

艾克姆科技

# nRF52840 开发指南-Mesh

[基于 Nordic 蓝牙低功耗/802.15.4 Soc-nRF52840]

艾克姆科技飞宇团队

[2020.4.15]

[本文档以艾克姆科技 IK-52840DK 开发板为硬件平台，基于 SDK16.0 的库，针对 BLE 主机开发，从基本概念和新建 BLE 主机工程模板开始，一步步讲解蓝牙 BLE 主机程序框架和流程：扫描、建立连接、MTU 交换、连接参数管理、PHY 更新、服务发现、使能通知、接收通知以及读写从机特征值等]

1 / 18

# 第一章：MESH：运行第一个 MESH 例子

写在前面话：

BLE Mesh 涉及到了很多新的概念，从现在开始我们会逐步接触到这些概念，艾克姆科技 Mesh 的教程是从开发者角度编写的，教程从运行第一个 MESH 例子开始，逐步讲解开发环境的搭建、代码的实现、Mesh 规格在代码中的体现。同时，Mesh 使用了 SES 集成开发环境，很多开发者熟悉的是 keil、IAR 集成开发环境，因为也需要花时间来熟悉 SES。

总之，学习 MESH 是需要花费一定时间和精力，“枯燥”的学习现在开始，“Are you ready? Go!”。

## 1. 学习目的

1. 逐步理解 BLE Mesh 涉及的基本概念，对于红字标注的概念一定要做到深刻理解。
2. 了解 Nordic 发布的用于 Mesh 的 SDK。BLE Mesh 会使用新的 SDK，Nordic 发布的用于 Mesh 的 SDK 名称是：nrf5SDKforMesh，当前版本是 V4.0，下文简称 Mesh SDK。
3. 了解 MESH 实验的流程。

## 2. 实验概述

蓝牙从 4.0 版本开始，加入了全新的蓝牙低功耗技术（BLE），BLE 最重要的特点就是省电，可以使一粒纽扣电池连续工作数年，同时由于 BLE 优异的互操作性、安全性等诸多特性，让 BLE 技术迅速得以大范围应用，各种低功耗设备如 beacon、手环、键鼠、智能遥控器、智能锁层出不穷，可以说“BLE”已无处不在。

自从 BLE 发布以来，广大用户一直关心一个问题：BLE 支持 Mesh 组网吗？终于，蓝牙技术联盟（SIG）不负众望，于 2017 年 7 月发布了 Mesh 的首个版本。各个芯片厂商也开始发布 Mesh 的 SDK，Nordic Semiconductor 作为 ULP 无线技术全球领先的方案厂商，在 Mesh 规格发布的当天即推出首个 Mesh SDK 版本，以帮助开发者快速构建基于蓝牙 Mesh 的应用，截止当前时间，Mesh SDK 已更新到 4.0 版本。

### ■ 什么是 Mesh

BLE Mesh 实现的是无线设备之间的多对多（many-to-many m:m）的互通性。Mesh 能让我们创建基于成千上万个设备的大型网络，网络中的设备可以互相安全可靠地通信，同时支持数据中继，从而可以辐射更大的物理区域。蓝牙 MESH 支持一个网络中最多 32767 个设备，最大网络直径为 126 跳。由此可见，BLE Mesh 无疑为大型设备网络的理想之选，可以预见，BLE Mesh 必将在楼宇自动化，无线传感器网络，资产跟踪等应用中大显身手。

### ■ 第一个实验实现的功能

对于无线实验来说，最经典的就是点灯实验，本章我们要运行的第一个实验同样是灯光控制实验，即 Mesh SDK 里面的“light\_switch”实验。

在进行“light\_switch”实验之前，我们需要先了解一些基本的概念，以方便我们执行实验步骤（重要概念）。

- Device（设备）：能够被配置到 Mesh 上的实体，也就是 Mesh 中的节点在配置前称为 Device。
- Node（节点）：经过 Provisioner 配置的 Device，Node 可以在网状网络中发送或接收消息。
- Provisioner（预置器）：能够将设备添加到网状网络的节点。

因此，对于 MESH 网络来说，一个设备必须经过 Provisioner 的配置才能够入网，该设备在没入网之前称为 Device，配置成功加入网络后称为网络中的 Node，设备成为 Node 后即可收/发消息了。

接下来，我们完成“light\_switch”实验，使用手机作为 Provisioner，直观地感受一下配置过程。

## ■ 软硬件需求

表 1-1：第一个 MESH 实验软硬件需求

序号	名称	描述
1	nRF.Mesh-V2.1.4.apk	安卓手机端 APP。IOS 系统自行到苹果商店下载安装。
2	IK-52840DK 开发板	至少 2 块。
3	J-LINK V9 仿真器（含转接板和转接线）	1 个。
4	USB 数据线	若干。

## 1. 实验步骤

本实验中使用手机作为 Provisioner 配置设备入网，实验示意图如下，这里我们直接使用 Mesh SDK 里面已经编译好的“light switch”固件。

将两块开发板分别烧写“light switch Server”固件和“light switch Client”固件，之后通过手机分别对它们进行配置，配置成功后，按下 Client 上的 S1 按键，Server 上的指示灯 D1 点亮，按下 Client 上的 S2 按键，Server 上的指示灯 D1 熄灭。

可以看到，这里面有 3 个主要的步骤。

1. 烧写固件：固件烧写后，开发板就是一个“能够被配置到 MESH 上的实体”，这时它是“Device”。
2. 配置：Provisioner 将烧写好固件的开发板配置为网络中的节点，配置完成后开发板成为“Node”。
3. 收/发消息：按下 Client 上的按键控制 Server 上指示灯的亮灭。

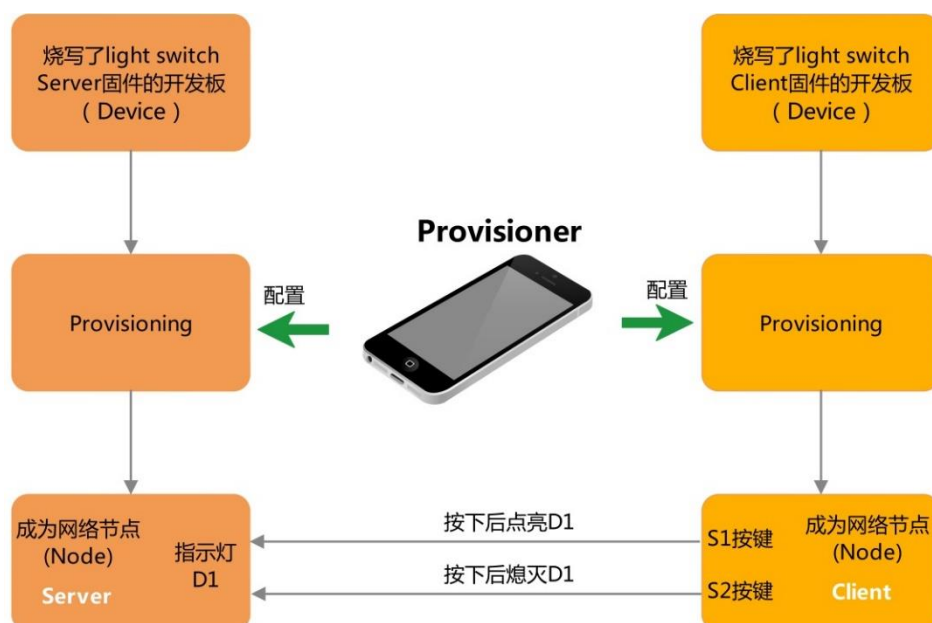


图 1-1: 手机作为 Provisioner 实验示意图

## 1.1. 烧写和配置 Sever

Mesh SDK 中提供了已经合并好协议栈的 Light switch 例子的固件，在 Mesh SDK 中的路径：`nrf5_sdk_for_Mesh_v4.0.0_src\bin\merged\ospace\examples\light_switch`。

### 1. 烧写作为 Sever 的 IK-52840DK 开发板

- 打开 J-Flash，将作为 Sever 的 IK-52840DK 开发板烧写固件 “light\_switch\_server\_nrf52840\_xxAA\_s140\_7.0.1\_merged\_sd.hex”。
- 烧写成功后，开发板的 4 个 LED 指示灯同时闪烁 2 次，指示程序正常运行。

### 2. 配置 Sever

- 1) 手机上打开 APP：“nRF Mesh”，点击“ADD NODE”启动扫描，扫描到“nRF5x Mesh Light”后，点击“nRF5x Mesh Light”，如下图所示。

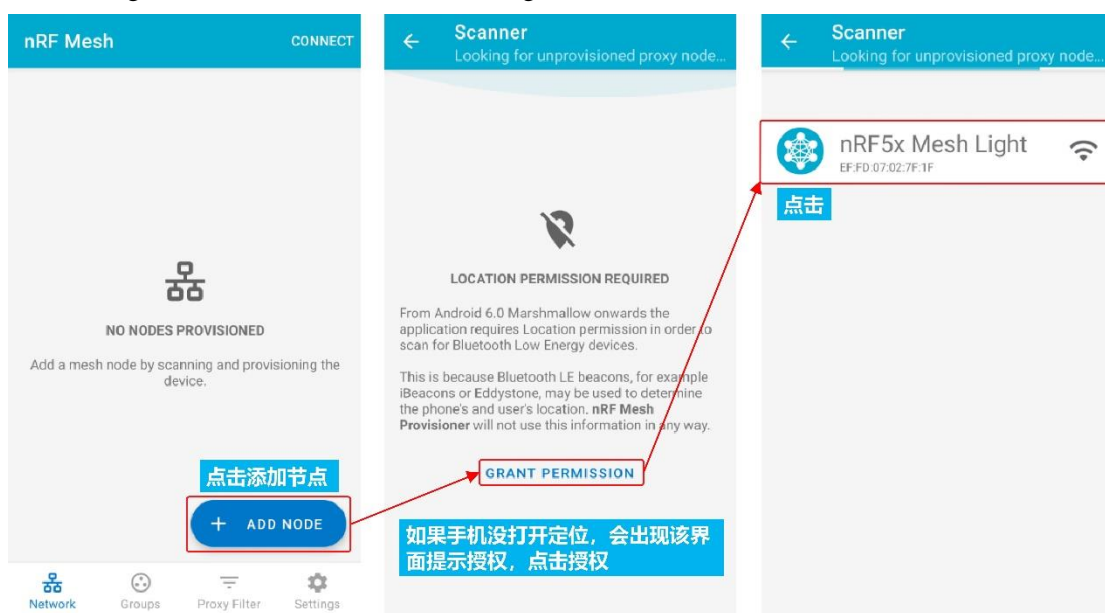


图 1-2: 扫描设备

- 2) 点击[IDENTIFY], 之后会观察到开发板上指示灯 D3 和 D4 快速闪烁一段时间。点击[PROVISION]启动配置流程, 配置过程需要一段时间, 在这个过程中会观察到开发板的 4 个指示灯 D1~D4 同时闪烁 3 次表示配置成功, 之后弹出配置完成窗口, 点击[OK]完成配置。

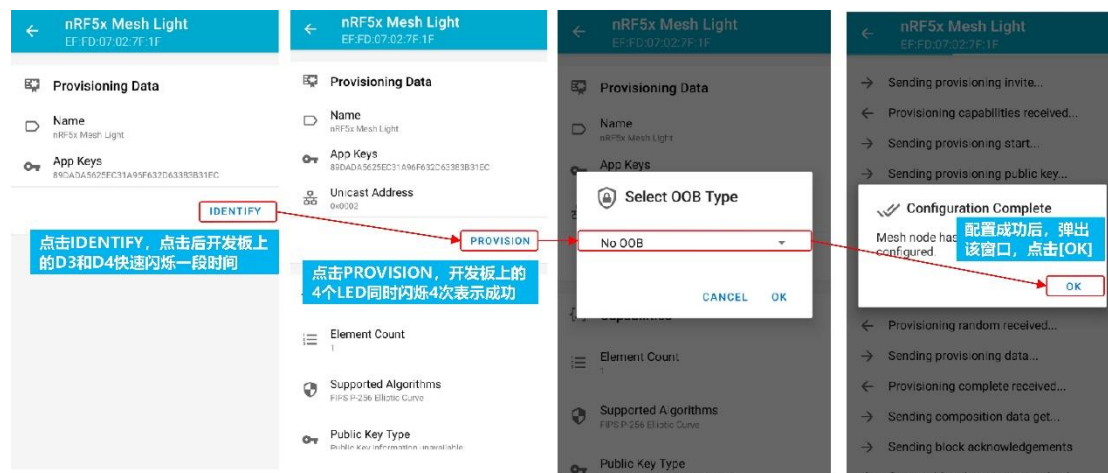


图 1-3: 配置 Server

- 3) 点击配置按钮, 之后点击 “Generic On Off Server”。

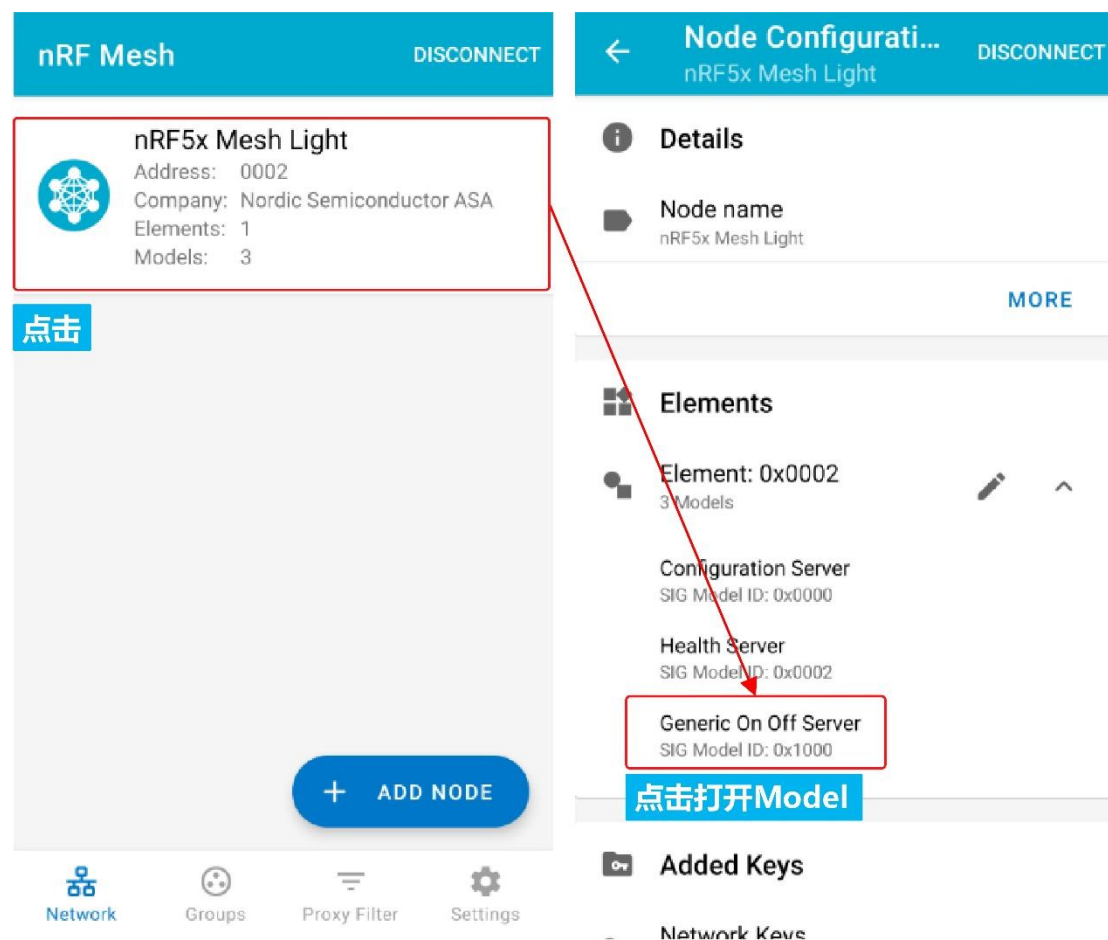


图 1-4: 打开 Sever Model

- 4) 点击[BIND KEY], 绑定 NetKey, 如下图所示。

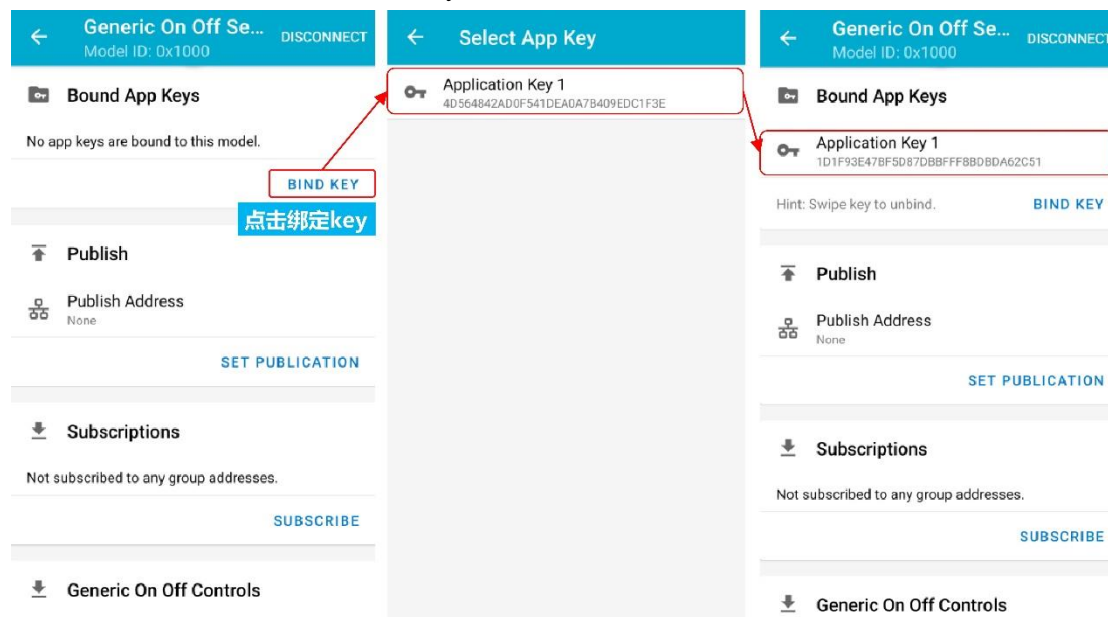


图 1-5: 绑定 NetKey

- 5) 绑定 NetKey 完成后, 可以通过 APP 控制开发板上的指示灯 D1 的亮灭和读取指示灯状态, 如下图所示。

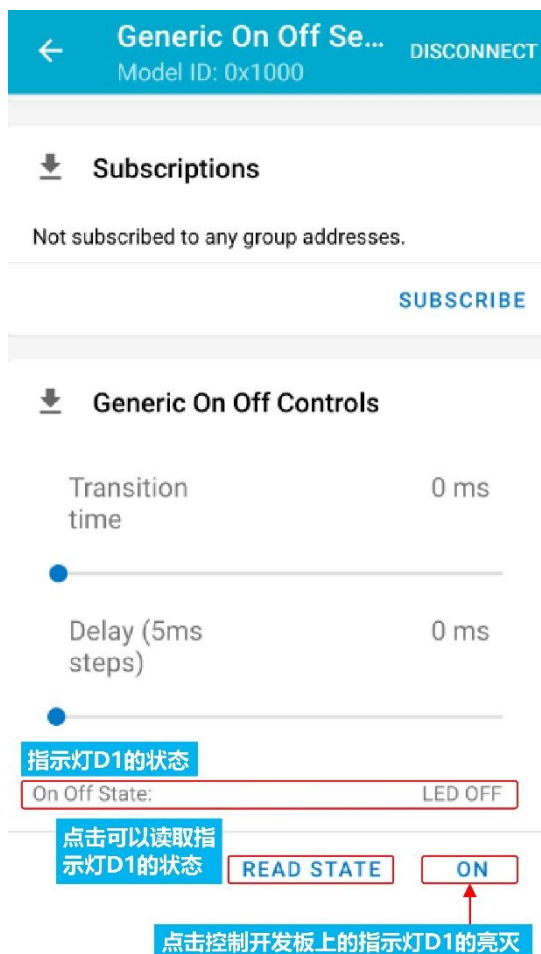




图 1-6: APP 控制 D1 和读取 D1 状态

## 1.2. 烧写和配置 Client

### 1. 烧写作为 Client 的 IK-52840DK 开发板

- 1) 打开 J-Flash, 将作为 Client 的 IK-52840DK 开发板烧写 “light\_switch\_client\_nrf52840\_xxAA\_s140\_7.0.1\_merged\_sd.hex” 固件。
- 2) 烧写成功后, 开发板的 4 个 LED 指示灯同时闪烁 2 次, 指示程序正常运行。

### 2. 配置 Sever

- 1) 手机上打开 APP: nRF Mesh, 点击 “+” 启动扫描, 扫描到 “nRF5x Mesh Switch” 后, 点击 “nRF5x Mesh Light”, 如下图所示。

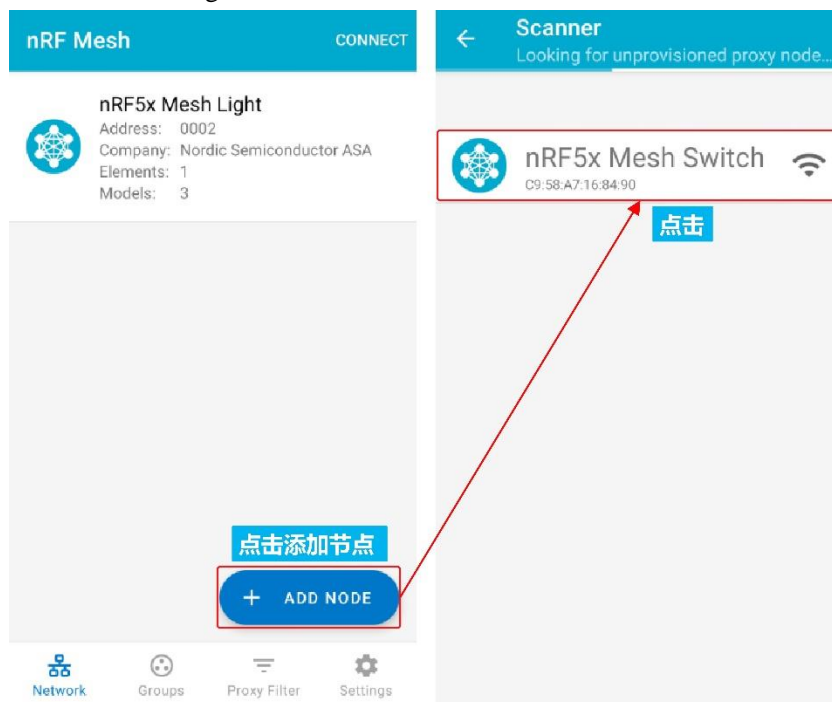


图 1-7: 扫描设备

- 2) 点击[IDENTIFY], 之后会观察到开发板上指示灯 D3 和 D4 快速闪烁一段时间。点击[PROVISION]启动配置流程, 配置过程需要一段时间, 此过程中会观察到开发板的 4 个指示灯同时闪烁 3 次表示配置成功, 之后弹出配置完成窗口, 点击[OK]完成配置。

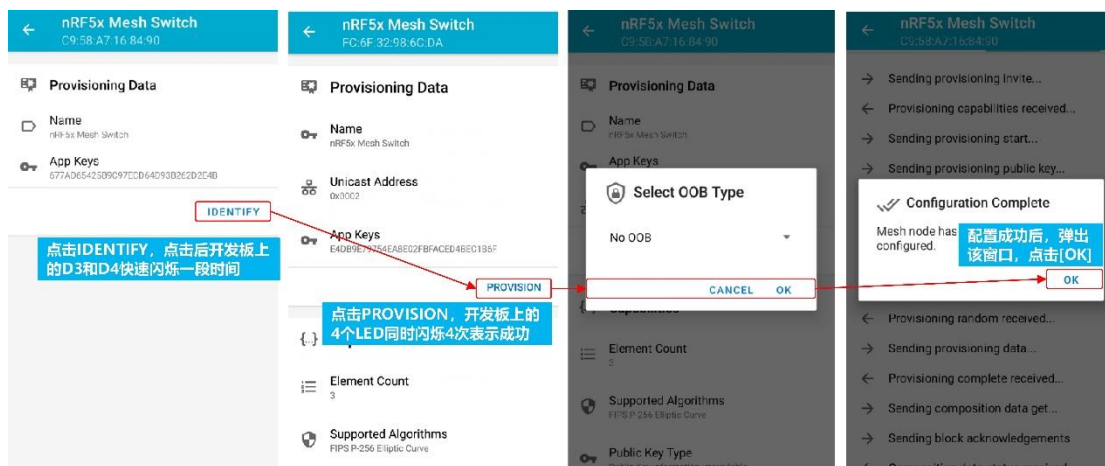


图 1-8: 配置 Client

3) 点击配置按钮, 之后点击展开“Element:0x0004”列表, 如下图所示。

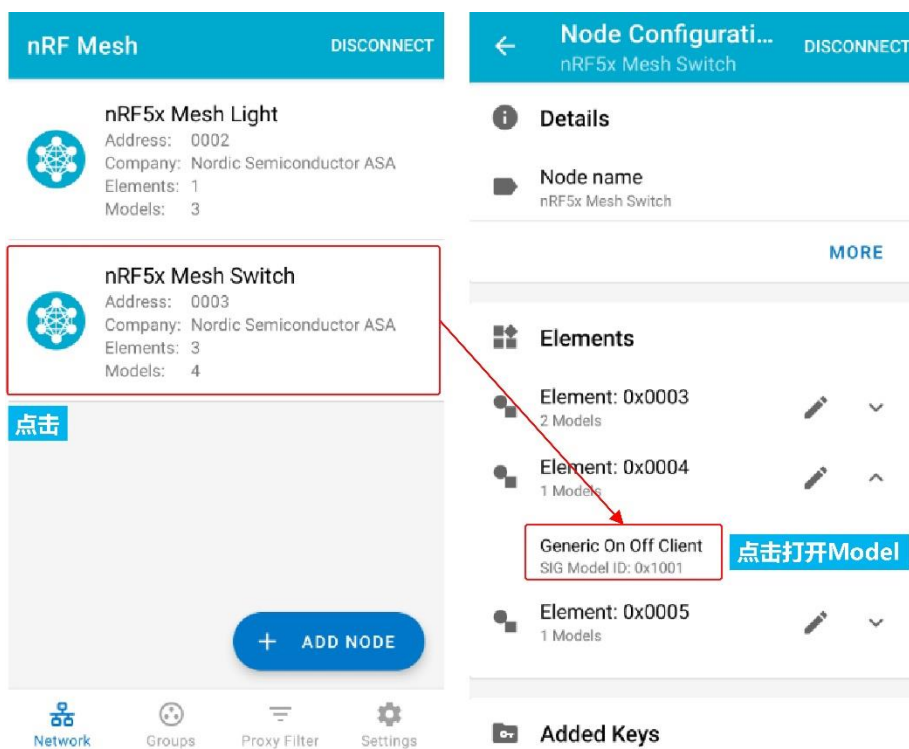


图 1-9: 展开“Element:0x0004”

4) 点击[BIND KEY], 绑定 NetKey, 如下图所示。



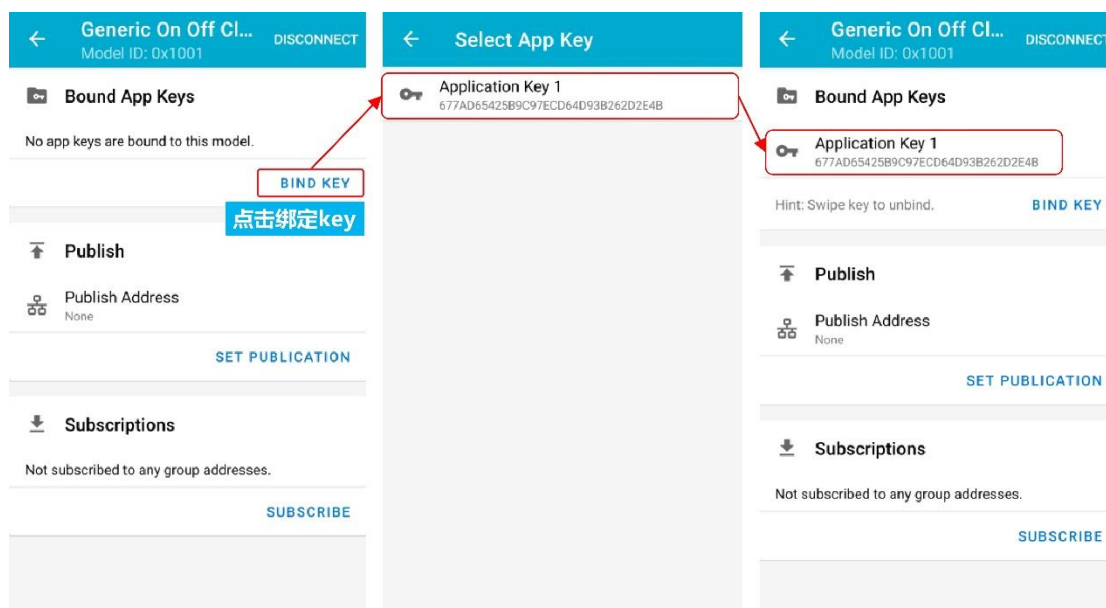


图 1-10：绑定 NetKey

5) 点击“nRF5x Mesh Light”，查看 Sever 的单播地址，如下图所示。

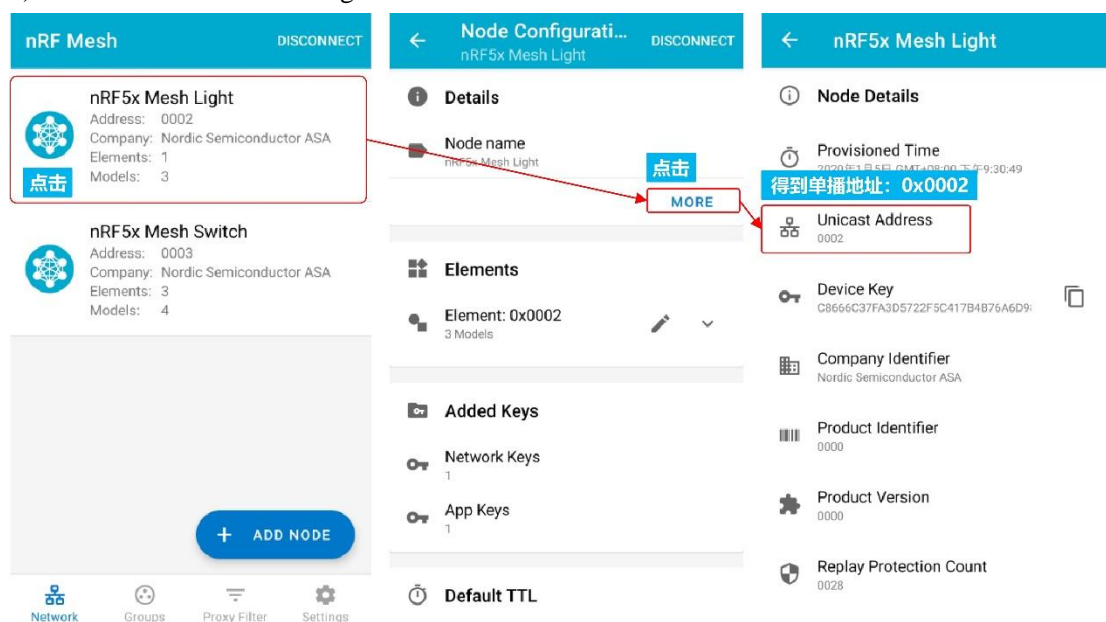


图 1-11：获取 Sever 的单播地址

6) 设置 Client 的 Publish 地址，注意地址输入后一定要点击“APPLY”，如下图所示。

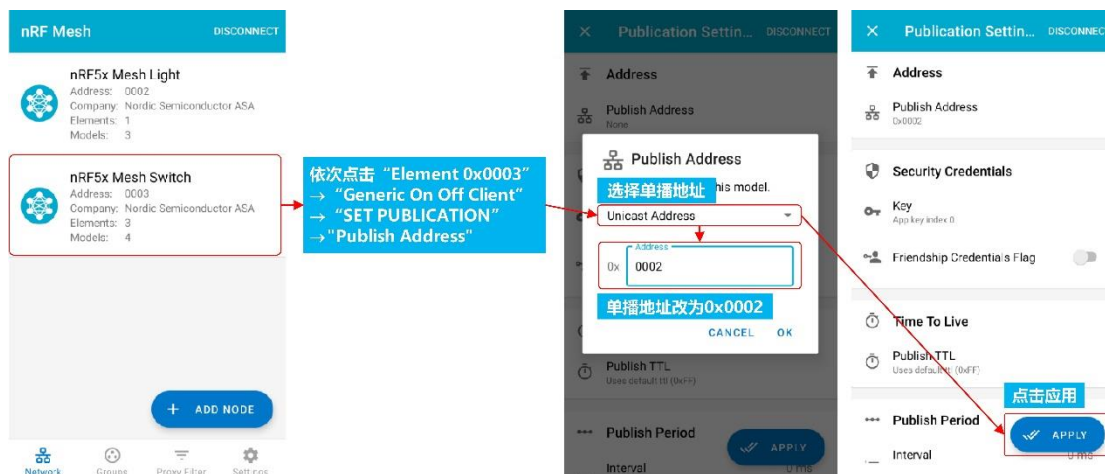


图 1-12: 设置 Client 的 Publish 地址

### 1.3. 测试

经过前面的步骤，两块开发板均已配置入网，通过 Client 上的按键 S1 和 S2 即可发送消息控制 Server 上的指示灯 D1 的亮灭。

- 按下 Client 上的 S1 按键，Server 上的指示灯 D1 点亮，
- 按下 Client 上的 S2 按键，Server 上的指示灯 D1 熄灭

我们还可以继续将一块开发板烧写 Sever 的固件，烧写后按照前文描述的方式配置，配置成功入网后，可以通过 Client 上的按键 S3 和 S4 发送消息控制 Server 上的指示灯 D1 的亮灭。当然，我们可以烧写更多的 Server 并配置入网，

## 第二章：搭建 Mesh 开发环境

### 1. 需要的工具软件

表 2-1: 需要的工具软件

序号	软件工具	描述
1	SEGGER Embedded Studio (SES)	SEGGER 推出的集成开发环境。
2	nRF Mesh.apk	安卓手机端 APP。
3	nRF5_SDK_16.0.0_98a08e2.zip	nRF5 软件开发包。
4	nrf5_SDK_for_Mesh_v4.0.0_src.zip	Mesh 软件开发包。

## 1.1. SEGGER Embedded Studio

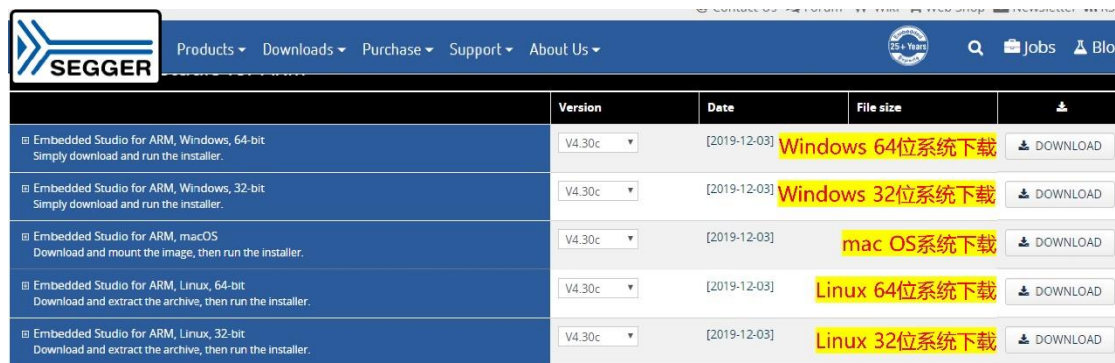
SEGGER Embedded Studio, 简称 SES, 是 SEGGER 公司推出的一款嵌入式跨平台集成开发环境, 支持 Windows、Linux 和 MAC 系统。

做单片机、嵌入式开发的朋友有可能对 SEGGER 公司不了解, 但是相信大多都用过或听说过“J-LINK 仿真器”, 没错, SES 正是设计 J-Link 仿真器的公司 SEGGER 推出的嵌入式集成开发环境。更为重要的是, SES 针对非商业用途是免费的, 如果使用 SES 开发 Nordic 的芯片, 可以直接申请免费的授权许可。

■ 开发板配套资料包里面已经下载好 Window 系统下的 SES 安装文件, 安装文件在开发板资料包的“教程盘(B 盘)\7 - Mesh\1: 所需工具软件”目录下。

■ 如果我们使用的是 MAC 或者 Linux 系统, 可以到 SEGGER 的网站下载对应的版本。

下载链接: <https://www.segger.com/downloads/embedded-studio/>。打开后界面如下, 根据自己电脑系统选择对应的版本下载即可。



	Version	Date	File size	
Embedded Studio for ARM, Windows, 64-bit Simply download and run the installer.	V4.30c	[2019-12-03]	Windows 64位系统下载	DOWNLOAD
Embedded Studio for ARM, Windows, 32-bit Simply download and run the installer.	V4.30c	[2019-12-03]	Windows 32位系统下载	DOWNLOAD
Embedded Studio for ARM, macOS Download and mount the image, then run the installer.	V4.30c	[2019-12-03]	mac OS系统下载	DOWNLOAD
Embedded Studio for ARM, Linux, 64-bit Download and extract the archive, then run the installer.	V4.30c	[2019-12-03]	Linux 64位系统下载	DOWNLOAD
Embedded Studio for ARM, Linux, 32-bit Download and extract the archive, then run the installer.	V4.30c	[2019-12-03]	Linux 32位系统下载	DOWNLOAD

图 2-1: SES 下载页面

❖ **说明:**开发 Nordic 系列的芯片是可以免费使用 SES 的(Nordic 已经为开发者购买了 SES 使用版权), 也就是说使用 SES 开发 Nordic 的芯片无需担心版权的问题。

## 1.2. 手机端 APP

Mesh 手机端用的调试 APP 是“nRF Mesh”, 安卓系统将资料包里面的安装文件发送到手机安装即可。安装文件在开发板资料包的“教程盘(B 盘)\7 - Mesh\2: 手机端 APP”目录下。

## 2. 安装 SES

### 2.1. 安装

1. 双击 SES 安装文件, 弹出 SES 安装向导, 单击【Next】。

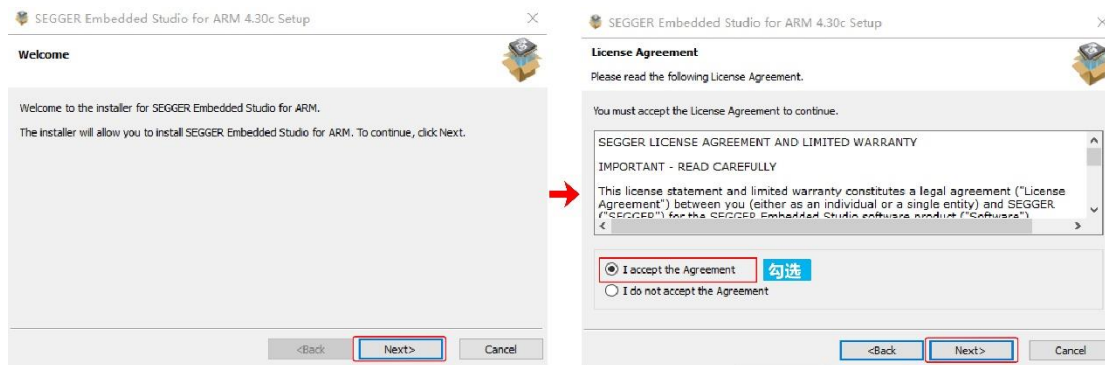


图 2-2: SES 安装

2. 设置安装路径后，单击【Next】，建议使用默认的安装路径。

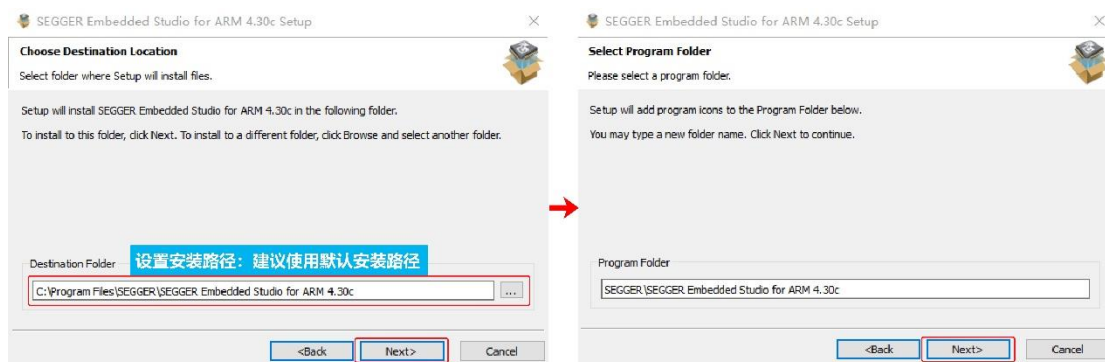


图 2-3: 设置安装路径

3. 设置文件关联和安装的附加组件后，单击【Next】。

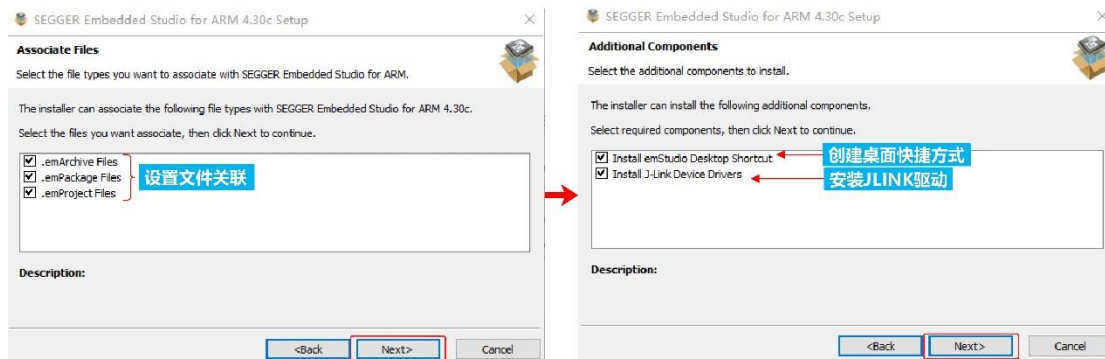


图 2-4: 设置文件关联和安装的附加组件

4. 等待 SES 安装进度完成后，单击【Finish】完成安装。

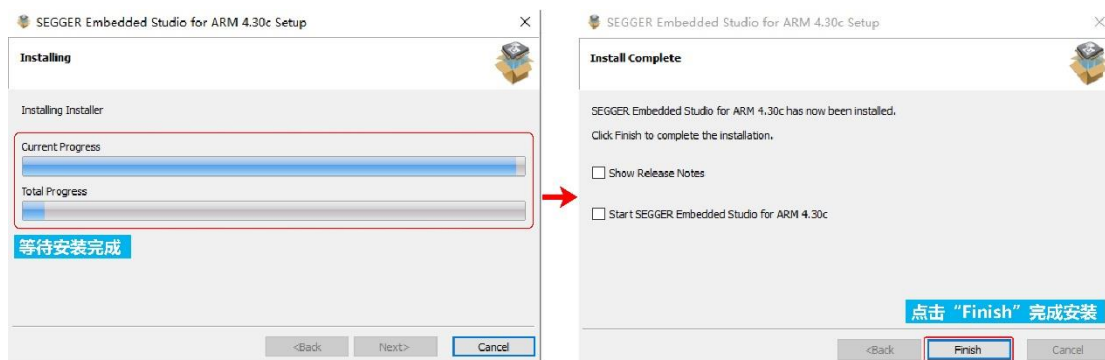
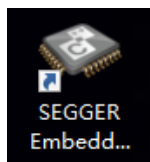


图 2-5: 完成安装

## 2.2. 激活 SES

SES 安装完成后, 我们还需要激活才能正常使用, 这里我们激活一个免费的许可, 步骤如下。

1. 双击 SES 桌面快捷方式图标, 启动 SES。



2. 如果弹出了许可申请窗口, 忽略这一步。如下图所示, 选中工程后右键, 在弹出的菜单中点击[Build], 因为我们还没有激活 SES, 因此执行编译就会弹出。

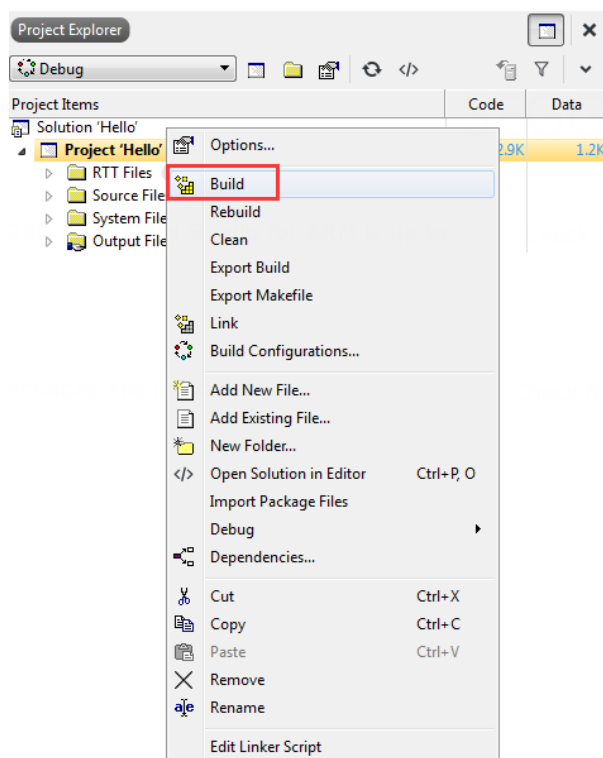


图 2-6: 打开许可申请窗口

3. 这时会弹出如下界面, 提示我们没有检测到许可, 点击“Activate Your Free License”申请许可。

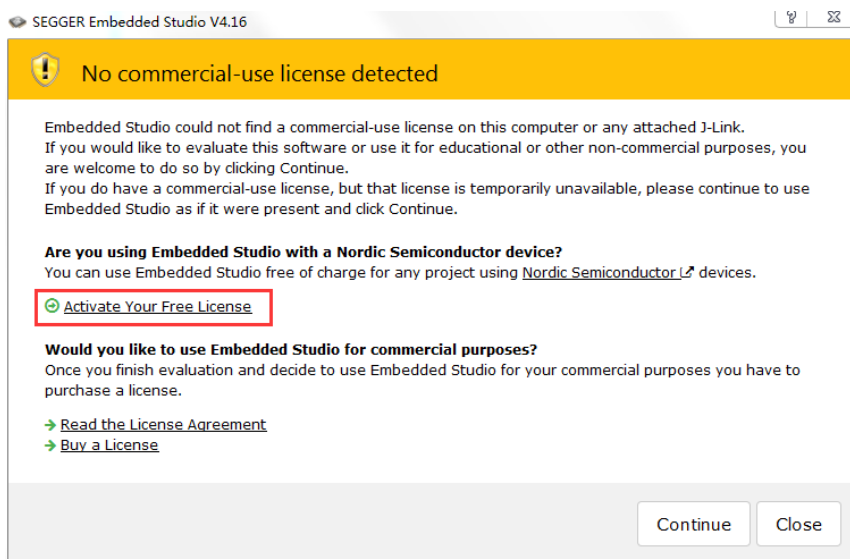


图 2-7: 打开许可申请窗口

4. 填写好相关信息后, 点击【Request License】, 序列号会发到申请的邮箱里面, 注意: 序列号发送可能会有延迟, 如果申请后邮箱没有接收到邮件, 稍等一段时间后就会收到邮件。

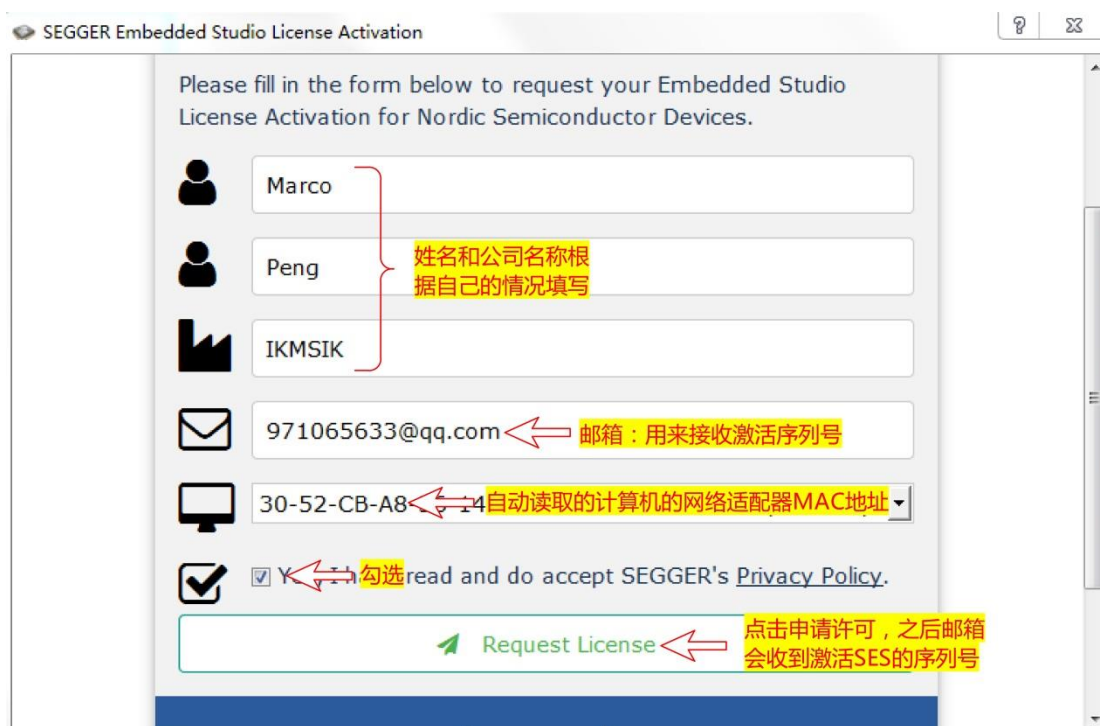


图 2-8: 申请序列号

5. 打开邮箱拷贝序列号, 如下图所示, 注意需要拷贝的内容。





图 2-9: 拷贝序列号

## 6. “Tools→License Manager” 输入序列号，完成激活。

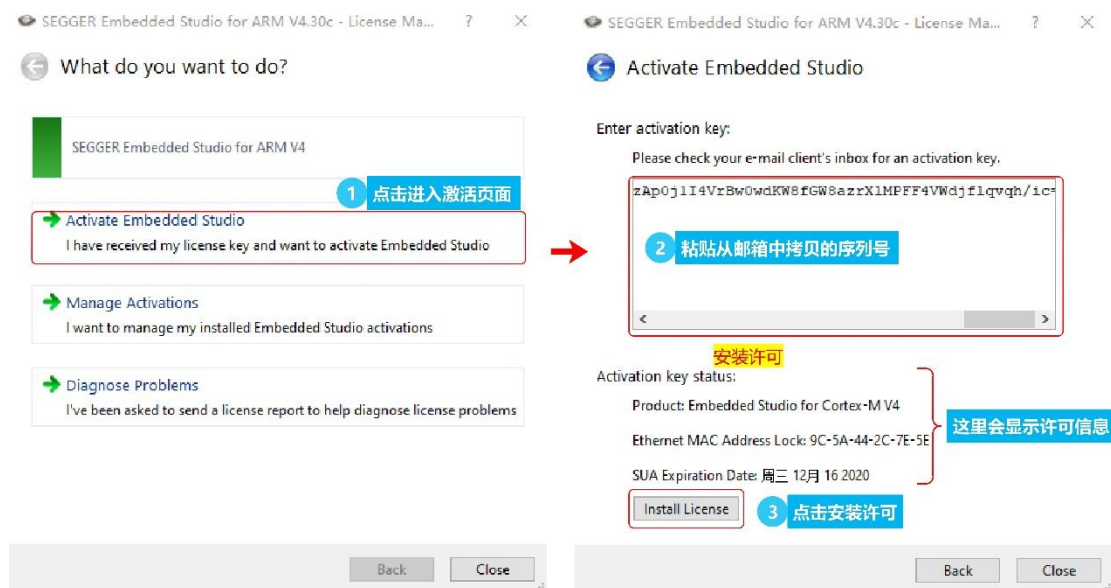


图 2-10: 激活 SES

❖ 注意: 如果申请序列号后关闭了窗口, 可以到“Tools→License Manager”中激活 SES。

## 2.3. 安装 Pack

执行“Tools→Package Manager”, 打开 Pack 管理器, 找到“Nordic pack”的安装项, 按照下图所示安装 nRF 的 Pack。

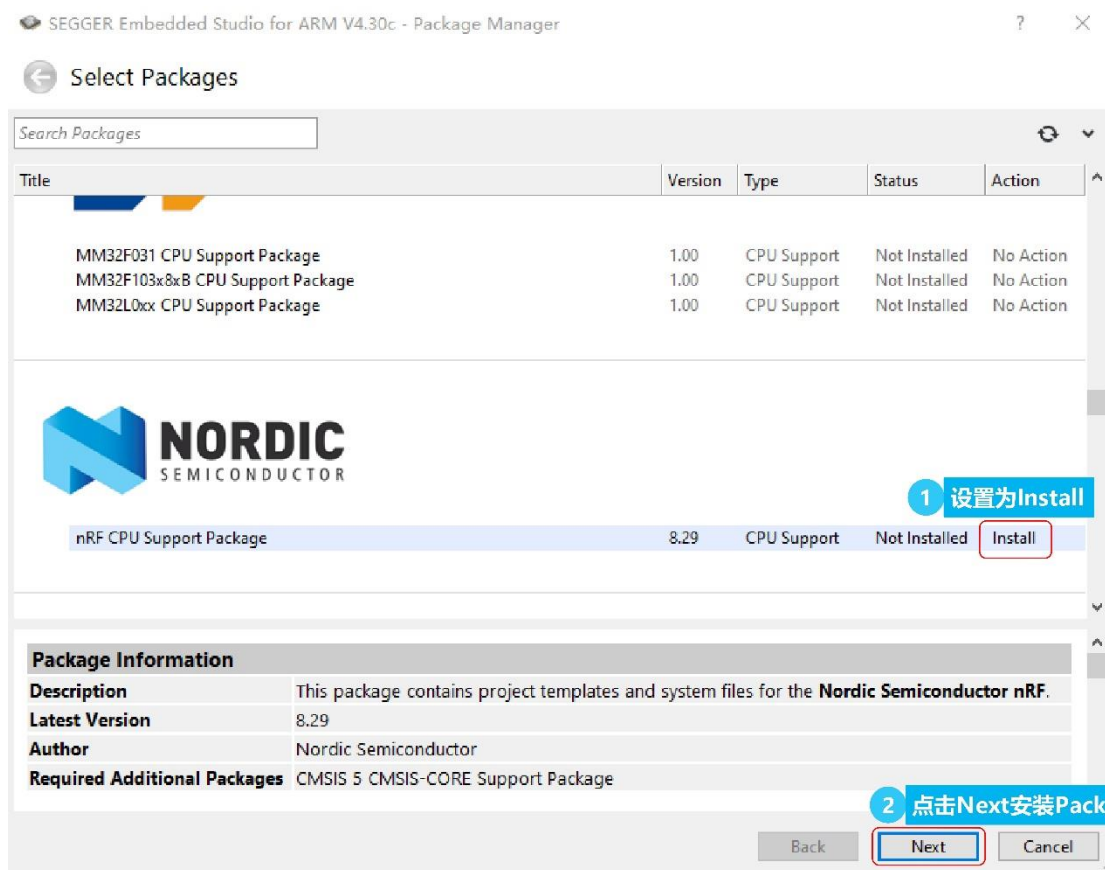


图 2-11：安装 Pack

Pack 安装需要保持联网，安装会占用一些时间，请耐心等待安装完成，安装完成后如下图所示，点击“Finish”完成 pack 安装。

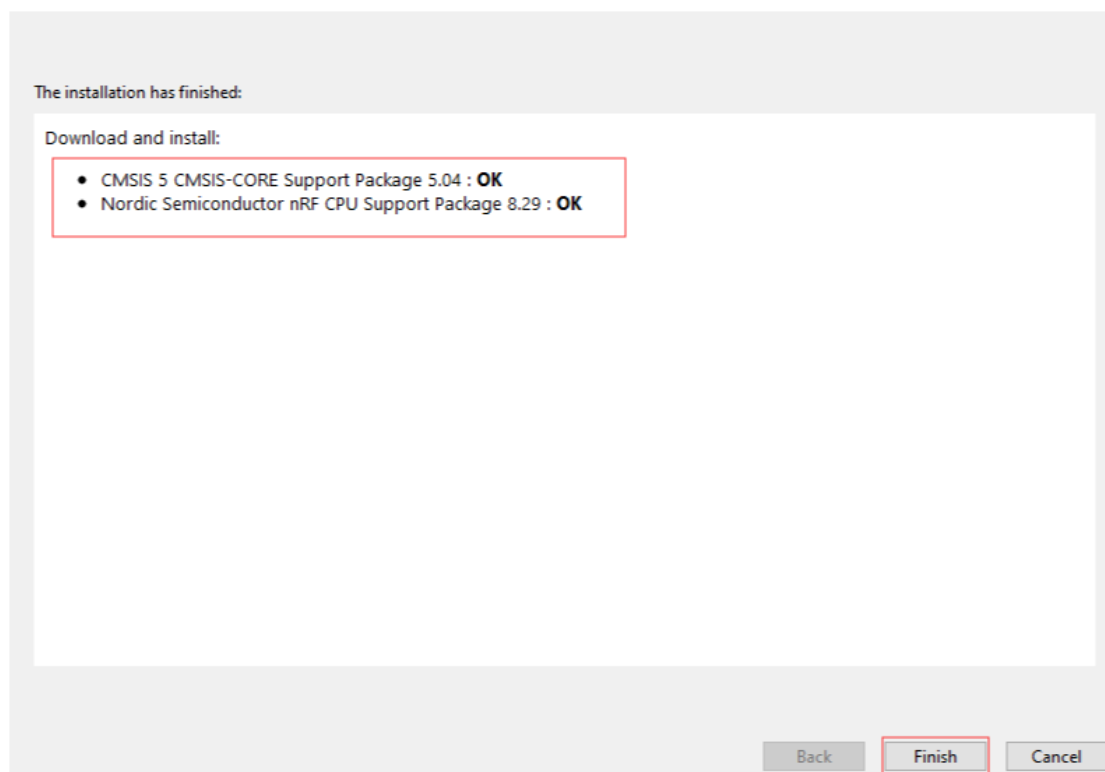


图 2-12: Pack 安装完成

### 3. 编译 MESH 工程

本节以 MESH SDK 中的“light\_switch”的“server”为例说明如何编译 MESH 的工程。MESH 的工程会用到 SDK 中的库文件，因此需要注意 SDK 的版本和放置的路径。

- SDK 版本: MESH SDK-V4.0.0 对应的 SDK 版本是 SDK16.0。
- 将 MESH SDK (V4.0.0) 和 SDK16.0 解压到同级目录下，如下图所示。

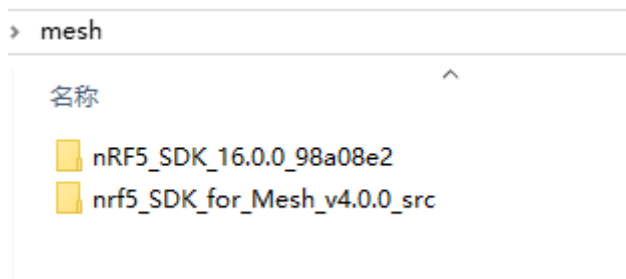


图 2-13: MESH SDK 和 SDK16.0 解压到同级目录

#### 1. 打开工程

打开“..\nrf5\_SDK\_for\_Mesh\_v4.0.0\_src\examples\light\_switch\server”目录，双击“light\_switch\_server\_nrf52840\_xxAA\_s140\_7\_0\_1.emProject”打开工程。

#### 2. 编译工程

点击编译按钮编译工程，如下图所示。

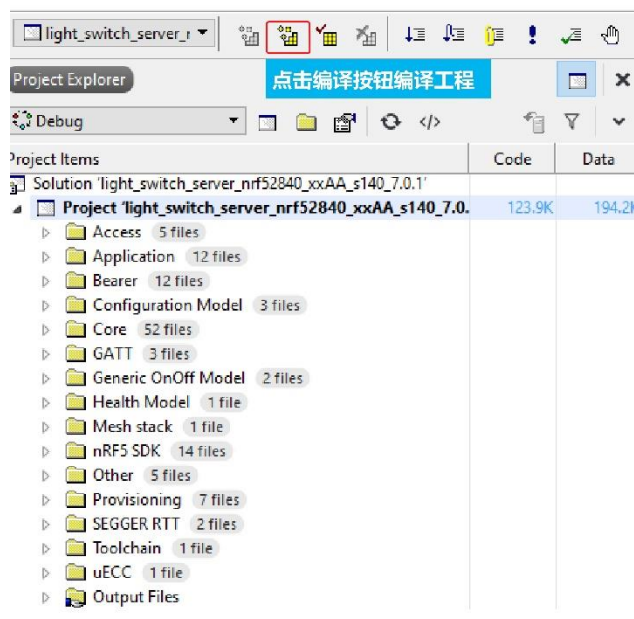


图 2-14: 编译工程

编译成功后，信息输出窗口会显示工程编译后的 Flash 和 RAM 大小，如下图所示。

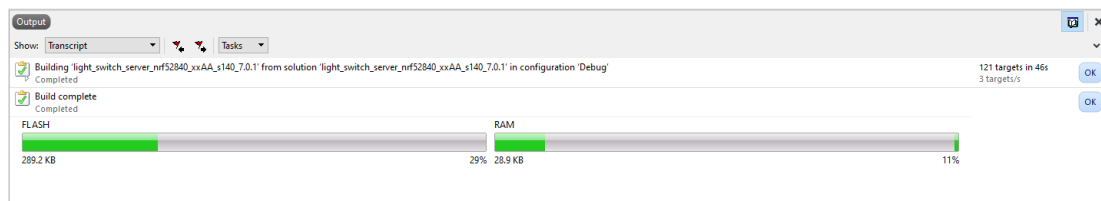


图 2-15：编译成功

## 参考文献

1. Cortex-M4 Technical Reference Manual。
2. nRF5\_SDK\_16.0.0\_98a08e2, Nordic Semiconductor。
3. nRF5\_SDK\_16.0.0\_offline\_doc, Nordic Semiconductor。
4. nRF52840\_PS\_v1.1, Nordic Semiconductor。
5. Bluetooth Core Specification 4.2, SIG。
6. Bluetooth Core Specification Addendum 6, SIG。
7. Bluetooth Core Specification 5.0, SIG。
8. Bluetooth Core Specification Addendum 7, SIG。
9. Device Information Service(DIS) V11r00, SIG。
10. Mesh Profile Specification 1.0.1, SIG。
11. Mesh Model Specification 1.0.1, SIG。
12. Mesh Device Properties 1.2, SIG。