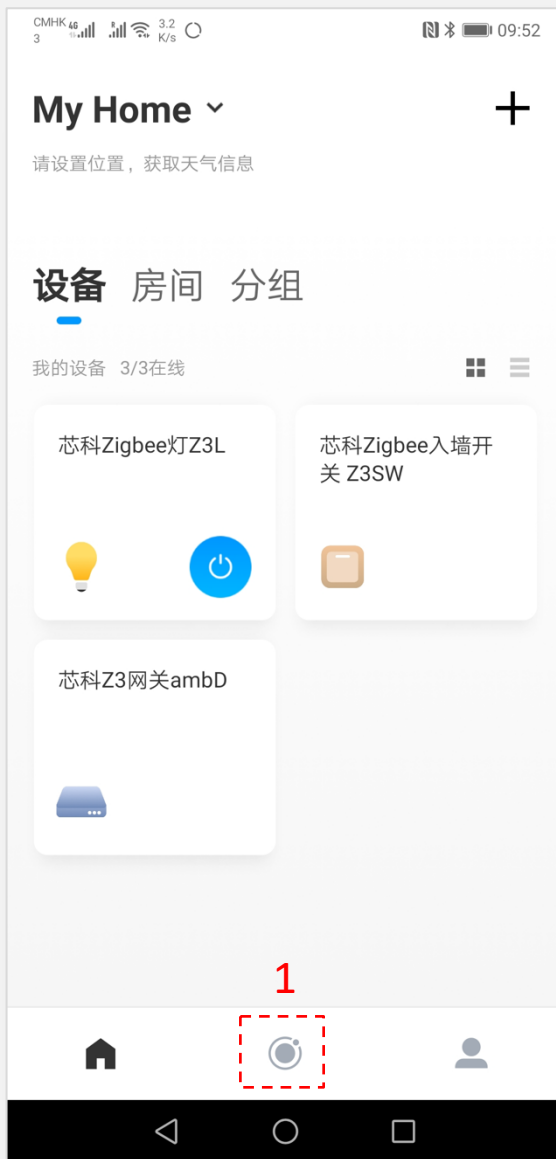


# 阿里云：9. 云智能手机应用 - 设备自动化

- 在云智能手机应用程序上，用户可以为网关的子设备设置基于云的规则，实现自动化。
- 例如，设置Z3Switch控制Z3Light的规则：按Z3Switch WSTK上的切换按钮来控制Z3Light WSTK上的LED指示灯开/关。

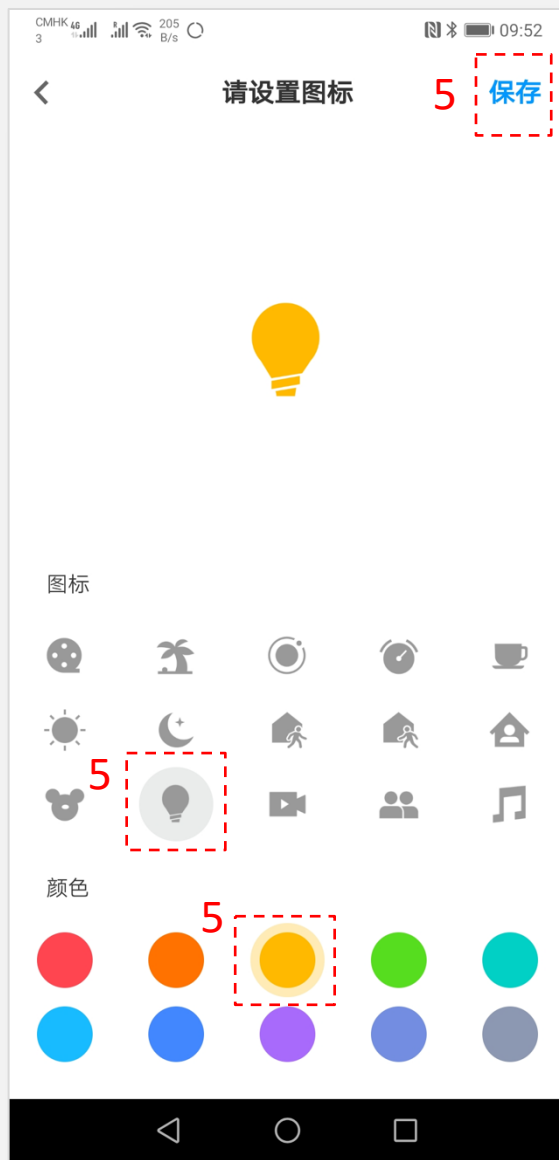


# 阿里云：9. 云智能手机应用 - 自动化设置 (1)



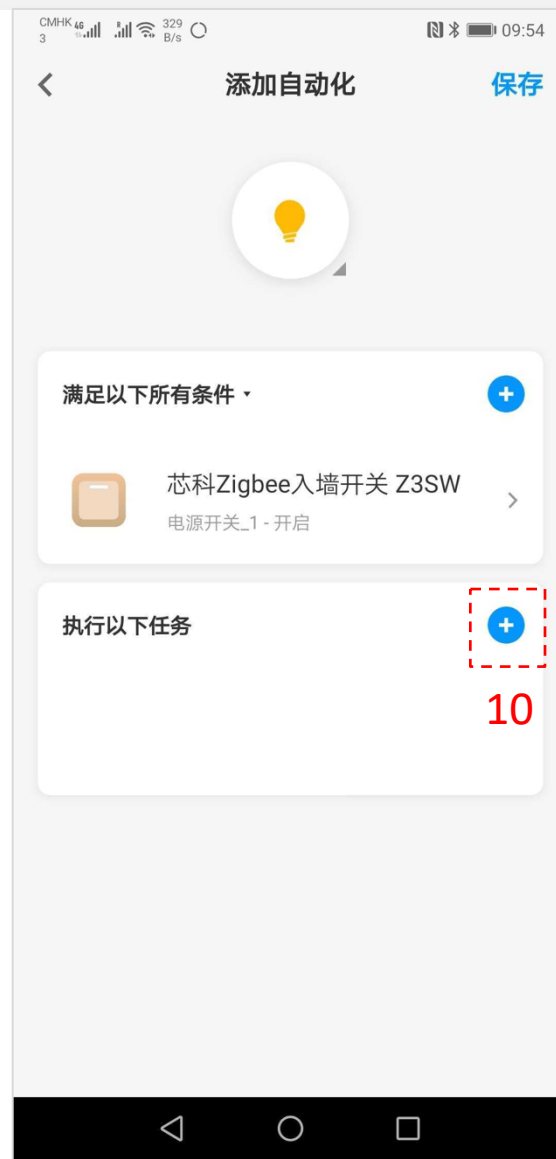
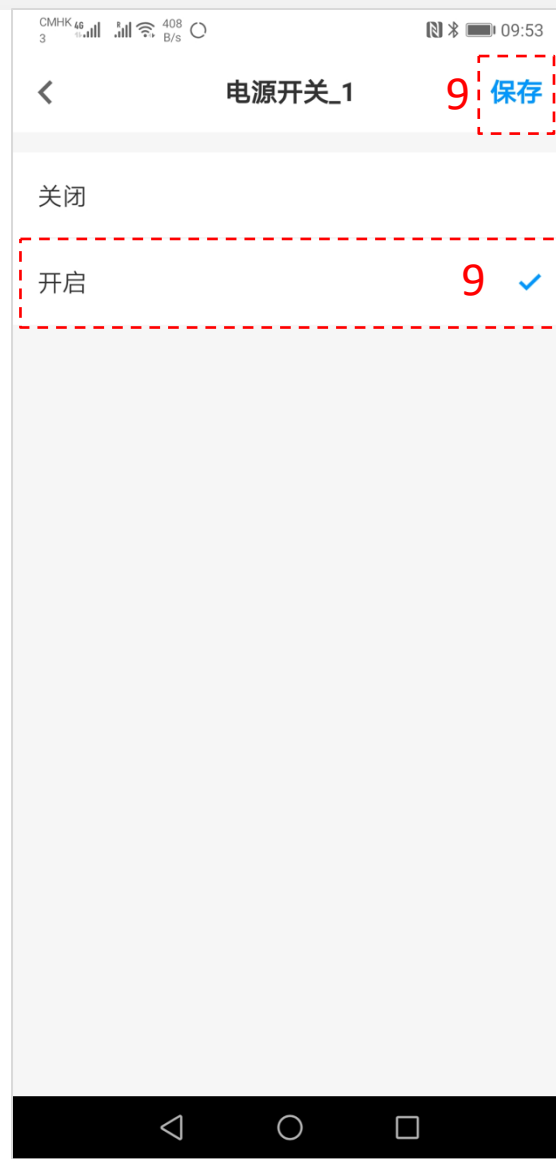
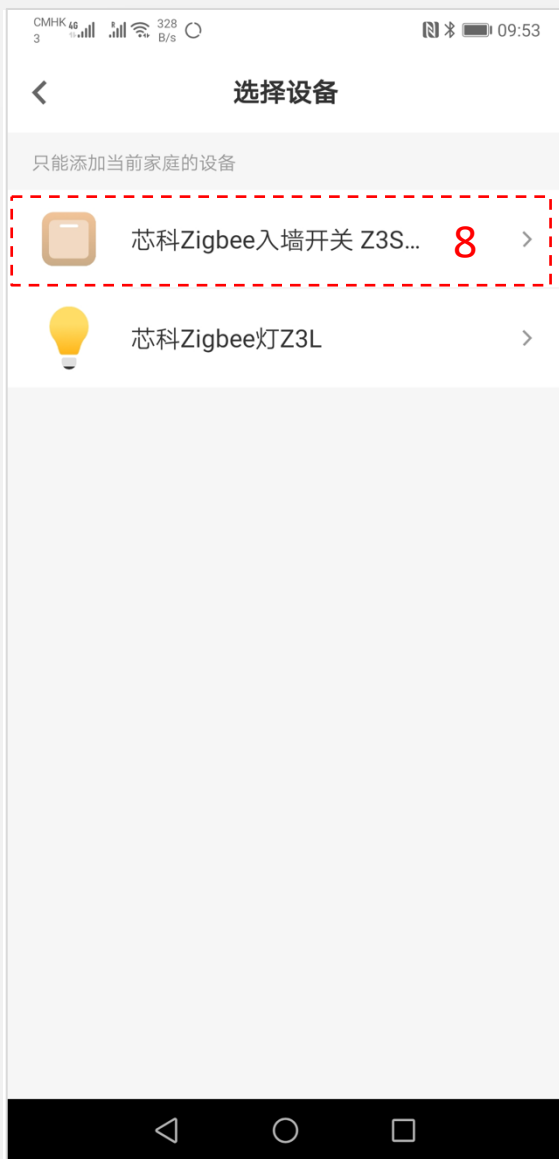
1. 点击底部中间的圆圈图标进入“智能”页面
2. 轻触“自动化”标签
3. 轻触“添加自动化”页面右上角的“+”
4. 点击中间图标可自定义图标

# 阿里云：9. 云智能手机应用 - 自动化设置 (2)



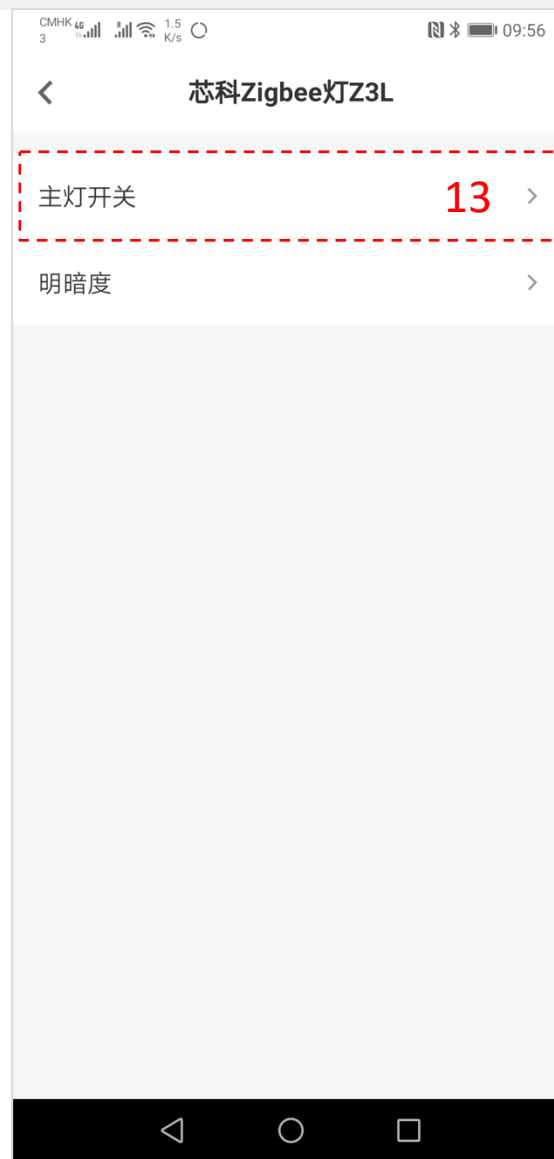
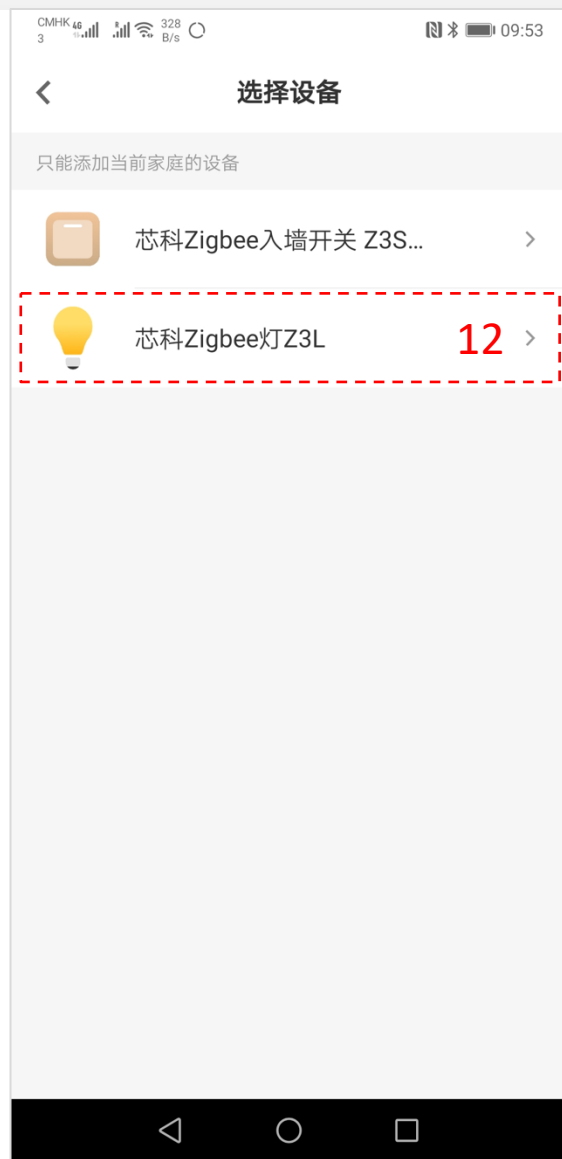
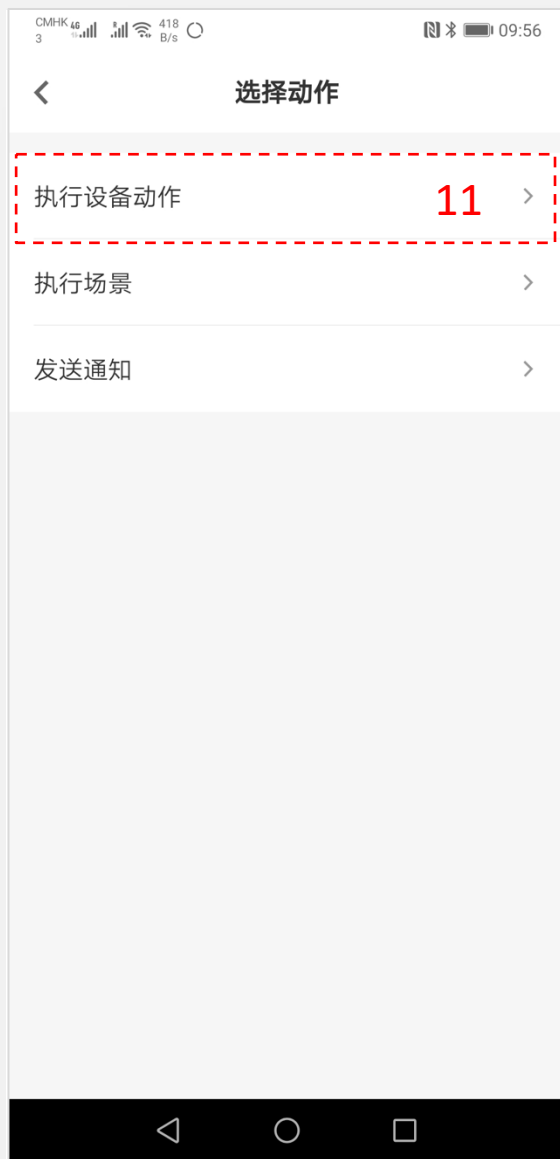
5. 选择图标和颜色，然后点击“保存”
6. 点击“+”打开“添加条件”页面
7. 选择“设备状态”并转到“选择设备”页面

# 阿里云：9. 云智能手机应用 - 自动化设置 (3)



8. 选择所需的开关设备“芯科Zigbee入墙开关Z3SW”
9. 选择“开启”并轻触“保存”
10. 点击“+”打开“选择动作”页面

# 阿里云：9. 云智能手机应用 - 自动化设置 (4)



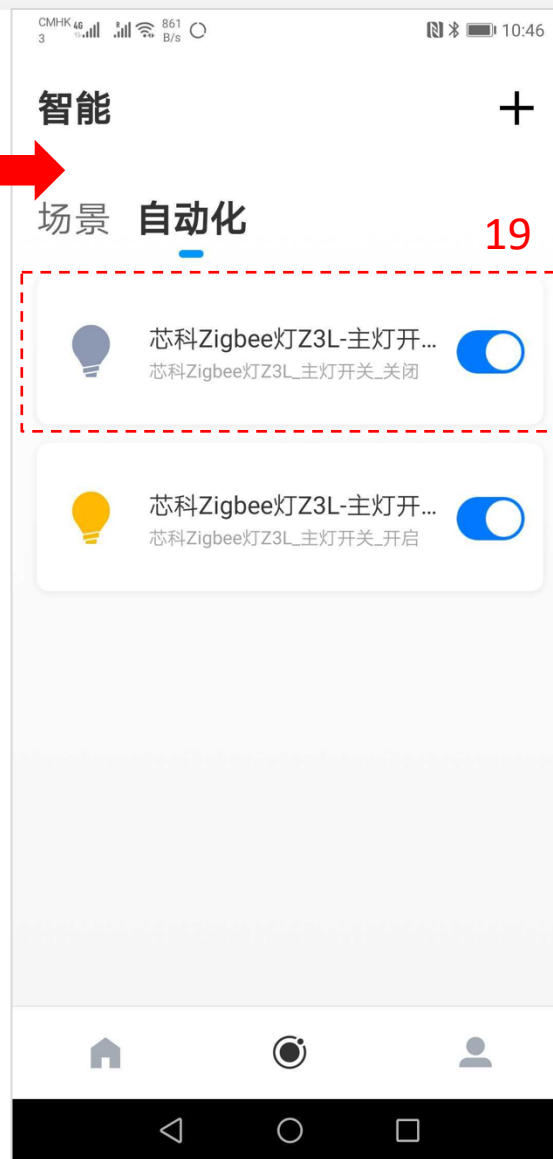
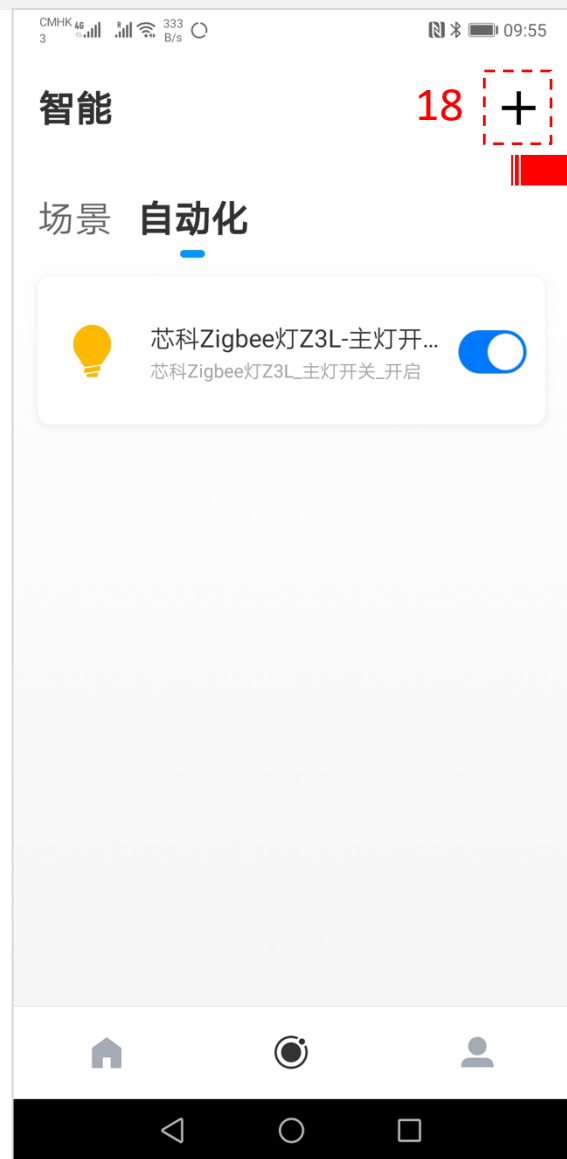
11. 选择“执行设备动作”
12. 选择所需的照明设备“芯科Zigbee灯Z3L”
13. 选择“主灯开关”

# 阿里云：9. 云智能手机应用 - 自动化设置 (5)



14. 选择“开启”并轻触“保存”
15. 点击右上角的“保存”
16. 输入自动化规则的名称并点击“确定”

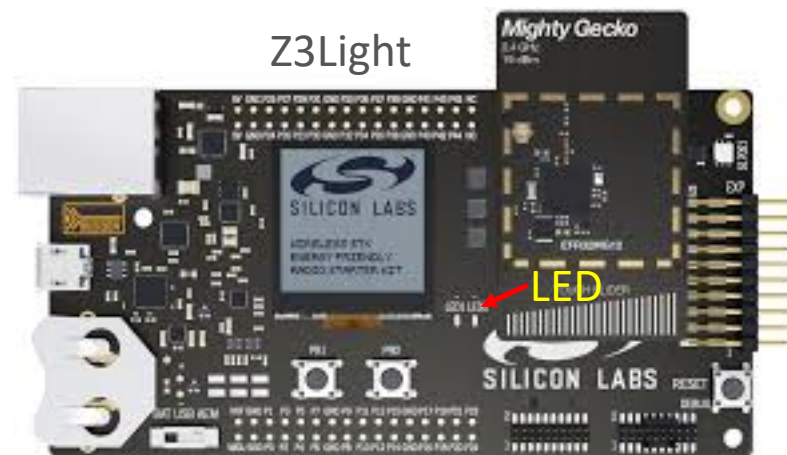
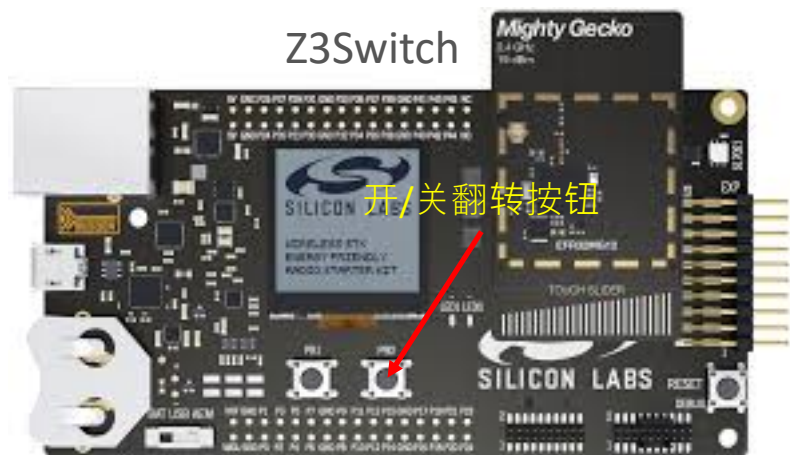
# 阿里云：9. 云智能手机应用 - 自动化设置 (6)



17. 新创建的“开启 -> 开启”自动化规则出现在“自动化”页面上
18. 轻触“+”图标并按照前面的步骤以相同方式创建另一个“关闭 -> 关闭”规则
19. 新创建的“关闭 -> 关闭”自动化规则出现在“自动化”页面上

# 阿里云：9. 云智能手机应用 - 自动化操作

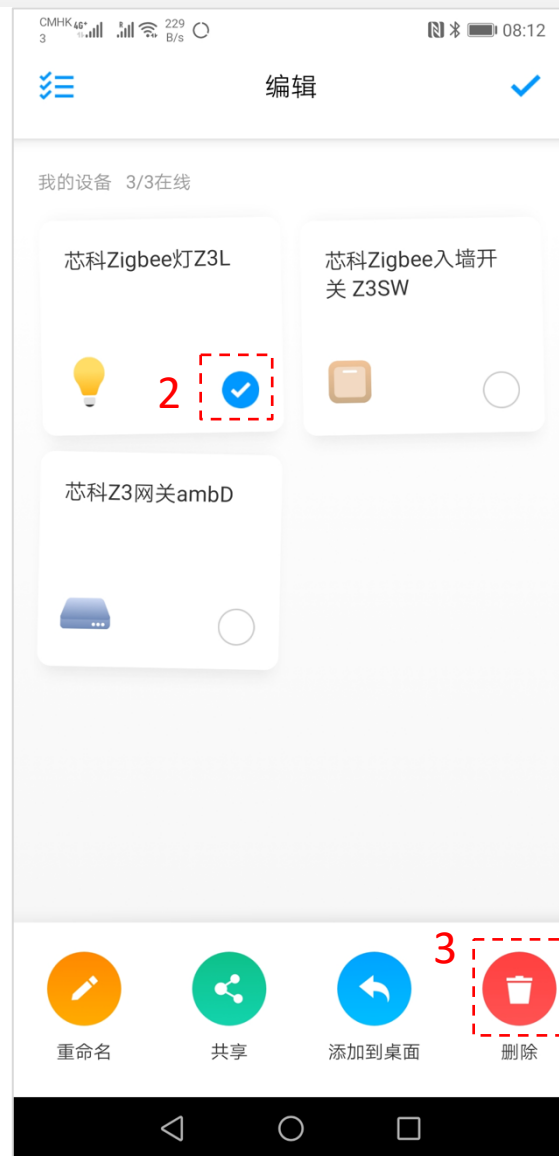
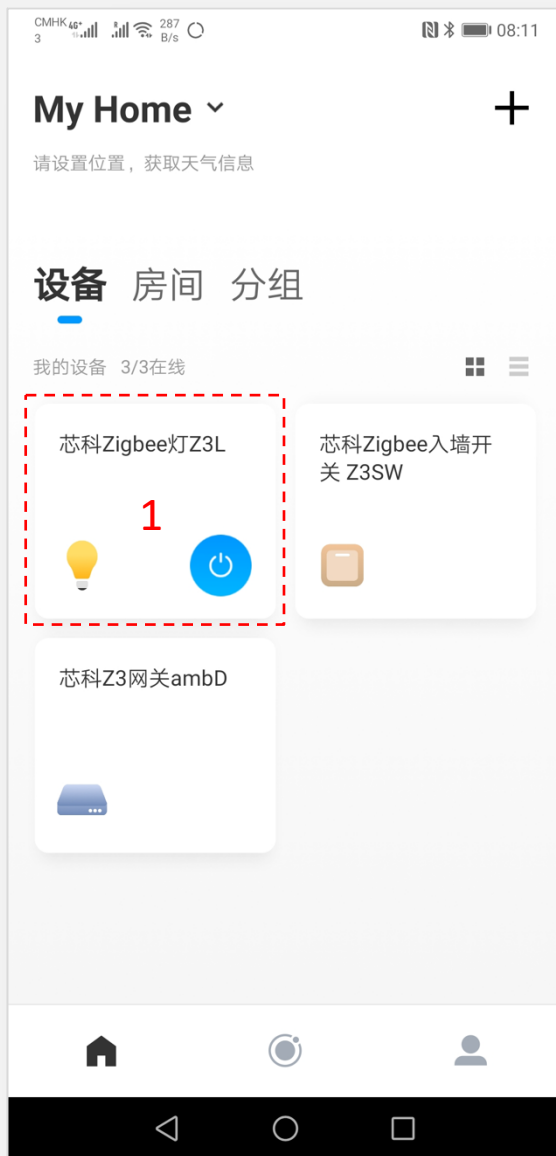
- 按下Z3Switch WSTK上的“开/关切换”按钮，控制所选WSTK上的LED指示灯开启/关闭。並可在手机应用程序上观察所选Z3Light设备的开/关状态变化。



- 注：
  - 当Z3Switch WSTK在按下“开/关翻转”按钮时发送切换属性时，Alink 灯物模型只接受设置其开或关属性。Alink灯物模型不支持“开/关翻转”属性。因此，阿里Zigbee网关示例代码需要将“开/关翻转”属性值转换为Alink灯物模型操作的特定开或关属性。
  - 然而，由于自动化规则是在云端设置的，LCGW并不知道Z3Switch试图控制哪个Z3Light。因此，当前的转换依赖于一个静态变量来记住，这可能不等于灯的实际状态。因此，第一次按下“开/关翻转”按钮可能观察不到LED指示灯有开/关变化。



# 阿里云：10. 云智能应用-删除设备



1. 轻触并按住主页面设备列表中的设备并切换到编辑模式
2. 点击设备旁边的单选按钮，准备下一步操作
3. 点击底部的“删除”图标按钮删除该设备，即从手机应用程序中用户解除设备的绑定
4. 或者，在个别设备的设置面板中，向下滚动并点击“删除设备”按钮并确认以绑

# 阿里云：11. 云控制台中查看网关设备状态

**ALinkGateway**

已发布 WiFi 网关

ProductKey: a1VtEtCqVsx

激活: 1    在线: 0

创建时间: 2019-10-09

alink_gw_0001	<span>● 在线</span>	2020-05-08 10:25:55	<a href="#">查看</a> <a href="#">调试</a> <a href="#">子设备(1)</a> <a href="#">激活凭证</a>
---------------	-------------------	---------------------	--

**alink\_gw\_0001** 在线

产品: ALinkGateway -    ProductKey: a1VtEtCqVsx [复制](#)    DeviceSecret: \*\*\*\*\* [复制](#)

[设备信息](#)   [Topic列表](#)   [运行状态](#)   [事件管理](#)   [服务调用](#)   [设备影子](#)   [文件管理](#)   [日志服务](#)   [在线调试](#)   [子设备管理](#)


**子设备管理 (1)** ?

[搜索](#) [刷新](#)

<input type="checkbox"/>	DeviceName	设备所属产品	节点类型	状态/启用状态	最后上线时间	操作
<input type="checkbox"/>	alilink_light_0001	ALinkZigbeeLight	设备	<span>● 在线</span> <input checked="" type="checkbox"/>	2020/05/08 10:25:55	<a href="#">查看</a> <a href="#">删除</a>

[批量删除](#)   [批量禁用](#)   [批量启用](#)

# 阿里云：12. 云控制台中调试Zigbee灯设备



**ALinkZigbeeLight**

已发布 ZigBee 灯

ProductKey: a1cSttZ7TXM [复制](#)

激活: 4 在线: 0

创建时间: 2019-10-26

ID	状态	最后上报时间	操作
alilink_light_0003	● 离线	2020-04-28 15:58:04	<a href="#">查看</a> <a href="#">调试</a> <a href="#">激活凭证</a>
alilink_light_0002	● 离线	2020-04-28 15:57:59	<a href="#">查看</a> <a href="#">调试</a> <a href="#">激活凭证</a>
alilink_light_0001	● 在线	2020-05-08 10:25:55	<a href="#">查看</a> <a href="#">调试</a> <a href="#">激活凭证</a>

### alilink\_light\_0001 在线

产品: ALinkZigbeeLight - ProductKey: a1cSttZ7TXM [复制](#) DeviceSecret: \*\*\*\*\* [复制](#)

设备信息 Topic列表 运行状态 事件管理 服务调用 设备影子 文件管理 日志服务 在线调试

调试设备: alilink\_light\_0001

调试真实设备 调试虚拟设备

属性调试 服务调用

调试功能: 主灯开关 (Light... 方法: 设置

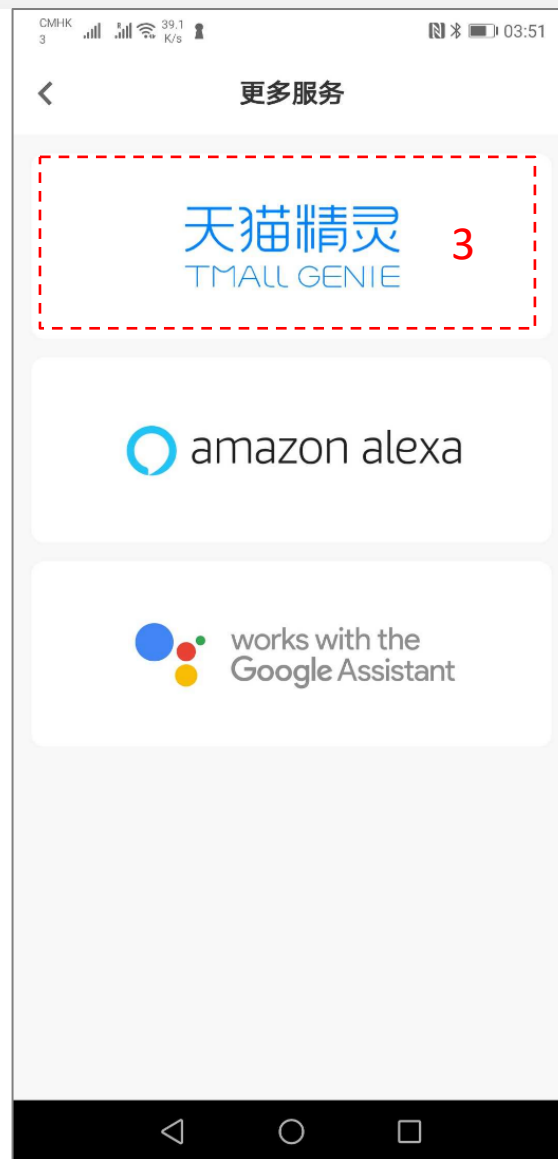
```
1 {
2   "LightSwitch": 0
3 }
```

#### 实时日志 ● 在线 (真实设备)

自动刷新  [刷新](#) [清屏](#)

类型 / 时间	内容
---------	----

# 天猫精灵控制LCGW Zigbee子设备 (1)



- 阿里云帮助文档  
[https://help.aliyun.com/document\\_detail/131285.html?](https://help.aliyun.com/document_detail/131285.html?)
  - 使用云智能手机应用配置要由天猫精灵智能扬声器控制LCGW的Zigbee设备
1. 轻触“我的”图标
  2. 轻触“更多服务”
  3. 轻触“天猫精灵”
  4. 轻触“绑定帐户”

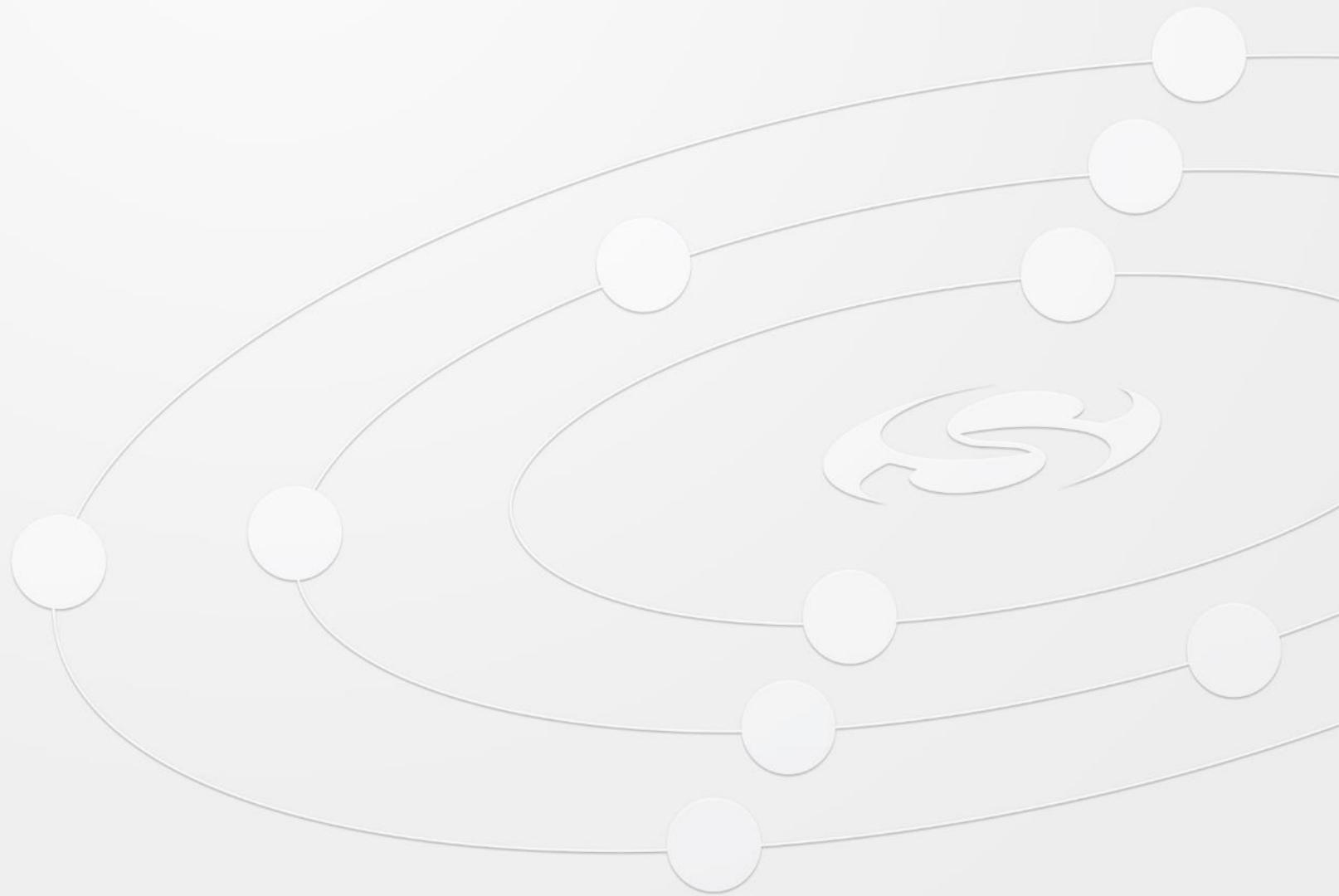
# 天猫精灵控制LCGW Zigbee子设备 (2)



5. 输入要绑定的淘宝用户帐户和密码
6. 轻触“授权并登录”按钮
7. 轻触“授权”按钮确认操作
8. 介面提示“绑定成功”

# LCGW软件质量保证

WWW.SILABS.COM



# SQA : LCGW功能测试和大网络测试

## ■ 基于Jenkins的自动化LCGW固件编译和SQA测试

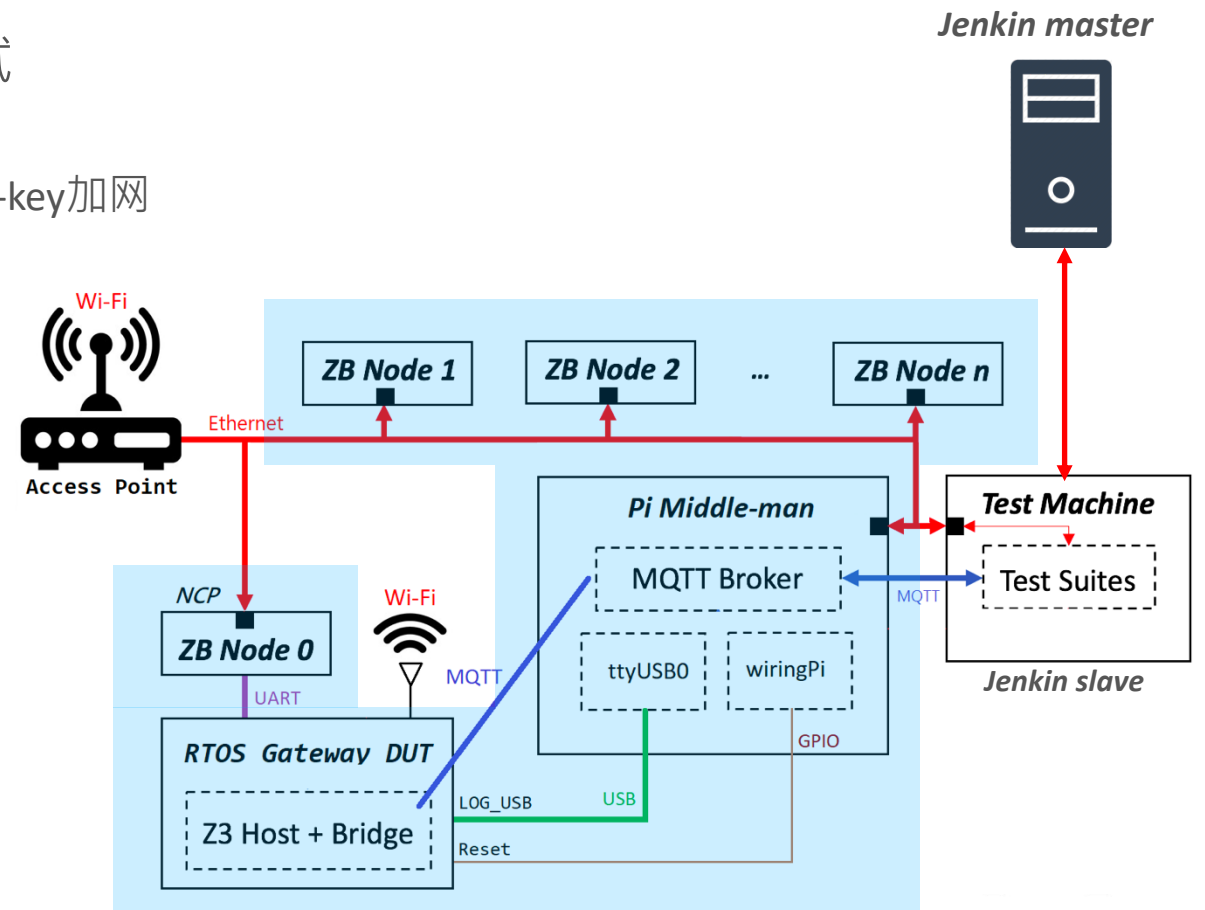
1. 网关网络创建(自动与特定设置)/离开网络
2. Zigbee设备使用Z3 Install Code加网或使用默认全局link-key加网
3. 网关命令 (单播和广播)
4. 网关命令 (分组和多播)
5. 网络性能 (单播、多播、广播)
6. 绑定设备，设置属性报告
7. 网关重启，设备重启
8. 设备OTA更新固件，网关在OTA期间发送命令
9. 网关NCP固件更新
10. 设备离网并重新加入

## ■ (一) 大型网络测试(50个节点)

- 模拟真实网络部署的露天测试

## ■ (二) 小型网络功能测试(9节点)

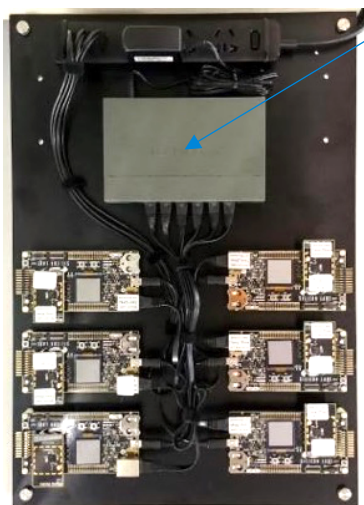
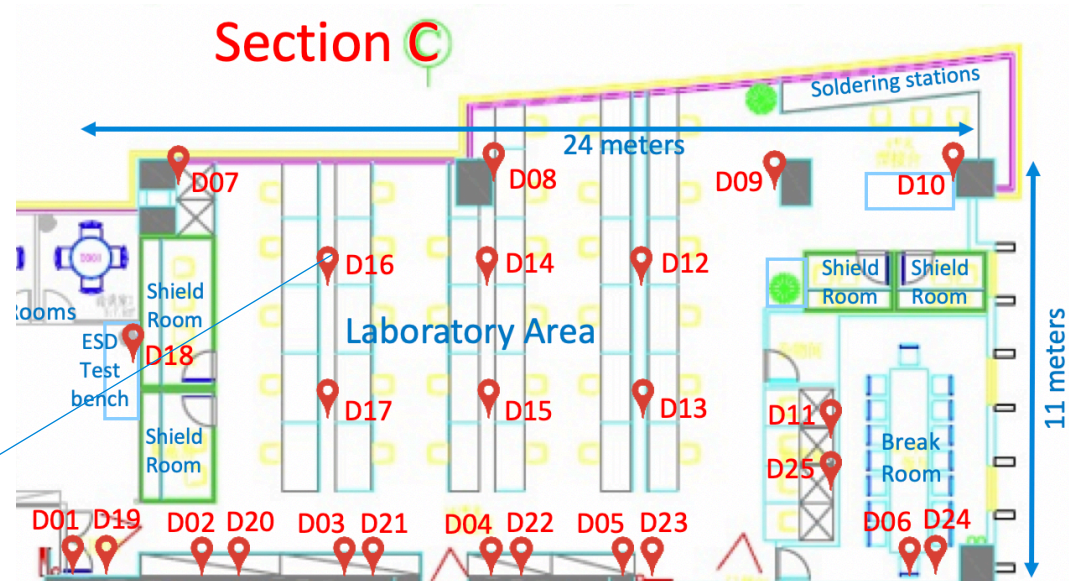
- 屏蔽箱中功能测试以消除外部射频干扰



\*Regression test: 在屏蔽箱中进行

# Silicon Labs深圳办事处：RTOS网关相关的测试设施

150节点网状测试网



6节点测试网络连接板



LCGW和Zigbee设备功能试验台



# Silicon Labs 深圳办事处：150节点网状测试网



- 总共有18个以太网端口位置，每个位置相距3~4米，并安装了1或2个测试网络连接板
- 总共安装了25个测试网络连接板：25 x 6 = 150个节点

# LCGW演示

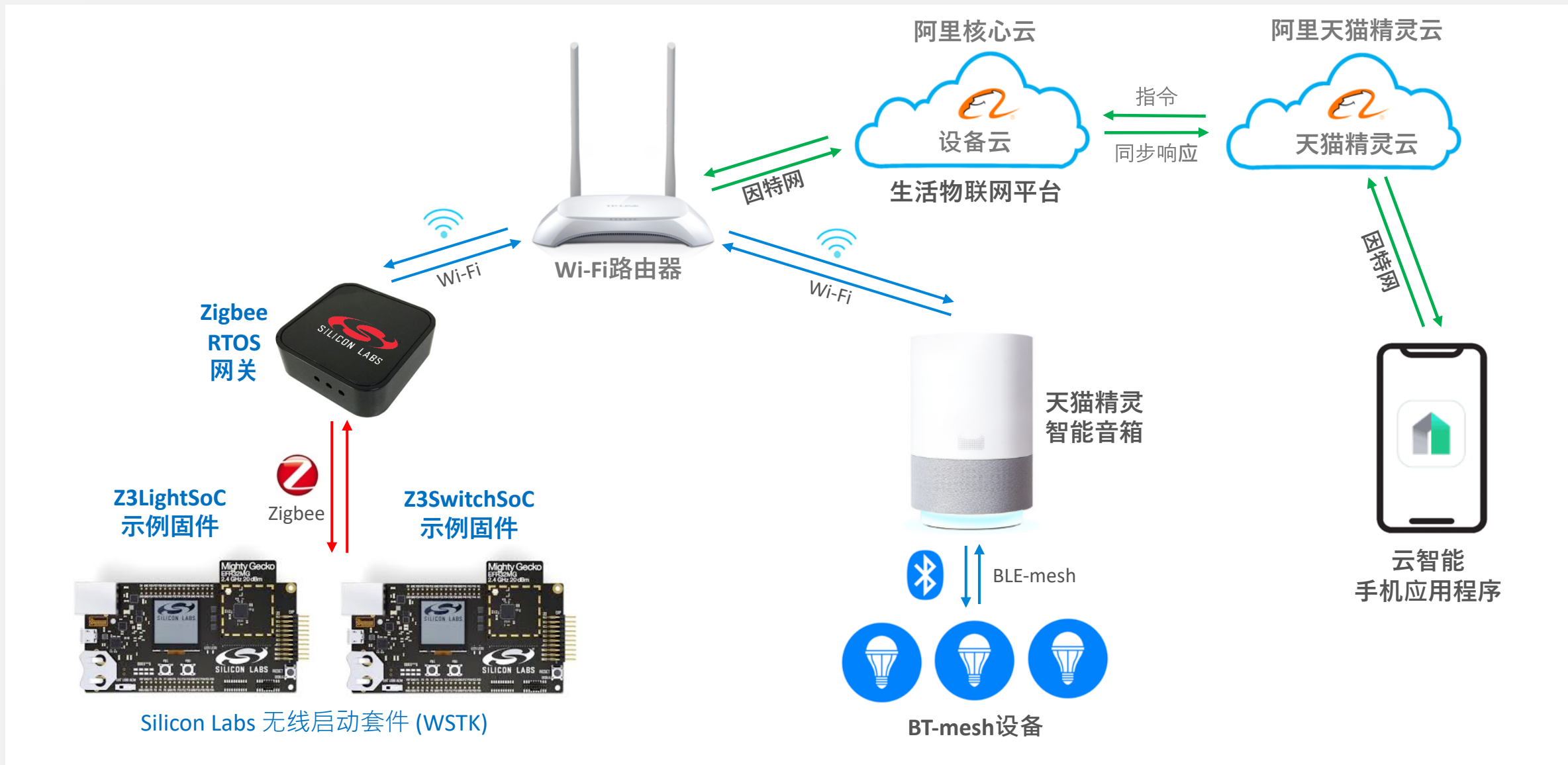
WWW.SILABS.COM



# 1. LCGW演示：阿里云物联网平台



## 2. LCGW演示：生活物联网平台



## 第二天：问答环节



谢谢

