

智能家电产业 NB-IoT应用及开发指导书

智能家电产业NB-IoT联合应用中心
2018年5月



编纂单位及组织：

中国家用电器研究院

中国电信集团有限公司

华为技术有限公司

天翼物联产业联盟

目 录

1 智能家电产业分析	1
1.1 智能家电产业现状.....	1
1.1.1 我国家电行业发展分析.....	1
1.1.2 智能家电产业发展分析.....	2
1.1.3 智能家电的发展瓶颈.....	4
1.1.4 WiFi 应用痛点.....	4
1.2 NB-IoT 技术背景.....	6
1.2.1 NB-IoT 技术简介.....	6
1.2.2 NB-IoT 与其他无线通信技术的对比.....	7
1.2.3 NB-IoT 应用于家电行业.....	9
1.2.4 NB-IoT 产业进展.....	10
1.3 基于 NB-IoT 的智能家电解决方案架构.....	12
1.3.1 终端.....	12
1.3.2 NB-IoT 网络.....	13
1.3.3 IoT 网关.....	13
1.3.4 智能家电应用.....	13
1.4 基于 NB-IoT 的智能家电应用进展.....	14
2 基于 NB-IOT 的智能家电开发指导	17
2.1 NB-IoT 智能家电开发集成流程介绍.....	17
2.1.1 智能家电与 IoT 网关对接开发流程.....	17
2.1.2 智能家电与 IoT 网络对接开发流程.....	18
2.2 NB-IoT 智能家电物联网卡.....	18
2.2.1 NB-IoT 物联网卡介绍.....	18
2.2.2 NB-IoT 智能家电物联网卡选择建议.....	25
2.3 NB-IoT 智能家电开发资源申请.....	25

2.3.1 NB-IoT 物联网卡申购	25
2.3.2 IoT 网关测试账号申请	26
2.4 智能家电与 IoT 网关对接开发	27
2.4.1 IoT 网关对接	27
2.4.2 终端网络质量数据上报	41
2.4.3 终端固件远程升级 (FOTA)	47
2.5 智能家电与 NB-IoT 网络对接开发	52
2.5.1 下行控制时延优化 (提前寻呼)	52
2.5.2 终端功耗优化 (快速释放连接)	55
3 智能家电 NB-IOT 应用测试验证	59
3.1 终端接入验证	59
3.2 终端与 IoT 网关对接验证	60
3.2.1 IoT 网关对接验证	60
3.2.2 终端网络质量数据上报	60
3.2.3 终端固件远程升级 (FOTA)	61
3.3 终端与网络对接验证	62
3.3.1 下行控制时延优化 (提前寻呼)	62
3.3.2 终端功耗优化 (快速释放连接)	62
4 附录	63
4.1 AT+MLWULDATAEX 命令 (以海思为例, 具体详见对应模组厂家指令集)	63
4.1.1 命令语法	63
4.1.2 命令描述	63
4.1.3 参数描述	63
4.2 AT 命令返回结果	64
4.2.1 华为海思芯片	64
4.3 设备升级能力定义	66

4.3.1 Profile 定义.....	66
4.4 FOTA 升级与 MCU 交互.....	67
4.5 编解码插件开发指导.....	67
4.5.1 Profile 模板开发.....	67
4.5.2 编解码插件开发.....	69
4.5.3 输入数据（码流，JSON 结构体）格式定义.....	79

第一册

智能家电 NB-IOT 技术应用

1 智能家电产业分析

1.1 智能家电产业现状

1.1.1 我国家电行业发展分析

我国家电行业是改革开放以来发展起来的新兴行业，在“十五”、“十一五”期间建立了一批有较高水平的骨干企业。上个世纪 90 年代以后随着国有企业的产权改制，民营资本大量涌入，加快了产业结构调整，带动了家电行业的重组和整合，加上外资企业纷纷来到中国，使我国的家电行业进入快速发展时期。家电行业在提高人民生活水平、繁荣经济、为国家积累资金方面发挥了巨大作用。

依据工信部、中商产业研究院数据，2017 年我国家电行业主营业务收入 1.51 万亿元，同比增长 18.7% (2017 年统计口径有所调整，调整后，2016 年主营业务收入为 1.27 万亿元，调整前为 1.46 万亿元)，其中：房间空调器累计生产 18039.8 万台，同比增长 26.4%；家用电冰箱累计生产 8670.3 万台，同比增长 13.6%；家用洗衣机累计生产 7500.9 万台，同比增长 3.2%；厨房电器市场表现亮眼，得益于宏观消费环境、产业内部结构升级、新兴品类市场爆发等多方面积极因素的集中释放，零售额规模为 794 亿元，同比增长 14.3%；小家电行业 2017 年上半年整体实现主营收入 218.72 亿元，同比增长 20.89%。

我国家电行业的发展呈现出以下 4 个特点：

(1) 中国已成为全球家电的生产大国

中国大陆已是全球最大的白色家电生产基地，约占全球白电产能的 60%-70%，其中空调器约占全球 80% 的产能，电冰箱和洗衣机约占 50% 左右。我国空调器、电冰箱、洗衣机、微波炉、电风扇、电饭锅以及众多小家电的产量均已位于世界前列，家电行业已经成为国民经济中的重要组成部分。

(2) 我国已成为家电产品的出口大国和全球主要供应商

随着全球性产业转移浪潮，由于我国具有的零部件配套支撑和低成本劳动力等优势，我国家电产品的出口大幅度增加。产品不仅出口到东南亚等发展中国家，而且大量出口到美国、欧洲等发达国家。2017 年我国空调器、电冰箱、洗衣机的总销量中，出口占比分别为 37%、40%和 31%。

(3) 家电企业逐渐重视开发自主的关键技术

家电行业是技术密集型产业，涉及了机械、电子、化工、环保等领域的科技成果，产品正在向多功能、高品质、节能、静音等方向发展。网络化、智能化、节能、环保是全球家电行业发展的主题，对于已完全融入全球经济的中国家用电器行业来讲，这些技术理念已逐步贯穿到产品开发、市场营销、企业发展战略等各个环节，并将持续深入。

(4) 家电企业综合成本居高不下、经营风险加大，效益下滑

人民币的升值加大了家电产品的出口难度，另外，原材料价格波动剧烈和劳动力成本大幅增加，这些都使得众多家电企业近年来的效益明显下降，家电行业进入微利时代，行业内的重组和整合趋于活跃。

我国家电市场目前仍存在高竞争性、高集中度、规模经济明显、技术壁垒越来越高、产品创新能力较差等特征。经历了过去 30 年的高速发展，今后，以技术、质量、服务为代表的非价格竞争方式将成为众多家电企业在竞争中取胜的新的突破口。

1.1.2 智能家电产业发展分析

智能家电就是将微处理器、传感器技术、网络通信技术引入家电设备后形成的家电产品，具有自动感知住宅空间状态和家电自身状态、家电服务状态，能够自动控制及接收住宅用户在住宅内或远程的控制指令；同时，智能家电作为智能家居的组成部分，能够与住宅内其它家电和家居、设施互联组成系统，实现智能家电功能。

智能家电与传统家电的不同，在于智能家电实现了拟人智能，产品通过传感器和控制芯片来捕捉和处理信息，除了根据住宅空间环境和用户需求自动设

置和控制，用户可以根据自身的习惯进行个性化设置，另外，当智能家电与互联网连接后，其也就具备了社交网络的属性。

中国家电行业自 2013 年开始进入智能化阶段，但到目前为止我国智能家电市场的消费观念仍不成熟，市场规模同整个家电行业相比还相对较小，2016 年我国智能家电行业销售收入仅为 89.07 亿元，同比增长 77.90%，2015 年则增长 127.59%。但是随着智能家居市场推广的进一步落实，消费者对智能家居有更深层次的了解。一旦培育出消费者使用智能家电的习惯，那么智能家电市场将会急剧扩张。预计 2017-2022 年我国智能家电销售收入将快速增长，至 2022 年我国智能家电销售收入将突破千亿，达 1015 亿元，同比 2016 年增长 10 倍。

智能家电迅猛发展将成为必然趋势，主要原因如下：

- (1) 网络技术和通信技术的成熟和广泛应用；
- (2) 信息化水平的不断提高，逐渐达到支持智能家电产业大规模发展的水平；
- (3) 互联网基础设施和技术条件，为智能家电的发展做了必要的准备；
- (4) 用户对高水平家电的需求。

同传统家电产品相比，智能家电具有如下特点：

- **网络化：**各种智能家电可以通过家庭网络、电信网或广电网连接到一起，可以通过网络与制造商的服务平台相连，实现信息共享。
- **智能化：**智能家电可以根据周围环境的不同自动做出响应，不需要人为干预。例如智能空调可以根据不同的季节、气候及用户所在地域，自动调整工作状态以达到最佳效果。
- **开放性和兼容性：**由于用户家庭的智能家电可能来自不同的厂商，智能家电服务平台必须具有开放性和兼容性。
- **节能化：**智能家电可以根据周围环境自动调整工作时间、工作状态，从而实现节能。
- **易用性：**由于复杂的控制操作流程已由内嵌在智能家电中的控制器或网络服务平台解决，因此用户只需了解非常简单的操作。

1.1.3 智能家电的发展瓶颈

(1) 缺乏统一的行业或者国家标准

智能家电发展中遇到的最突出问题就是缺乏统一的行业或者国家标准，尤其是通信接口标准，导致家电企业开发的智能家电系统较为封闭，无法获得社会化的支持。生产厂商按照各自企业规范生产的智能产品适用面窄，与其他品牌互不兼容性、无法实现家电之间的联动，难以形成整体的解决方案。

(2) 功能华而不实，产品可靠性低

整合了多种功能的智能产品其实并没有真正与用户需求相契合，不实用、甚至是不必要的功能反而降低了消费者对于智能产品的消费和使用欲望；同时，华而不实的功能、繁琐的操作也在不停的考验着本就不高的系统稳定性。

(3) 投入过高，令企业望而却步

智能家电开发新增了互联网接入软硬件开发、服务器后台开发、智能终端应用开发等，开发成本相对比较高。智能产品和市场的成熟又不可能一蹴而就，企业的投入短期内无法得到回报。

(4) 安全性得不到保障

智能家电会与网络相连，将带来网络安全、信息安全、以及电器安全的风险。没有用户愿意将自己暴露在随时会被攻破隐私的危险环境下，尤其是在黑客技术无处不在的今天，这种担忧阻碍着消费者对智能家电的选择。

(5) 产品售后与服务体系不完善

当智能家电被整合进越来越多的功能，虽然会满足消费者更加多变的使用需求，但同时增加了维修服务的难度。智能家电的普及，必须伴随着更加专业的售后服务团队来提供技术保障。

1.1.4 WiFi 应用痛点

网络通信是家电智能化应用的基础。在通信方式上，家电企业做过很多尝试，例如有线通信方式有电力线载波、BACnet、CEBUS、RS485、LonWorks、HomePNA 等，无线通信方式有 Bluetooth、ZigBee、HomeRF、3G、4G 以及今

天普及率极高的 WiFi。但这些通信方式在功耗、信息安全、硬件成本、覆盖范围等方面并未实现相对完善的解决方案。

目前我国大部分住宅建设尚未考虑家庭联网的需要，基础设施严重不足，重新布线费时费力，影响推广。采用无线通信方式相对而言比较灵活，更适合家庭。国内的智能家电产品大都采用 WiFi 方式接入家庭网络。相比于 ZigBee 和 Bluetooth 技术，WiFi 技术有着更容易接入到互联网中、芯片模组产业成熟、更便于快速部署到相关产品中的优势，但同时，WiFi 也存在这相当多的无法忽视的痛点问题。

（1）接入问题

目前已经有大量安装有 WiFi 模块的家电在市场上销售，但是每台售出的家电，都需要安装人员或消费者对 WiFi 接入进行设置，由于每家 WiFi 名称不统一，接入配置无法统一化完成，造成人力安装成本无形上升。

（2）在线问题

带有 WiFi 模块的家电经过安装后，会出现经过一段时间掉线无法重联的情况，或者由于路由重启、重置等其他问题造成家电再也无法连接到网络上。

（3）信号覆盖问题

WiFi 基于高频共有频段，对于房间角落、家居密集摆放区等对高频无线传输影响较大的地方，会出现无法获取信号的情况，导致很多位于该方位的家电无法连接到网络。

（4）安全问题

WiFi 采用共有频段进行数据传输，这种非授权频谱的自组网技术，其抗干扰性、数据管理水平良莠不齐，造成数据在传输过程中被破解、盗用的风险极高。

（5）功耗问题

由于 WiFi 技术的功耗较高，对于越来越多的对功耗要求严格的节能环保型家电，WiFi 模块将不能满足其需求。

为满足物联网发展需求，NB-IoT（Narrow Band Internet of Things，窄带物联网）通信随之产生，凭借大连接、低功耗、低成本、广覆盖、高安全的特点，

越来越多的被应用于物联网行业。在家用电器行业，今年上半年，海尔、美的、长虹、小天鹅、荣事达等家电企业分别与电信运营商在 NB-IoT 领域签署合作协议，达成战略合作。

1.2 NB-IoT 技术背景

1.2.1 NB-IoT 技术简介

NB-IoT 是一种基于蜂窝通信 3G/4G 演进的物联网通信技术，全球移动通信标准组织 3GPP 负责 NB-IoT 技术的标准化，首个标准版本 Release13 已经于 2016 年 6 月发布，并持续演进。NB-IoT 工作于专用授权频段，主要应用于低吞吐量、海量连接的场景，未来将承接大量物联网业务接入。

NB-IoT 技术的主要特点如下：

- **大容量：**在同一基站的情况下，与现有无线技术相比 NB-IoT 可提供 50~100 倍的连接数（具体连接容量根据实际业务模型评估）。
- **高覆盖：**在同样的频段下，NB-IoT 比现有蜂窝网络提升 20dB 增益，将极大的改进室内覆盖。
- **低功耗：**NB-IoT 借助 PSM (Power Saving Mode, 节电模式)和 eDRX (Extended Discontinuous Reception, 扩展型非连续接收)可实现更长的待机，设备续航时间可以大幅提升到几年，甚至 10 年以上。NB-IoT 芯片正常待机耗电只需毫瓦 (mW) 级，而在 PSM 模式下通信芯片功耗更只需微瓦 (μ W) 级。
- **低成本：**低速率、低带宽等带来的简化设计同样给 NB-IoT 芯片以及模块带来低成本优势，通信模组成本可以达到\$5 以下。相比其它通信技术，NB-IoT 可以使得大量物联网设备直接通过成熟稳定可靠的运营商网络接入，消费者、家电厂商无需自组网，也有望降低运营和维护成本。
- **高安全：**NB-IoT 基于专用频谱、空口双向鉴权及严格加密，提供电信级别的安全可靠能力，保障用户数据的传输安全性。

正因为 NB-IoT 具备的低功耗、广覆盖、低成本、大容量等优势，使其可以广泛应用于多种垂直行业，如远程抄表、资产跟踪、智能停车、智慧农业、智能安防、共享单车、智能家居等领域。

1.2.2 NB-IoT 与其他无线通信技术的对比

物联网通信从传输距离上可分为低功耗广域网通信和局域网通信（见表 1）。针对于智能家电不同的应用场景，不同类型的通信技术各有优势。相对于局域网通信技术，低功耗广域网技术代表着新的一类趋势。

表 1 几种常用的无线通信技术对比

指标 通信方式	类型	通信距离	理论速度	优点	缺点
NB-IoT	低功耗广域网	2.5-5Km	200Kbps	广覆盖 大容量 低功耗 低成本 高安全性	低速率
eMTC		2.5-5Km	1Mbps	广覆盖 移动性强 应用多元化 高安全性	速率仍低于 WiFi
Sigfox		30-50Km	1Kbps	极广覆盖 极大容量 极低功耗	极低速率，国内组网困难
Lora		10-15Km	180-37.5Kbps	广覆盖 大容量 低功耗 低成本	低速率，国内组网困难

指标 通信方式	类型	通信距离	理论速度	优点	缺点
WiFi	无线局域网	-100m	54-300Mbps	速度快 普及率高	移动能力弱 连接数量少 接入配置复杂 高功耗 安全性差
Bluetooth		-10m	100Kbps	低功耗 不依赖外部网络	组网能力弱 短距离 高成本
ZigBee		-30m	10Kbps	低功耗 组网能力强	短距离 极低速率 普及率低 高成本 易受干扰
NFC		-1m	100Kbps	高安全性	极短距离 低速率 普及率低

广域网通信技术一般定义为 LPWA (Low Power Wide Area, 低功耗广域技术), 是一种大跨越地域性的网络, 具有分布域广、可移动性、广接入数大等特性。

广域网通信技术可以分为如下两类:

(1) 工作在非授权频谱的技术: 如 LoRa、SIGFOX, 具有低速率、低功耗、低成本等特性; 这类技术大多是非标准的, 自定义实现, 因此抗扰性、安全性和接入数量受自建网络的性能的影响较大。

(2) 工作在授权频谱的技术: 如我们所熟悉的移动通信网络 GSM、LTE 等蜂窝通信技术, 以及 NB-IoT 技术及 eMTC 技术; 由于工作在授权频谱, 有运营商运营和维护, 因此具有网络可靠性高、安全性高、信号覆盖广的优势;

LTE 及其演进是对应于高速大流量、低时延数据应用；NB-IoT 技术对应于低速率、低功耗、低成本、网络信号穿透性强、海量连接数、时延不敏感（秒级）的数据应用；eMTC 技术则是介于 NB-IoT 技术与 LTE 技术两者之间，属于中速应用。

局域网通信技术，如 WiFi、蓝牙、ZigBee 等，是应用在某一个区域内的网络，具有传输速度比较高、传输距离短、功耗不低的特性，适用于数据的短距离、高速传输，不适用于广泛分布、大量连接、低功耗的场景。

NB-IoT 技术与其他技术相比，在低速率、广分布、低成本应用的场合具有明显的技术优势（如表 2 所示）。

表 2 广域网通信技术对标

通信技术 对标项目	NB-IoT	eMTC	Sigfox	Lora
标准	国际标准	国际标准	私有标准	私有标准
频谱	授权	授权	非授权	非授权
峰值速率	UL:204.8kbps DL:234.7kbps	UL:1Mbps DL:800Kbps	UL:100bps DL:600bps	180bps-37.5kbps
最大发射功率	23dBm	23dBm	14dBm	14dBm
安全性	高	高	低	低
通信可靠性	高	高	低	低

1.2.3 NB-IoT 应用于家电行业

在智能家电领域，NB-IoT 可以快速、安全、低成本的实现对智能家电的信息采集、状态监测和控制指令下发等远程操作，提高智能家电的网络接入率，并实现更有针对性的动态管理，从而为消费者提供更加智能的场景应用服务，也有望让家电厂商提供更智能、更高效的产品售前、售中、售后服务。

NB-IoT 对智能家电领域应用的利好，具体体现在以下 4 个方面。

(1) 接入方面

NB-IoT 是一种由电信运营商提供统一网络接入的技术，是全球标准通信技术，支持全球漫游使用。每台安装 NB-IoT 通信模组的家电，在售出时只需经用户授权开通，就可以直接经电信运营商网络接入家电厂家自有或其它第三方的业务云。同时，NB-IoT 大容量能力支持海量设备接入，避免类似由于家庭路由器接入数量限制而导致智能家电无法接入问题。因此，NB-IoT 使得智能家电接入配置统一化成为可能，简化安装接入配置流程，可有效降低人力安装成本。

(2) 信号方面

NB-IoT 网络是由专业电信网络运营商建设并提供服务的物联网专用通信网络，使用有别于 WiFi 共用频段的专用授权频段；同时，NB-IoT 技术本身具备更高覆盖能力，相比现有移动通信网络增加 20dB 覆盖，可提供最佳的室内覆盖。可解决家电安装分散、安装位置条件复杂等网络覆盖问题，更易实现接入并保持长久在线。

NB-IoT 是基于蜂窝移动通信网络的广域覆盖技术，如果家电配备电池供电，则可在售前物流、家电回收等环节实现智能家电的状态和位置跟踪，进而可协助厂商开展更高效便捷的供应链管理和家电回收管理。

(3) 安全方面

NB-IoT 的电信网络是基于授权频谱组建、具有双向通信鉴权和加密能力的网络，其抗干扰能力、数据安全性、技术服务等方面有高安全保障。NB-IoT 可协助确保家电合法接入、并保障数据传输的完整与安全性。

(4) 能耗方面

NB-IoT 正常功耗低于 WiFi，同时提供多种网络在线模式——类似手机的待机模式（DRX 模式）、省电待机（eDRX 模式）、节电模式（PSM 模式），智能家电可依据自身功耗需求选择和转换在线模式以满足智能家电未来越来越严苛的节能、环保要求。

1.2.4 NB-IoT 产业进展

(1) NB-IoT 全球化商用标准已发布

NB-IoT 首个版本 Release13 于 2016 年 6 月冻结。3GPP 国际标准组织以市场业务应用为导向，在兼容现有功能的基础上对协议功能进行持续增强，2017 年 Release14 标准也已冻结发布，其引入广播功能（适用于批量终端设备以广播方式同时进行软件升级）、并对性能做了增强，包括更高吞吐率（达到 100Kbps 以上）、更低功耗和更低时延等。

（2）NB-IoT 芯片和模组已经成熟

当前华为、高通、RDA、MTK、Nordic 等均已宣布其 NB-IoT 芯片计划，其中起步最早的华为已于 2017 年一季度推出了商用芯片，并于 4 月份开始支持大规模发货。在模组领域，移远、利尔达、u-blox、Telit、Gemalto、SIERRA 等模组厂商都可提供成熟的 NB-IoT 商用模组。

（3）国际大厂引领网络发展

NB-IoT 技术的最大优势是众多国际巨头厂商的投入和支持。华为、爱立信、诺基亚等通信设备厂商均已推出了支持 NB-IoT 的软件版本，通过软件升级即可在 2G/3G/4G 网络基础上快速开通 NB-IoT。

（4）NB-IoT 大网正在全球快速部署

2017 年全球已将部署 39 张 NB-IoT 商用网络，覆盖 28 个国家。目前中国、韩国、德国、荷兰、西班牙等国已经启动了商用部署，其中中国电信 2017 年上半年全网商用 NB-IoT，中国联通、中国移动也将在 18 年中实现全网商用。

（5）中国政府大力支持 NB-IoT 产业的发展

工业和信息化部今年 6 月发布了《关于全面推进移动物联网（NB-IoT）建设发展的通知》，指出：基础电信企业要加大 NB-IoT 网络部署力度，提供良好的网络覆盖和服务质量，全面增强 NB-IoT 接入支撑能力；到 2017 年末，实现 NB-IoT 网络覆盖直辖市、省会城市等主要城市，基站规模达到 40 万个；到 2020 年，NB-IoT 网络实现全国普遍覆盖，面向室内、交通路网、地下管网等应用场景实现深度覆盖，基站规模达到 150 万个；推动 NB-IoT 在个人生活领域的应用，促进信息消费发展，加快 NB-IoT 技术在智能家居、可穿戴设备、儿童及老人照看、宠物追踪及消费电子等产品中的应用，加强商业模式创新，增强消费类 NB-IoT 产品供给能力，服务人民多彩生活，促进信息消费。

1.3 基于 NB-IoT 的智能家电解决方案架构

基于 NB-IoT 的智能家电解决方案按照云管端的系统架构来建设，以满足未来的演进需求，方案包括终端、网络、IoT 网关和智能家电应用 4 个部分。通过物联网、云计算、大数据等技术将各个层面整合统一为有机的整体，支撑智能家电业务的构建和快速上线。基于 NB-IoT 的智能家电网络架构如图 1 所示。

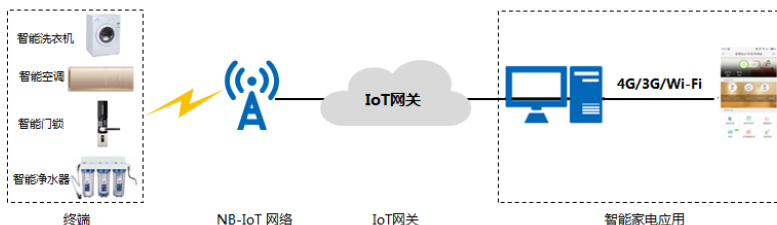


图 1 基于 NB-IoT 的智能家电网络架构

1.3.1 终端

终端主要包括各种智能家电产品，如智能空调器、智能电冰箱、智能洗衣机、智能小家电等。随着物联网的发展，家电由原有的哑终端向智能终端演进，通过增加各种传感器、通信模块使得家电变得可控、可管、可互通。智能家电集成 NB-IoT 通信模组后，就具有与 NB-IoT 基站连接通信的能力。智能家电通过 NB-IoT 基站实现与应用服务平台的数据信息双向交互。

根据目前已有的芯片和终端情况，采用海思芯片的 NB-IoT 通信模组支持 UDP 协议传输，同时支持基于 UDP 的 CoAP 协议的传输，而且使用海思模组的终端厂商也有自行开发基于 UDP 的 MQTT-SN 协议；采用高通芯片的 NB-IoT 通信模组，支持 UDP 协议和 TCP/IP 协议传输，因此支持基于 UDP 的 CoAP 协议以及基于 TCP/IP 的 MQTT 和 HTTP 协议的传输。

1.3.2 NB-IoT 网络

网络是整个物联网的通信基础，NB-IoT 网络具有低速率、低成本、低功耗、广覆盖的特点，符合当前物联网通信的需求。NB-IoT 使用较窄的频段实现广覆盖，大连接的能力，相比传统的 LTE 网络，成本会大大降低，与其他的 LPWA 技术相比，NB-IoT 网络基于授权频谱组建网络，其抗干扰能力、数据安全性、技术服务等方面均有高安全性保障。

1.3.3 IoT 网关

IoT 网关提供 NB-IoT 通信业务的连接管理、设备管理、数据管理、能力开放等基础功能，支持多种物联网终端设备的快速接入，支撑客户聚集应用和业务创新。

IoT 网关通过统一的接口实现不同终端的接入，提供连接感知、连接诊断、连接控制等连接状态查询和管理功能，智能家电应用（/平台）无需再关心智能家电具体的物理连接和数据传输，实现终端对象化管理。IoT 网关还提供灵活高效的数据管理，包括数据采集、分类、结构化存储、数据调用、使用量分析，提供分析性的业务定制报表，业务模块化设计，业务逻辑可实现灵活编排，满足智能家电应用的快速开发需求。

1.3.4 智能家电应用

智能家电应用提供端到端，即设备端、云端以及移动端的整体解决方案。设备端的嵌入式代理软件包含完全优化的网络协议栈，运行于联网设备或网关中，能够让设备安全可靠地连接到云端；云端包括一整套物联网运营支持服务，具备海量设备的连接管理能力，以及大数据处理分析能力；移动端的移动应用开发库包含了丰富的 API 接口，帮助客户快速高效的开发出安全可靠的移动应用。

【云服务】

接入服务：接入服务面向设备与手机应用，建立机器与机器、机器与用户的连接与数据实时通信能力。

企业 API: 企业 API 服务提供了 M2M 接入服务基础能力之上的业务接入与管理服务能力, 为设备与应用提供了 REST API 的服务访问能力, 包括设备管理、用户管理、分享、定时调度、数据查询、安全管理等功能。

消息处理引擎: 消息处理引擎实时处理各类消息, 提供消息的接收、解析、分发、存储的功能。

数据处理: 采集设备、应用数据, 并对设备数据、用户数据实时计算, 输出计算结果。

管理界面: 为企业的 IoT 管理提供可视化管理、运营支撑工具。

【端接入】

嵌入式固件: 提供硬件设备与云服务、手机应用实时通信的能力。解决设备连网、上报数据、接收指令等功能,

开发 SDK: 开发 SDK 能让企业快速完成 APP、微信应用的开发, 无需关注底层通讯与通用业务实现。

1.4 基于 NB-IoT 的智能家电应用进展

(1) 基于 NB-IoT 的智能空调器

智能空调依托 NB-IoT 网络广覆盖、海量连接、成本低等优势, 联手进行新一代智能空调器的研发、推广, 以实现远程控制、定位、预测性维护等功能。

智能空调器在硬件上增加 NB-IoT 通信模组, 通过 NB-IoT 网络和 IoT 网关接入到后端智慧云平台, 并通过智慧云平台实现对空调器的远程控制和状态采集。

该方案在 2017 年上海家电博览会上进行演示, 为消费者提供五大功能: 开关机、切换空调模式、调整温度、调整转速、温度数据上报。

(2) 基于 NB-IoT 的智能洗衣机

自助洗衣平台以微信公众号作为用户操作平台, 实现远程机位查询、提前预约、在线支付、完工提醒等服务。NB-IoT 通信模块安装在商用智能洗衣机的控制板上, 通过 NB-IoT 网络可以实现智能洗衣机与自助洗衣平台间的双向数

据交互，解决了高校洗衣房无无线网络传输的难题，能够以最快最准确的方式为用户找到距离最近的洗衣房。

(3) 基于 NB-IoT 的智能门锁

门锁使用了 NB-IoT 技术，具备“不需外接网关，永久连接，更长的待机时间，长达 10 年的电池使用寿命”等优势，并与其他智能家居产品实现了互联互通、场景联动。用户通过指纹、密码、门卡、钥匙等方式开锁时，用户的手机 App 可以收到开锁信息。

(4) 基于 NB-IoT 的智能小家电

目前很多智能小家电也开始采用 NB-IoT 技术构建自己的优势，如智能豆浆机通过 NB-IoT 网络将使用信息上报到业务端，通知厂家更换豆浆包；智能咖啡机通过 NB-IoT 网络，自动告知店家原材料的使用情况，及时备货；智能空气净化器及时通知商家配件使用快到期，帮助顾客自动下单，方便快捷。NB-IoT 技术将会为越来越多的小家电插上“智能的翅膀”，助力家电企业越飞越高。

第二册

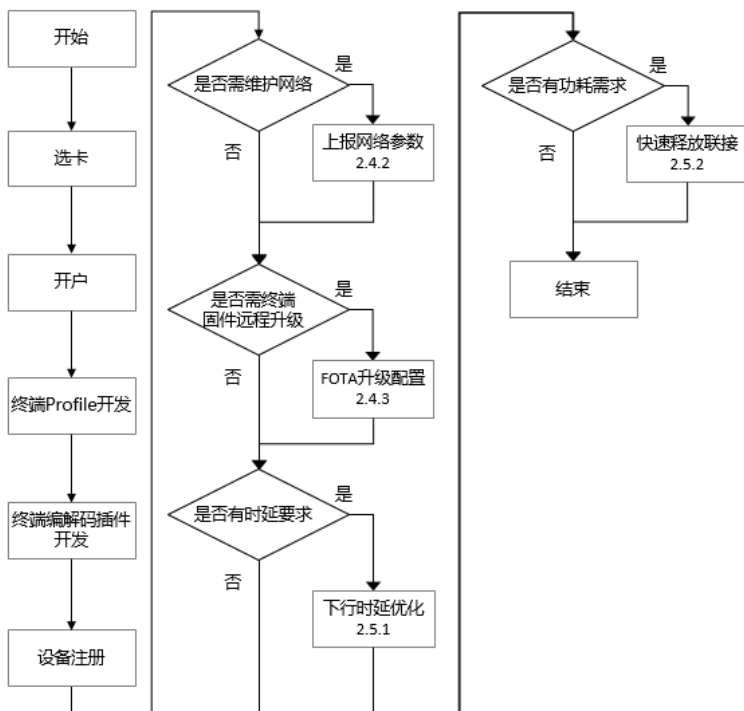
智能家电 NB-IOT 技术开发指导

2 基于 NB-IoT 的智能家电开发指导

2.1 NB-IoT 智能家电开发集成流程介绍

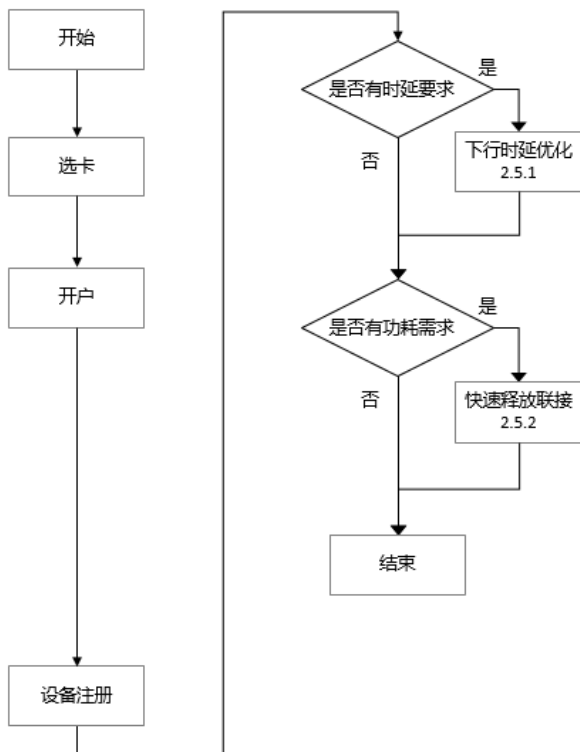
2.1.1 智能家电与 IoT 网关对接开发流程

智能家电与 IoT 网关对接开发流程包含：选卡阶段、注册阶段、IoT 网关对接阶段、IoT 网关与应用平台对接阶段，终端与网络适配阶段等；



2.1.2 智能家电与 IoT 网络对接开发流程

智能家电与 IoT 网络对接开发流程包含：选卡阶段、注册阶段、终端与网络适配阶段等；



2.2 NB-IoT 智能家电物联网卡

2.2.1 NB-IoT 物联网卡介绍

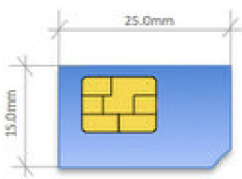
表 3 中国电信主要提供下表几种物联网卡产品：

产品名称	封装类型	型号	规格
消费级物联网卡	插拔卡	2FF(SIM 卡)	25mm*15mm

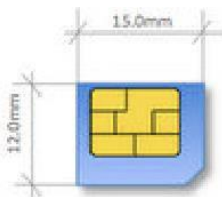
产品名称	封装类型	型号	规格
工业级物联网卡		3FF(Micro-SIM 卡)	15mm*12mm
		4FF(Nano-SIM 卡)	12.3mm*8.8mm
	贴片卡	MFF2	5mm*6mm
	插拔卡	插拔卡	2FF(SIM 卡)
3FF(Micro-SIM 卡)			15mm*12mm
4FF(Nano-SIM 卡)			12.3mm*8.8mm
	贴片卡	MFF2	5mm*6mm

2.2.1.1 插拔式物联网卡

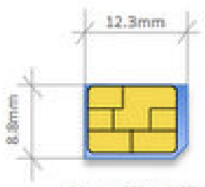
插拔卡包括 2FF、3FF 与 4FF，格式与布局符合 ISO7816—1 标准。



2FF 尺寸：25×15mm

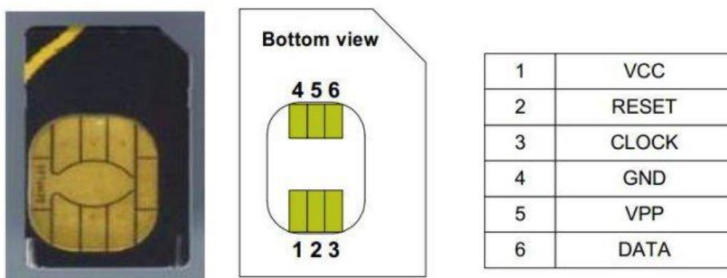


3FF 尺寸：15×12mm



4FF 尺寸：12.3×8.8mm

常用的 2FF/3FF/4FF 卡为 6 脚，管脚定义如下图。



2.2.1.2 贴片式物联网卡

贴片卡采用 VQFN8 封装，市面主流的 MF2 尺寸布局如下。

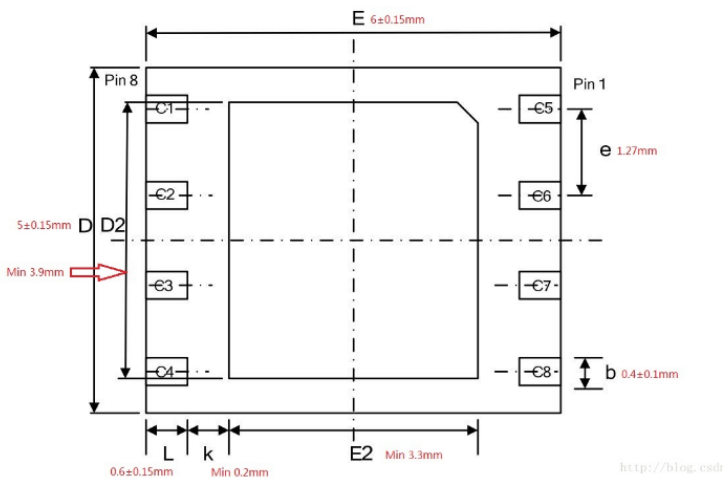


表 4 MF2 结构参数

参数	描述	尺寸(mm)
E	封装水平方向长度.	$6,00 \pm 0,15$
D	封装垂直方向长度..	$5,00 \pm 0,15$
L	封装外部管脚水平方向长度.	$0,60 \pm 0,15$
B	封装外部金属管脚的最小垂直距离	$0,40 \pm 0,10$

E2	封装内部散热片水平长度.	min 3,30
D2	封装内部散热片垂直长度.	min 3,90
K	封装外部管脚与内部散热片最近的距离	min 0,20
E	封装外部金属管脚的中心线到相邻金属管脚中心线的距离	1,27 公差见 bbb 和 ddd 中描述
bbb	中心线公差	0,10
ddd	触点间距离公差电	0,05

2.2.1.3 插拔式物联网卡与贴片式物联网卡对比分析

所有类型物联网卡电气特性与传输协议相同，区别点在于物理特性，适用于消费级与工业级应用场景。消费级插拔卡、消费级贴片卡、工业级插拔卡与工业级贴片卡物理特性如下。

表 5 消费级插拔卡参数表

技术参数描述	消费级插拔卡
工作和存储温度	T1 -25 to +85°C
湿度	H1 在 50 度温度，相对湿度范围 90%~95%，1000 小时的条件下，可以保证卡的操作和存储正常。
数据保留时间	R1 10 年
擦写次数	E1 10 万次
震动	V1 5Hz to 500Hz
防静电	S1 在卡暴露在 2000V 的静电环境中，不应降低卡片功能。
附着力	A1 对卡施加 60N 的拉力并持续 1 分钟，模块不应从卡上分离，卡的操作和存储正常
抗电磁干扰	M1 在卡暴露在稳定的 79500A/m (1000Qe) 磁场下，不应降低卡片功能
腐蚀	C1 在 5% 盐度(NaCl)的环境下存储至少 24 小

智能家电产业 NB-IoT 应用及开发指导书

	时, 卡的操作和存储正常
抗 X 射线	X1 在卡的任何一面每边在受到 0.1Gy 剂量, 相当于 70~140KeV 中等能量 X 射线照射时 (一年的累计剂量), 不应降低卡片功能
触点与卡基表面偏差	触点最高点不能高于卡表面 0.05mm、最低点不能低于卡表面 0.1mm
抗紫外线	卡接受总能量为 15Ws/cm ² 的光照射后, 卡功能应正常
机械强度	施加 10N 的力, 应保持其功能完好
翘曲	从水平刚性平台到卡任何表面凸起部分应不大于 1.0mm
触点位置和尺寸	应符合 GB/T 16649-2 要求

表 6 消费级贴片卡参数表

技术参数描述	消费级贴片卡
工作和存储温度	T1-25 to +85°C
湿度	H1 在 50 度温度, 相对湿度范围 90%~95%, 1000 小时的条件下, 可以保证卡的操作和存储正常。
湿度/回流焊	MX1 符合国际技术规范 IPC/JEDEC J-STD-020 中的规定: 温度 260°C (Tc) 支持无铅工艺;湿度敏感等级 3;无铅装配回流标准曲线级别。
数据保留时间	R1 10 年
擦写次数	E1 10 万次
震动	V1 5Hz to 500Hz
抗静电	S1 在卡暴露在 2000V 的静电环境中, 不应

智能家电产业 NB-IoT 应用及开发指导书

	降低卡片功能。
抗电磁干扰	M1 在卡暴露在稳定的 79500A/m (1000Qe) 磁场下, 不应降低卡片功能
腐蚀	C1 在 5% 盐度(NaCl)的环境下存储至少 24 小时, 卡的操作和存储正常
抗 X 射线	X1 在卡的任何一面每边在受到 0.1Gy 剂量, 相当于 70~140KeV 中等能量 X 射线照射时 (一年的累计剂量), 不应降低卡片功能
抗紫外线	卡接受总能量为 15Ws/cm ² 的光照射后, 卡功能应正常

表 7 工业级插拔卡参数表

技术参数描述	工业级插拔卡
工作和存储温度	T3 -40 to +105°C
湿度	H2 在 85 度温度, 相对湿度范围 90%~95%, 1000 小时的条件下, 可以保证卡的操作和存储正常。
数据保留时间	R2 10 年
擦写次数	E2 50 万次
震动	V1 5Hz to 500Hz
抗静电	S2 在卡暴露在 4000V 的静电环境中, 不应降低卡片功能。
附着力	A1 对卡施加 60N 的拉力并持续 1 分钟, 模块不应从卡上分离, 卡的操作和存储正常
抗电磁干扰	M1 在卡暴露在稳定的 79500A/m (1000Qe) 磁场下, 不应降低卡片功能
腐蚀	C1 在 5% 盐度(NaCl)的环境下存储至少 24

智能家电产业 NB-IoT 应用及开发指导书

	小时，卡的操作和存储正常
抗 X 射线	X1 在卡的任何一面每边在受到 0.1Gy 剂量，相当于 70~140KeV 中等能量 X 射线照射时（一年的累计剂量），不应降低卡片功能
触点与卡基表面偏差	触点最高点不能高于卡表面 0.05mm、最低点不能低于卡表面 0.1mm
抗紫外线	卡接受总能量为 15Ws/cm ² 的光照射后，卡功能应正常
机械强度	施加 10N 的力，应保持其功能完好
翘曲	从水平刚性平台到卡任何表面凸起部分应不大于 1.0mm
触点位置和尺寸	应符合 GB/T 16649-2 要求

表 8 工业级贴片卡参数表

技术参数描述	工业级贴片卡
工作和存储温度	T3 -40 to +105°C 或 T4-40 to +125°C
湿度	H2 在 85 度温度，相对湿度范围 90%~95%，1000 小时的条件下，可以保证卡的操作和存储正常。
湿度/回流焊	MX1 符合国际技术规范 IPC/JEDEC J-STD-020 中的规定：温度 260°C (Tc) 支持无铅工艺;湿度敏感等级 3;无铅装配回流标准曲线级别。

数据保留时间	R1 10 年或 R2 15 年
擦写次数	E2 50 万次
震动	V1 5Hz to 500Hz
抗静电	S2 在卡暴露在 4000V 的静电环境中，不应降低卡片功能。
抗电磁干扰	M1 在卡暴露在稳定的 79500A/m (1000Qe) 磁场下，不应降低卡片功能
腐蚀	C1 在 5% 盐度(NaCl)的环境下存储至少 24 小时，卡的操作和存储正常
抗 X 射线	X1 在卡的任何一面每边在受到 0.1Gy 剂量，相当于 70~140KeV 中等能量 X 射线照射时（一年的累计剂量），不应降低卡片功能
抗紫外线	卡接受总能量为 15Ws/cm2 的光照射后，卡功能应正常

2.2.2 NB-IoT 智能家电物联网卡选择建议

由于家电行业产品的使用环境相对于比较复杂，同时对于产品有不同的使用要求，对于使用环境良好，设备运行较平稳的智能家电设备可以采用成本较低的插卡式物联网卡；对于使用环境较为恶劣，同时设备运营状态震动大等，建议使用消费级贴片卡或工业级贴片卡。

2.3 NB-IoT 智能家电开发资源申请

2.3.1 NB-IoT 物联网卡申购

客户联系所在省、地市电信公司，并提出申请，会有物联网专属人员提供相关服务。

2.3.2 IoT 网关测试账号申请

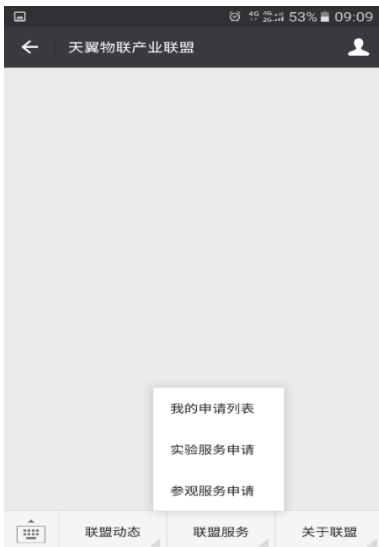
要申请测试的登录“天翼物联产业联盟”微信公众号填写 IoT 网关对接测试（即，平台对接服务）账号申请。

1、查找微信公众号并进入



3、完成表格填写并提交

2、点击“联盟服务”→“平台对接认证”



4、账号申请通过后，收到账号和初始密码

智能家电产业 NB-IoT 应用及开发指导书

返回 平台对接认证

中国电信物联网开放实验室
平台对接认证

初次申请平台对接认证的客户会分配平台测试账号

申请单位

单位名称 *

联络人 *

邮箱 *

电话 *

服务需求说明

物联网应用简述 *

【中国电信物联网开放平台（对接测试）】开户密码通知

中国电信物联网开放平台

发件人: 中国电信物联网开放平台 <ctiotcmp@189.cn>

大小: 14 KB

USER_INITPWD_shangyanyu... (7 KB)

贵公司在中国电信物联网开放平台申请的账号已开通, 请登录中国天翼联盟网站: www.tianyiliot.com; 进入联盟服务频道的下载请登录中国电信物联网开放平台企业门户: <https://180.101.147.135:8843>, 该用的管理操作

请登录中国电信物联网开放平台开发者门户: <https://180.101.147.208:8093>, 插件的在线编辑和上传功能

测试平台应用对接地址:
 中国电信物联网开放平台北向API-企业应用接入地址: <https://180.101.147.89>
 测试平台设备对接地址:
 中国电信物联网开放平台南向-终端设备接入地址: 180.101.147.115:5683 (说且需模组芯片支持)

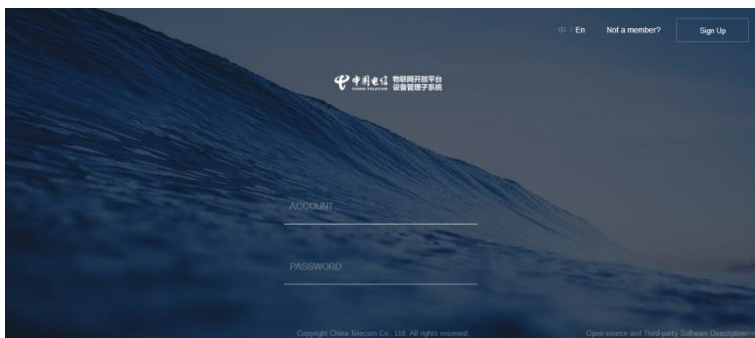
2.4 智能家电与 IoT 网关对接开发

2.4.1 IoT 网关对接

2.4.1.1 创建应用

登录中国电信物联网开放平台企业门户（即，IoT 网关入口）（<https://180.101.147.135:8843/main.html#/login>）。

图2-1 中国电信物联网开放平台企业门户



选择【应用管理】->【应用】点击右上角“+创建应用”。设置应用参数，所属行业选择公用事业(NB-IoT)，关联 API 包一栏选择基础 API 包和公用事业 (NB-IoT) API 包，其他参数默认即可。

图2-2 创建应用



创建 APP 成功后，得到 appId 和 secret，请保存。

2.4.1.2 NB-IoT 参数配置

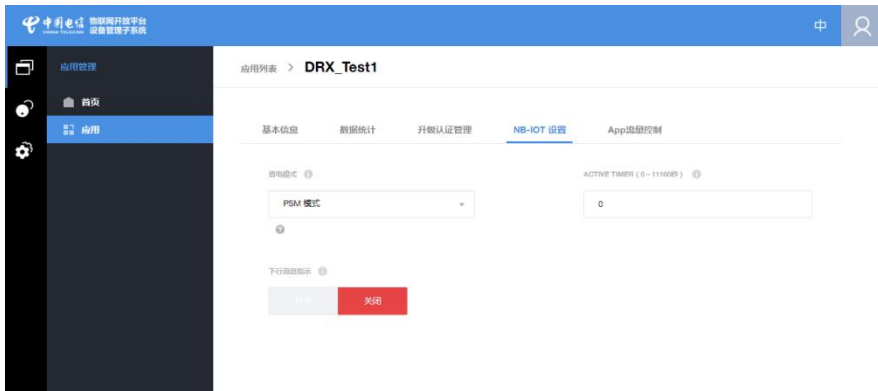
点击已经创建好的应用，并对 NB-IoT 参数进行设置。

图2-3 已创建应用



目前平台支持设置三种模式，分别为 DRX 模式(Discontinuous Reception，不连续接收模式)，EDRX 模式(Extended Idle Mode DRX，扩展不连续接收模式)及 PSM 模式(Power Saving Mode，省电模式)。如果 NB-IoT 设置没配置(即“自定义省电模式配置参数”未勾选)，则默认设备的模式认为 PSM 模式。若启用了“自定义省电模式配置参数”，则用户根据 ACTIVE TIMER 和 eDRX 周期两个参数设置设备的模式。

图2-4 NB-IoT 设置



和物联网卡的 APN 设置保持一致；若设置了 ACTIVE TIMER 参数则表示启用 PSM 模式；若仅设置了 eDRX 周期则表示启用 EDRX 模式；若两个参数都不设置则表示启用 DRX 模式。

说明

这里两个参数的设置要与网络侧保持一致，参数的设置只是配置到 IoT 网关中，不会配置到网络或者设备。对于下行网络时延有要求的，请使用 DRX 模式，请勿配置 eDRX 及 PSM 参数。

2.4.1.3 定义终端 Profile

设备的 Profile 文件是用来描述一款设备是什么、能做什么以及如何控制该设备的文件。每款设备都需要一个 Profile 文件。

一个 Profile 文件的内容主要包括两部分，一部分是设备的基础属性信息，如 manufacturerId（厂商 ID）、manufacturerName（厂商名字）、deviceType（设备类型）、model（设备型号）、protocolType（协议类型）等；另一部分是设备的服务信息，这一部分定义了设备的业务数据，包括设备上报的上行数据和厂商服务器下发给设备的下行数据。

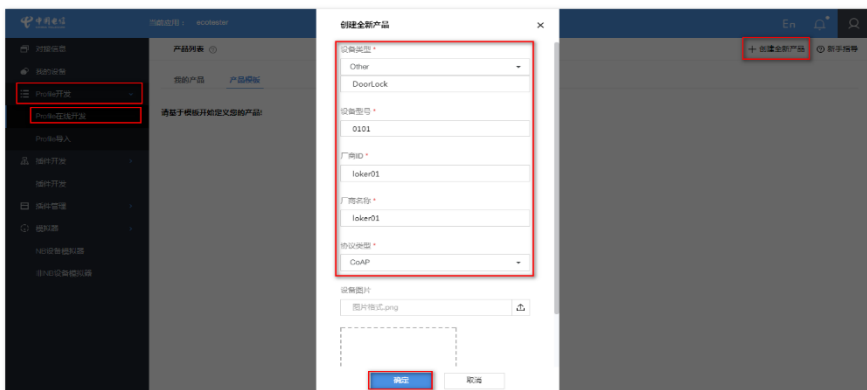
Profile 定义设备模型有两种方式，一种方式参考《中国电信 IoT_设备能力描述文件 profile 开发指南.doc》离线制作设备 Profile 插件，另一种方式为通过中国电信物联网开放平台企业门户制作 Profile 文件。我们建议终端用户通过中国电信物联网开放平台企业门户来制作 Profile 文件。

离线 Profile 文件开发

离线 Profile 文件开发请参考[《中国电信物联网开放平台_设备能力描述文件 profile 开发指南》](#)。

在线 Profile 文件开发

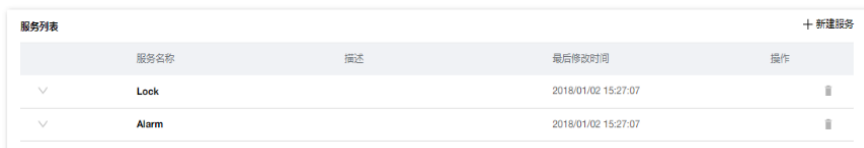
登录中国电信物联网开放平台开发者门户（<https://180.101.147.208:8093>）->Profile 开发->Profile 在线开发->自定义产品->创建全新产品->填写设备类型、设备型号、厂商 ID、厂商名称、协议类型，点击确定，如下图所示：



再根据设备的功能和业务场景添加服务，每个服务下可以增加属性（上报的数据）和命令（下发的命令，命令必须有字段），如下图所示：



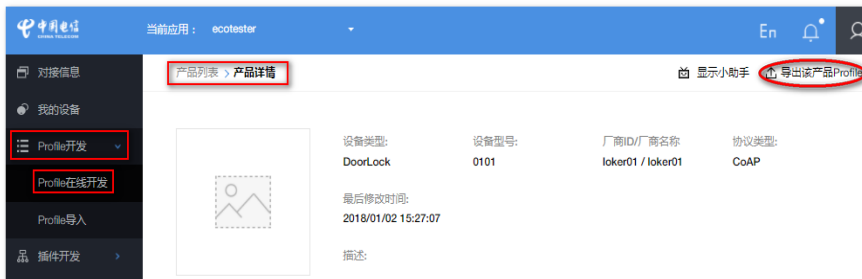
一个设备可以有多个服务：



 说明

在线开发的 profile 可以导出成 profile 资源包，然后上传到其他 IoT 网关上使用。

导出方法如下图所示：



2.4.1.4 终端编解码插件开发

NB-IoT 设备和中国电信物联网开放平台之间采用 CoAP 协议通讯，CoAP 消息的数据为应用层数据，应用层数据的格式由设备厂商自行定义。由于 NB-IoT 设备对省电要求通常较高，所以应用层数据一般采用十六进制格式。应用层数据在中国电信物联网开放平台做协议解析时，会转换成统一的 json 格式，以方便应用服务器使用。要实现十六进制消息与 json 格式消息的转换功能，中国电信物联网开放平台需要使用设备厂商提供的编解码插件。一款设备对应一个编解码插件。

开发编解码插件有两种方式：

方式一：中国电信物联网开放平台开发者门户在线插件开发。此方式实现了无码化的开发，开发复杂度低；适用于二进制消息和 json 格式消息不需经过计算，直接相互转换的场景；但并不适用于二进制消息和 json 格式消息需要计算后再转换的场景。

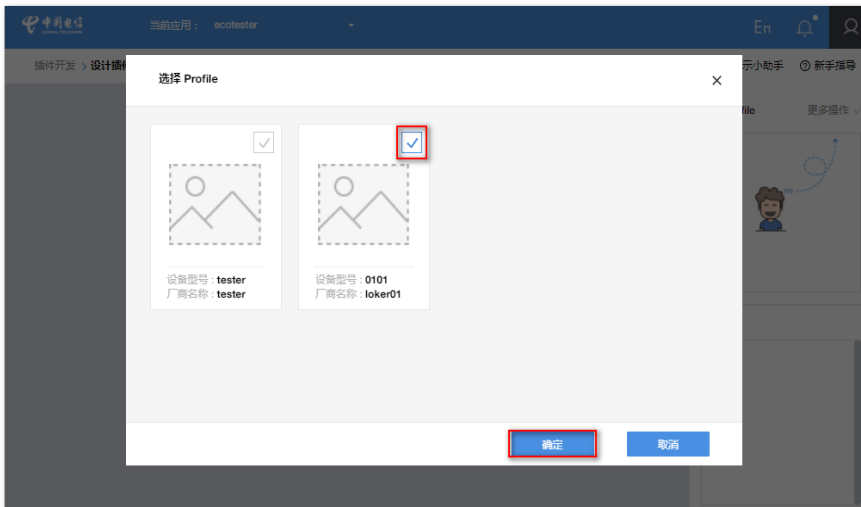
方式二：离线开发编解码插件。使用 Java 语言开发，开发复杂度相对方式一更高，适用于全部场景。

在线开发编解码插件

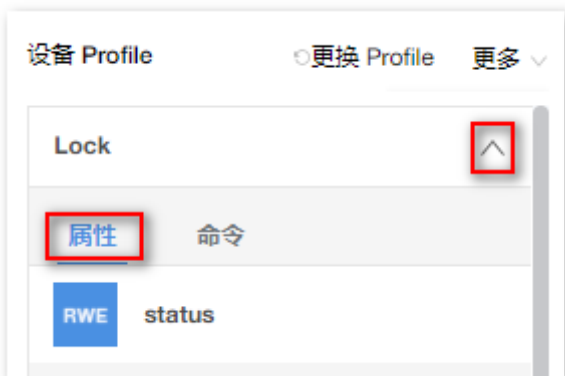
平台提供了编解码插件模板，推荐用户在对提供的模板进行修改后使用。如果需要自己定义插件，则可以根据如下步骤进行操作：

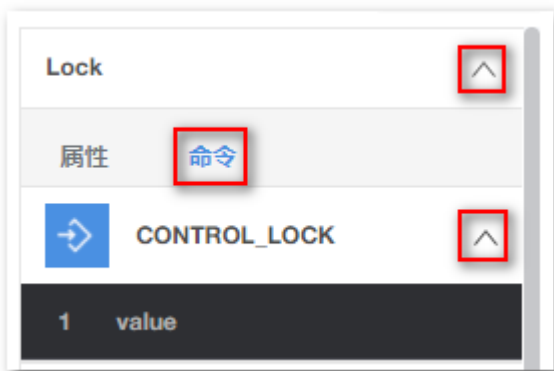
智能家电产业 NB-IoT 应用及开发指导书

登录中国电信物联网开放平台开发者门户->插件开发->插件开发(子菜单)
->开始设计->新建插件->选择 Profile (请等待 profile 加载完成后在选择对应
profile) ->点击确定

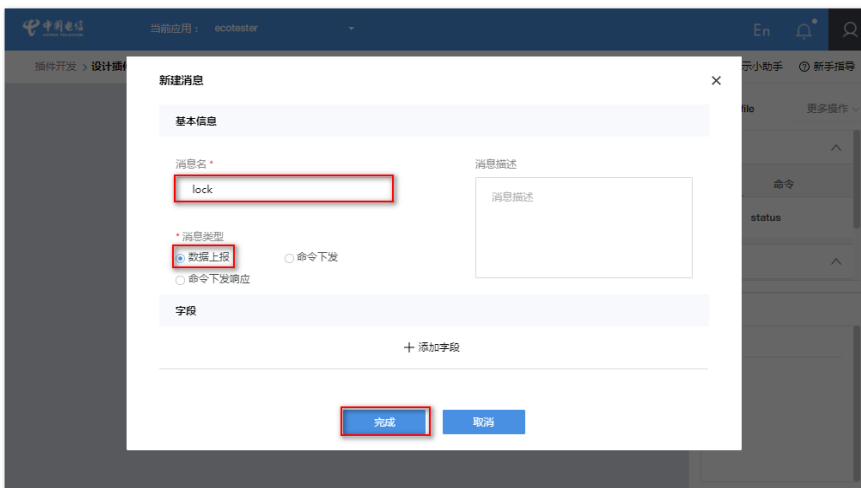


profile 中定义的属性和命令将会在界面右侧显示。





点击新增消息按钮，定义一条数据上报消息（消息名称可自定义）：



根据业务设计，一条码流可以被分成几段，每一段码流代表不同的业务数据，将需要上报的码流段提取出来，作为 profile 属性值上报给平台，再推送给应用。一条码流可以上报多个服务下的多个属性，也可以只上报一个服务的一个或者多个属性。

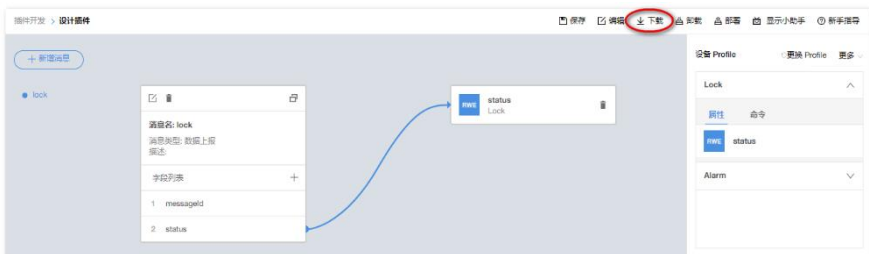
命令下发则与数据上报相反，是把命令字段的值取出来放入码流中。而一个命令下的参数值只能放到一条下发命令的消息中。



说明

中国电信物联网开放平台开发者门户支持下载在线完成的插件，下载后的编解码插件已经完成签名。

下载编解码插件：



离线编解码插件开发

离线编解码插件开发可参考[《中国电信 IoT 平台编解码库开发指南》](#)文档。编解码插件开发完毕后，请使用[“中国电信 NB-IoT 编解码插件检测工具”](#)进行校验。



插件包离线签名

当编解码插件开发完后，在安装到平台之前，需要先对插件包进行签名。此时需要下载离线签名工具并进行签名操作。操作步骤如下：

- (1) 使用浏览器登录中国电信物联网开放平台企业门户
- (2) 下载离线签名工具

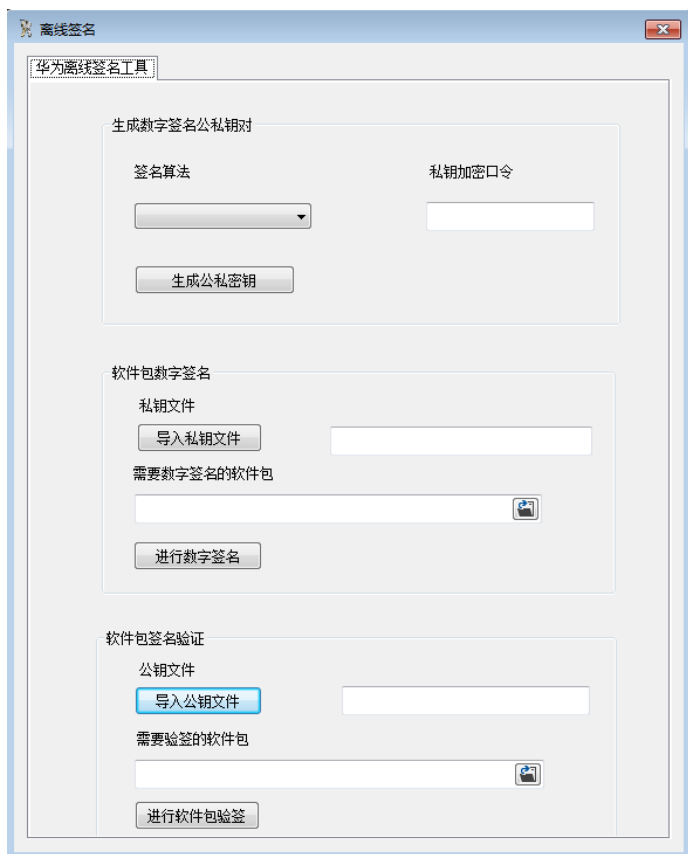
• 单击左侧  图标，打开管理页面

• 单击左侧导航栏“工具”，在右侧区域单击“下载”，下载离线签名工具

(3) 在下载路径找到压缩包“signtool.zip”，右键菜单中选择“Extract to signtool”解压缩至文件夹“signtool”

(4) 进入 signtool 文件夹，运行“signtool.exe”。操作界面如下所示

图2-5 离线签名工具



(1) 生成数字签名公私钥对。

- 根据实际情况选择签名算法。目前提供两种签名算法：
 - ECDSA_256K1+SHA256
 - RSA2048+SHA256
- 设置“私钥加密口令”。口令复杂度说明：
 - 口令长度至少为 6 个字符
 - 口令必须包含如下至少两种字符的组合：

- ◆ 至少一个小写字母
- ◆ 至少一个大写字母
- ◆ 至少一个数字
- ◆ 至少一个特殊字符：`~!@#% ^&*()-_+=\|[{ }];:","<.>/?` 和空格

- 单击“生成公私密钥”，在弹出的窗口中选择需要保存的目录，单击“确定”。

可在保存的目录下查看生成的公私密钥文件。

- 公钥文件：public.pem

- 私钥文件：private.pem

- (2) 对软件包进行数字签名及验证。

说明

离线签名工具只能对.zip 格式的压缩包进行数字签名。

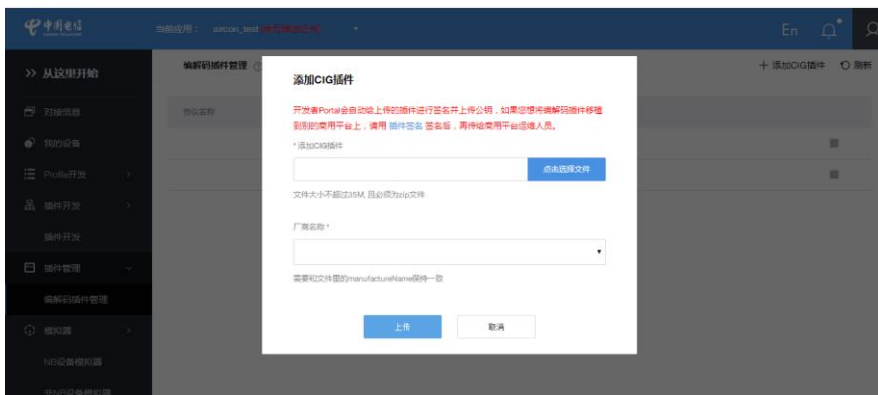
上传公钥及插件包

编解码插件化能够动态的新增编解码库，这样新增设备接入时，只需要把对应的编解码库动态导入到平台，就能够兼容新增的设备。请将公钥以及签名后的插件包发送给中国电信物联网开放平台支持人员，由中国电信完成公钥和签名后的插件包的上传配置。操作步骤如下：

- (3) 上传签名后的编解码插件包。

- 登录中国电信物联网开放平台开发者门户。
- 选择“菜单管理 > 配插件管理> 编解码插件管理”。
- 单击右上角的“添加 ICG 插件”按钮。
- 单击“点击上传文件”上传签名后的编解码插件包。

图2-6 上传签名后的编解码插件包



2.4.1.5 订阅终端消息

为了能接受设备上报的数据，NA 需要向 IoT 网关订阅通知消息。订阅成功后，设备上报数据时，IoT 网关会将数据推送到 callbackurl 上。

具体请参考[《中国电信物联网开放平台 API 参考》](#)。

2.4.1.6 注册设备中国电信物联网开放平台 API 参考

所有设备必须先在北向进行注册，才允许连接到平台。通过注册设备，平台会为每个设备分配一个唯一的标识 deviceId，后续应用操作这个设备时都通过 deviceId 来指定设备。另外，还返回 psk 参数（如果用户未指定 psk 参数，平台会随机分配一个参数），南向设备绑定时，如果设备与平台之间走 DTLS 加密通道，则须用到该参数，请保存。在中国电信物联网开放平台企业门户上也可以通过 deviceId 来找到设备。

📖 说明

verifyCode 和 nodeId 需要填写为设备唯一标识，请务必使用设备 IMEI。timeout 建议填写为 0。

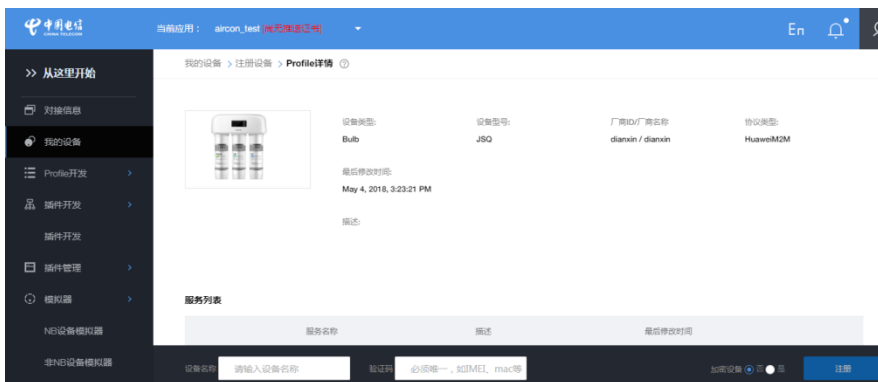
2.4.1.7 设置设备信息

为方便后续的管理及升级，所有接入平台的设备需要设置完整的设备信息。在 NB-IoT 方案中，注册设备后必须修改设备类型、厂商 ID、设备型号和协议类型。

具体请参考[《中国电信物联网开放平台 API 参考》](#)。

2.4.1.8 设备注册

- 登录中国电信物联网开放平台开发者门户。
- 选择“菜单管理 > 我的设备”。
- 单击右上角的“注册设备”按钮。
- 选择设备需要注册设备的 profile。
- 输入设备名称和验证码，即可注册成功。

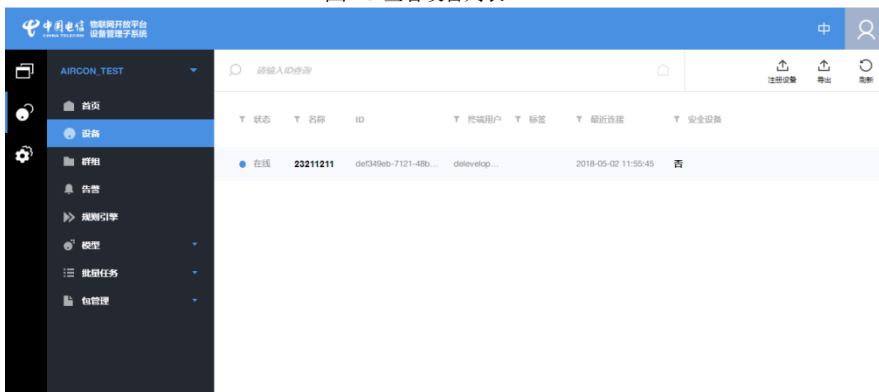


2.4.1.9 设备上线

配置好网络，开启设备，观察设备是否成功接入到平台。

登陆中国电信物联网开放平台企业门户的设备管理页面，查看设备列表，字段 ID 即为在第三步里注册设备时生成的 deviceId，status 字段表示设备的在线状态，如果状态是在线（online）表示设备已经成功的接入到平台，接着就可以接收设备的数据。

图2-7 查看设备列表



2.4.2 终端网络质量数据上报

2.4.2.1 特性描述

NB-IoT 业务时延、成功率等体验与终端所处的无线覆盖质量息息相关，电信 IoT 网关支持终端网络质量数据的解析。当终端出现丢包/时延大等问题，电信可根据网络质量信息快速处理，解决网络覆盖问题，保障用户体验

2.4.2.2 特性价值

终端出现丢包/时延大等问题，电信和行业客户可根据 IoT 网关上解析的网络质量信息对终端网络环境进行分析，对造成时延的问题进行初步的分析和判断，提高网络问题定位效率。

2.4.2.3 典型应用场景

适用于所有 NB-IoT 场景

2.4.2.4 解决方案

(1) 方案实现

在终端 Profile 文件及编解码插件中增加对网络质量参数的定义和解析。终端在上报业务数据之前，调用 NB 模组 AT 命令获取网络质量参数，将网络质量参数数据与业务数据合并上报。IoT 网关将网络质量数据进行解析并在中国电信物联网开放平台企业门户上呈现。

设备 Profile 开发：

智能家电产业 NB-IoT 应用及开发指导书

用户可以根据提供的 profile 模板对参数进行修改，模板中定义的设备为一个公共类型设备，提供 Connectivity 服务和 Transmission 服务。Connectivity 服务可用于命令下发，接收命令响应以及数据上报。命令名为 SET_CONNECTIVITY_STATE，下发的属性为 string 类型，以 json 结构体的形式下发给设备，通过编解码插件将 json 结构体转为 byte 数组最终转为 16 进制码流下发给设备，设备返回给平台的响应数据也是 string 类型，实际上设备返回的是一串 16 进制码流，但是编解码插件并不对其进行解析，实现透传。数据上报一共上报 signalPower, totalPower, cellId, signalEcl, signalSnr, signalPci, linkQuality, signalIndicator, batteryVoltage, length, rawData 共 11 个属性，前 10 个属性编解码插件会对其进行解析形成属性所对应的真实数据，但是并不会对 rawData 进行解析，实现半透传。用户可以根据实际需要对接插件参数进行修改。

设备上报数据说明：

服务类型 serviceType	说明	属性/命令	属性说明/参数/描述	属性类型(数据类型)	最小值	最大值	单位	访问模式 (method)	支持要求	V120/V150 芯片获取方式
Connectivity 服务	连接信息									
		signalPower	信号强度	Int 2 字节	-140	-44	dBm	RE	必选	NUESTATS 命令返回的 Signal power/10

智能家电产业 NB-IoT 应用及开发指导书

		totalPower	信号总强度	Int 2 字节			d B m	RE	必选	NUESTATS 命令返回的 total power/10
		CellID	设备所在小区id	Int 4 字节	0	429 496 729 5		RE	可选	NUESTATS 命令返回的 cell id
		signalECL	信号覆盖等级,ECL	Int 1 字节	0	2		RE	必选	NUESTATS 命令返回的 ECL
		signalSNR	信噪比SNR	Int 2 字节	-2 0	30	d B		必选	+NUESTATS 命令返回的 SNR/10 字段
		signalPCI	小区PCI	Int 2 字节	0	503		RE	可选	NUESTATS 命令返回的 PCI
		linkQuality(RSRQ)	连接质量, RSRQ	Int 1 字节	-1 9. 5	-3	d B	RE	可选	NUESTATS 命令返回的 RSRQ/10
		signalIndicator (RSSI)	接收的信号强度指示	Int 2 字节						芯片暂时不支持, 可以不上报
		batteryVoltage	电压	Int 1 字节						

Transmission 服务	数据信息									
		length	透传数据长度	Int 2 字节						
		rawData	透传数据	int						

编解码插件开发

修编解码插件，将设备上报的网络信号质量数据在平台上进行解析。对于透传的数据，平台可以不做解析。

设备上报数据（网络质量数据），消息类型字段加上需要解析的字段一共 20 个字节也就是 40 位，所以设备上报数据的码流至少要保证有 40 字节，否则平台无法接收到消息，可以参照下表：

智能家电产业 NB-IoT 应用及开发指导书

字节数	1	2	3	4	5	6	7	8	9	10	11	12	13	14
名称	msgType	signalPower		totalPower		cellId			signalEcl	signalSnr	signalPci			
定义	消息类型	信号强度		总信号强度		小区号			信号覆盖等级	信噪比	小区PCI			
字节数	15		16	17	18			19	20		21 ~			
名称	linkQuality		signalIndicator		batteryVoltage			length		rawData				
定义	连接质量		接收的信号强度指示		电压			指定透传数据长度		需要透传的数据				

数据上报的码流格式如下所示，最后的 666768 为 rawdata，需要透传的数据，不做解析：

```
0000020002000000040100020002010002010003666768
```

一个字节的最高位为符号位，无论上报的数据值是正数还是负数，都要通过数据值的二进制补码来转换成对应的 16 进制码流，正数的原码和补码是相同的，负数的补码等于原码的符号位不变，其余各位取反再加一。例如-1 的二进制原码为 10000001，补码为 11111111，对应的 16 进制码流就为 FF。上报的 16 进制码流经过编解码插件的处理会给应用端返回一个 ObjectNode，ObjectNode 中数据的封装如下代码所示：

```

public ObjectNode toJsonNode() {
    try {
        //组装body体
        ObjectMapper mapper = new ObjectMapper();
        ObjectNode root = mapper.createObjectNode();

        // root.put("identifier", this.identifier);
        root.put("msgType", this.msgType);

        //根据msgType字段组装消息体
        if (this.msgType.equals("deviceReq")) {
            ArrayNode arrynode = mapper.createArrayNode();

            //serviceId=Connectivity 数据组装
            ObjectNode ConnectivityNode = mapper.createObjectNode();
            ConnectivityNode.put("serviceId", "Connectivity");
            ObjectNode ConnectivityData = mapper.createObjectNode();
            ConnectivityData.put("signalPower", this.signalPower);
            ConnectivityData.put("totalPower", this.totalPower);
            ConnectivityData.put("cellId", this.cellId);
            ConnectivityData.put("signalEc1", this.signalEc1);
            ConnectivityData.put("signalSnr", this.signalSnr);
            ConnectivityData.put("signalPci", this.signalPci);
            ConnectivityData.put("linkQuality", this.linkQuality);
            ConnectivityData.put("signalIndicator", this.signalIndicator);
            ConnectivityData.put("batteryVoltage", this.batteryVoltage);
            ConnectivityNode.put("serviceData", ConnectivityData);
            arrynode.add(ConnectivityNode);

            //serviceId=Transmission 数据组装
            ObjectNode TransmissionNode = mapper.createObjectNode();
            TransmissionNode.put("serviceId", "Transmission");
            ObjectNode TransmissionData = mapper.createObjectNode();
            TransmissionData.put("length", this.length);
            TransmissionData.put("rawData", this.rawData);
            TransmissionNode.put("serviceData", TransmissionData);
            arrynode.add(TransmissionNode);

            root.put("data", arrynode);
        } else {
            root.put("errcode", this.errcode);
            root.put("mid", this.mid); //mid

            //组装body体, 只能为ObjectNode对象
            ObjectNode body = mapper.createObjectNode();
            body.put("length", this.length);
            body.put("rawData", this.rawData);
            root.put("body", body);
        }
    }
    return root;
}

```

由于 rawdata 没有进行解析，所以数据传递到应用端需要用户自己解析。

(2) 参考文档

中国电信物联网开放平台_设备能力描述文件 profile 开发指南:

<http://www.tianyiot.com/statics/shtml/gd4qpoec12ce32t.html>

中国电信物联网开放平台 V100R001C30 编解码库开发与升级指南:

<http://www.tianyiit.com/statics/shtml/y29odry3sck27lp.html>

(3) 测试验证

测试验证请参考[终端网络质量数据上报](#)。

2.4.3 终端固件远程升级 (FOTA)

2.4.3.1 特性描述

NB-IoT 芯片功能还在新版本中不断增强和优化, 芯片 FOTA 升级成为终端升级芯片版本的有效途径。芯片升级可通过电信 IoT 网关中国电信物联网开放平台企业门户进行操作。

说明

当在 IoT 网关上配置升级任务后, 终端有业务消息上报时, IoT 网关会直接与 NB 芯片协商并启动升级任务。有可能会中断此次的业务流程, 升级后完成后恢复。

2.4.3.2 特性价值

- 支持终端固件版本远程升级, 降低固件问题修复成本
- 支持灵活创建升级任务 (终端组, 时间段, 重试次数), 提升升级效率
- 支持升级状态上报&断点续传, 避免升级异常中断
- 支持升级包管理

2.4.3.3 典型应用场景

适用于所有 NB-IoT 场景

2.4.3.4 解决方案

(1) 预置条件

• NB-IoT 芯片升级需要终端进行适配, 主要流程请参见附录 FOTA 升级与 MCU 交互;

• 行业用户获取从源版本到目的版本的差分升级包 (一般从模组厂商获取, 不支持跨版本升级);

• 设备 profile 文件中已定义 omCapabilities 能力; 定义方法请参考附录设备升级能力定义

- 上传公钥;
- 固件包文件名称不支持包含中文字符;

- 固件包文件需要使用 winRAR 等压缩工具将固件包版本文件打包成 ZIP 格式的压缩包，并完成签名；
- 终端固件批量升级只支持按照组升级，建议业务发放时建议对终端进行分组；
- 在设备升级期间，建议 NA 不要向设备下发其他命令消息，否则会导致设备升级失败。NA 可以通过调用平台北向接口，订阅设备升级状态，避免在设备升级期间下发其他命令消息；请参考附录 NA 北向订阅消息。

(3) 方案实现

获取离线升级包（差分包）

设备厂商自行发布用于设备固件升级的文件。

固件包离线签名及上传

使用离线签名工具对软件版本包进行签名，签名工具使用方法请参考[插件包离线签名](#)。

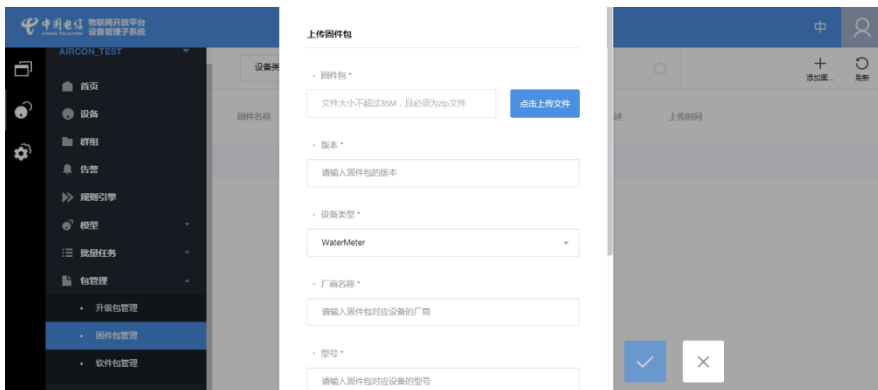
上传升级包之前需要上传对应的公钥信息，公钥来源为对编解码插件进行签名时生成的 public.pem 文件。登陆中国电信物联网开放平台企业门户，选择应用管理，点击进入对应的应用下，在升级认证管理页面上传公钥，界面如下图所示：



上传固件包

在设备管理页面，选择对应的应用，点击包管理，在固件包管理页面点击添加固件包，选择签名后的升级包，填写固件包的版本信息，可用的设备类型，厂商，型号，协议信息后，点击确认执行固件包上传，界面如下图所示：

固件包上传界面



创建固件升级任务

平台支持对批量设备进行固件升级，也支持指定某个设备进行固件升级，文中以批量设备固件升级为例，讲解具体的创建批量固件升级任务的操作步骤。

步骤 1: 用户登录中国电信物联网开放平台企业门户，进入设备管理→批量任务→软件批量任务→固件升级页面



步骤 2: 在右上角点击“+”创建批量固件升级任务，在弹出的固件升级任务框填写任务名称以及选择相应的重试策略，点击下一步按钮。

固件升级

1
 任务信息

2
 设备选择

3
 固件包选择

* 任务名

* 重试策略

* 重试次数

>

×

步骤 3: 选择要升级的目标设备群组，点击下一步按钮

固件升级

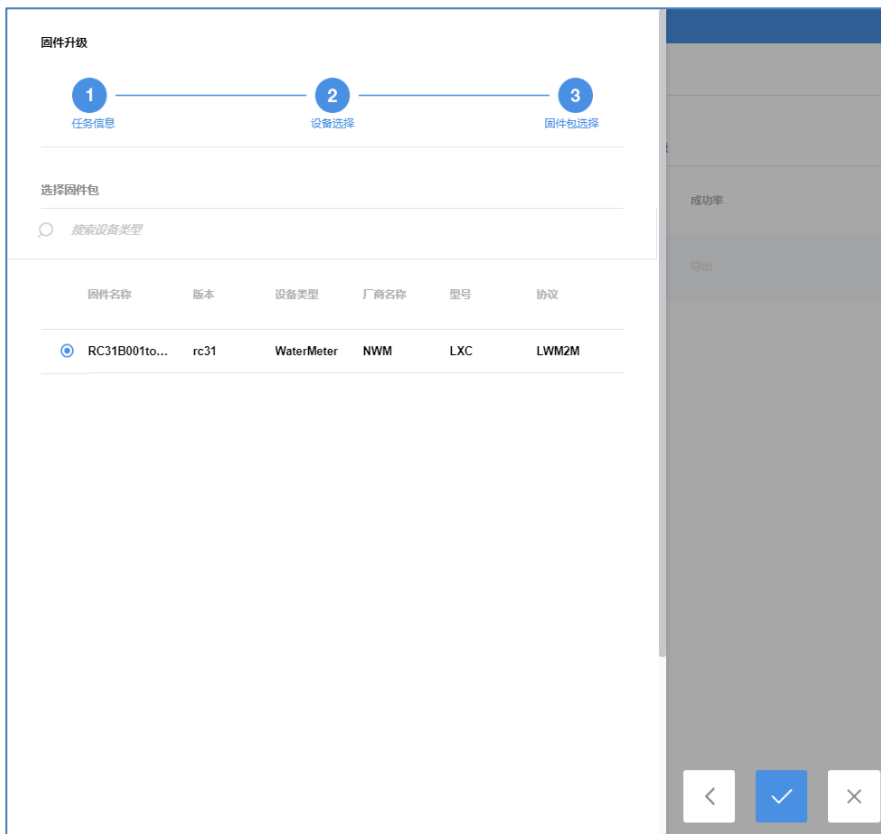


选择设备群组

<input checked="" type="checkbox"/>	组名	设备数	描述
<input checked="" type="checkbox"/>	Other		Other Devices



步骤 4: 选择要升级的目标版本，然后点击完成按钮，固件批量升级任务即创建完成。



(3) 测试验证

测试验证请参考[终端固件远程升级 \(FOTA\)](#)

2.5 智能家电与 NB-IoT 网络对接开发

2.5.1 下行控制时延优化（提前寻呼）

2.5.1.1 特性描述

通过手机 APP 或应用系统对智能家电终端进行远程控制时一般有时延要求。为降低控制指令到达终端的时延，建议在控制命令下发前对终端进行

一次寻呼（下发下行指令到终端），在用户下发下行指令时将节省寻呼终端的时延（一般平均为 1.28 秒）。

📖 说明

每次寻呼终端后，有效时间目前为 20 秒钟。如果 20 秒钟之内终端没有上行/下行数据，则再次下发命令仍然需要进行寻呼。

2.5.1.2 特性价值

提前激活终端进入连接态，降低远程控制时延，提升终端用户体验

2.5.1.3 典型应用场景

有严格时延要求的业务，如空调，洗衣机

2.5.1.4 解决方案

(1) 预置条件

- 中国电信物联网开放平台企业门户上 NB-IoT 应用参数已配置为 DRX 模式，具体配置方案请参考 NB-IoT 参数配置

- APN 关闭 eDRX 及 PSM 特性

(2) 方案实现

中国电信物联网开放平台企业门户上 NB-IoT 应用参数配置为 DRX 模式：



方案说明

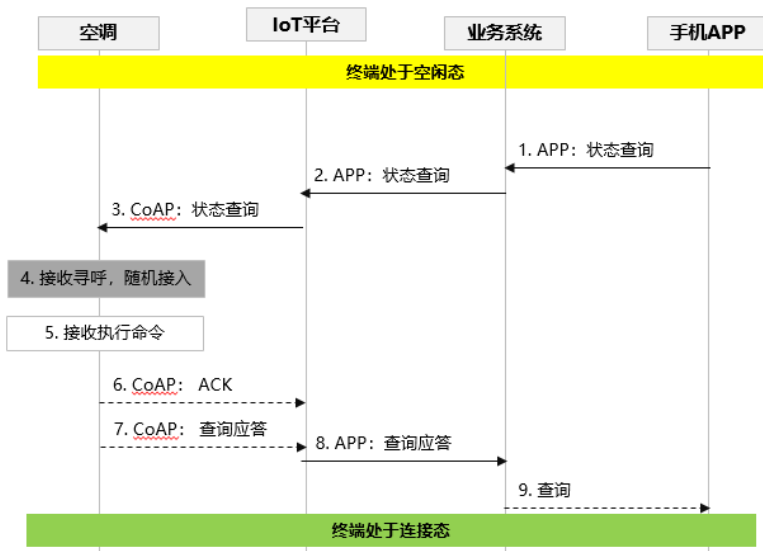
通常智能家电类设备通过手机 APP 控制进行控制，可在手机 APP 上查询设备状态，下发控制指令。在 NB-IoT DRX 模式下，当前全网最小寻呼周期设置为 2.56s，如果业务流程不做任何优化的情况下，通过手机 APP 下发控制指

令，加上各处理单元（应用服务器，IoT 网关，电信网络）的延迟，下行时延通常会超过 5s，用户体验不好。因此我们可以通过优化业务流程，提前触发终端进入连接态，降低时延。

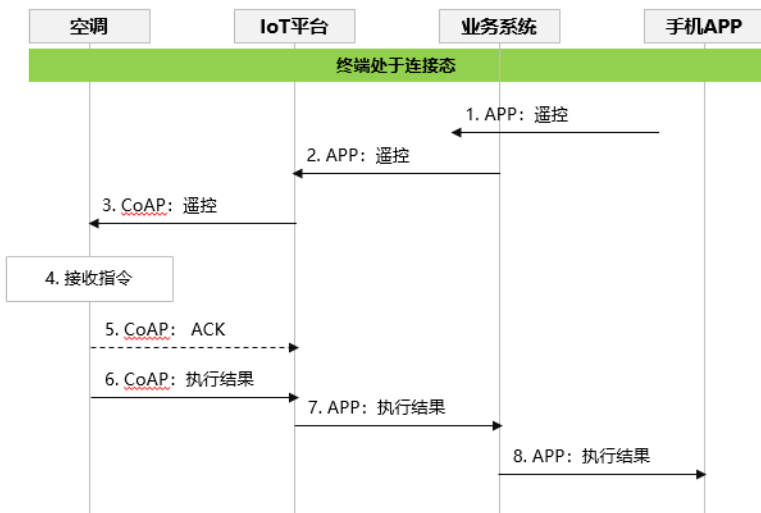
具体实现

当用户打开 APP，将设备与 APP 绑定后，即可通过获取终端状态的消息或其他消息，将设备激活进入连接态，等待用户下发下行命令。我们以空调为例，说明具体业务流程：

• 主动寻呼业务流程



• 主动寻呼成功后，命令下发业务流程



(3) 测试验证

测试验证请参考[下行控制时延优化（提前寻呼）](#)

2.5.2 终端功耗优化（快速释放连接）

2.5.2.1 特性描述

大部分 NB-IoT 终端由电池供电，对功耗有严格的要求，例如智能门锁，门锁上报一次数据后，业务流程就结束了。正常情况下，NB-IoT 终端上报完数据后，需要等 UE 不活动定时器超时（默认 20 秒），才会从连接态进入 IDLE 态（功耗大）。如果终端确认已经是最后一次业务交互，可提前释放与基站侧的连接，进入 IDLE 态和 PSM 态，降低终端功耗。以下是实验室测试的终端电流参考数据：

- 连接态平均电流参考：50mA
- IDLE 态平均电流参考：1mA
- PSM 态平均电流参考：5uA

计算使用 RA 单次业务流程可以节省的能耗：270uAH（实验室测试数据，供参考）

2.5.2.2 特性价值

- 门锁：终端快速释放连接，降低终端功耗
- 移动场景：提升小区重选及时性，确保终端无线信号质量

2.5.2.3 典型应用场景

终端确认已经是最后一次业务交互的场景下，如门锁，电动车。

2.5.2.4 解决方案

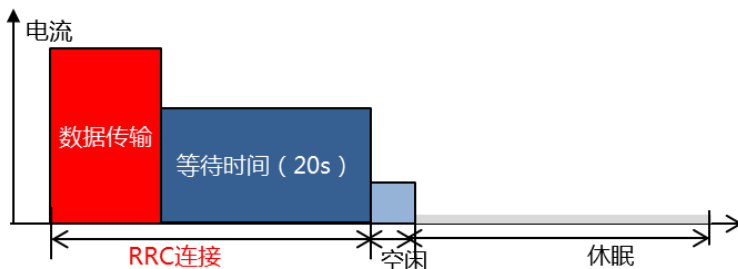
(1) 预置条件

- 海思 Boudica 芯片 V120 B657SP5 以上版本

(2) 方案实现

方案介绍

非快速离线的情况下，UE 上报完数据后等待 20s 后进入 IDLE 和 PSM 状态



支持快速离线的情况下，UE 完成最后一次数据交互后，基站与终端的连接快速释放，终端进入 IDLE 和 PSM 状态

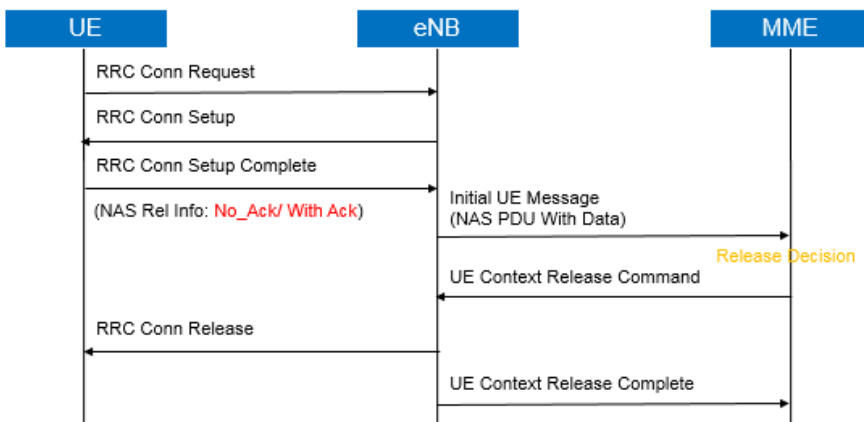


具体实现

当 UE 发送数据时，在数据中携带是否需要应用服务器下行确认消息的指示给 MME(No_Ack/ With_Ack).

- 如果是 No_Ack, MME 在将数据发送给 S-GW 后立即释放链路。
- 如果是 With_Ack, MME 在将应用服务器的 ACK 消息发送给 UE 后释放链路。

业务流程如下：



具体 AT 命令请参考 [AT+MLWULDATAEX\(以海思为例，具体详见对应模组厂家指令集\)](#)

(3) 测试验证

测试验证请参考[终端功耗优化（快速释放连接）](#)

第三册

智能家电 NB-IOT 应用测试验证

3 智能家电 NB-IoT 应用测试验证

3.1 终端接入验证

预置条件	1、已申请测试账号
测试步骤	<p>步骤1: 创建应用，并设置 NB-IoT 参数</p> <p>步骤2: 登录中国电信物联网开放平台开发者门户，在线开发设备 Profile 文件及编解码插件</p> <p>步骤3: 通过应用服务器订阅终端消息</p> <p>步骤4: 注册设备并修改设备信息</p> <p>步骤5: 通过 AT 命令触发终端上线</p> <ul style="list-style-type: none"> • 模组上电: AT+CFUN=1 • 模组进行网络注册，检验命令 AT+CEREG? • 检查 eDRX 模式: AT+CEDRXRDP (第一个参数为 0 表示关闭, 5 表示打开) • 关闭 eDRX 模式: AT+CEDRXS=0,5 • 检验 PSM 状态: AT+CPSMS ? (第一个参数为 0 表示关闭, 1 表示打开) • 关闭 PSM: AT+CPSMS=0 • 检验 PSM 状态: AT+CPSMS? (第一个参数为 0 表示关闭, 1 表示打开) • 检查模组接入的网络信息及网络质量参数: AT+NUESTATS <p>步骤6: 将获取的网络质量参数与原业务数据合并后上报: AT+NMGS 或 AT+MLWULDATAEX (支持 coap con 数据发送, 及 RA 功能)</p>

预期结果	1、 终端可正常接入网络并上报数据 2、 IoT 网关上可查看终端上报历史数据
实际结果	

3.2 终端与 IoT 网关对接验证

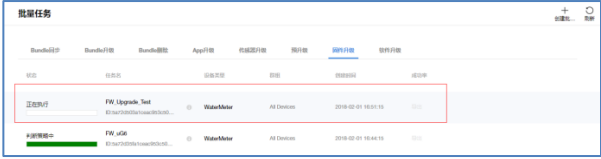
3.2.1 IoT 网关对接验证

3.2.2 终端网络质量数据上报

预置条件	1、 已申请测试账号 2、 已在 IoT 网关上创建应用并已设置 NB-IoT 参数为 DRX 模式
测试步骤	<p>步骤1: 登录中国电信物联网开放平台开发者门户, 在线开发设备 Profile 文件及编解码插件, Profile 及编解码插件中均已增加网络质量相关字段</p> <p>步骤2: 通过应用服务器订阅终端消息</p> <p>步骤3: 注册设备并修改设备信息</p> <p>步骤4: 通过 AT 命令触发终端上线</p> <ul style="list-style-type: none"> • 模组上电: AT+CFUN=1 • 模组进行网络注册, 检验命令 AT+CEREG? • 检查 eDRX 模式: AT+CEDRXRDP (第一个参数为 0 表示关闭, 5 表示打开) • 关闭 eDRX 模式: AT+CEDRXS=0,5 • 检查 eDRX 模式: AT+CEDRXRDP • 检验 PSM 状态: AT+CPSMS? (第一个参数为 0 表示关闭, 1 表示打开) • 关闭 PSM: AT+CPSMS=0 • 检验 PSM 状态: AT+CPSMS? (第一个参数为 0 表示关闭, 1 表示打开) • 检查模组接入的网络信息及网络质量参数:

	<p>AT+NUESTATS</p> <p>步骤5: 将获取的网络质量参数与原业务数据合并后上报: AT+NMGS 或 AT+MLWULDATAEX (支持 coap con 数据发送, 及 RA 功能)</p>
预期结果	<p>1、 IoT 网关上设备详情页面可查看终端信号强度</p> <p>2、 IoT 网关上设备历史数据可以查看完整的网络质量数据</p>
实际结果	

3.2.3 终端固件远程升级 (FOTA)

预置条件	<ol style="list-style-type: none"> 1、 已申请测试账号 2、 已在 IoT 网关上创建应用并已设置 NB-IoT 参数为 DRX 模式 3、 已在设备 profile 文件中定义 omCapabilities 能力 4、 已获取到差分升级包 5、 终端已接入 IoT 网关
测试步骤	<p>步骤1: 登录 IoT 网关中国电信物联网开放平台企业门户</p> <p>步骤2: 创建 FOTA 升级任务, 执行 FOTA 升级</p> <p>步骤3: 在 IoT 网关批量固件升级任务列表中查看到执行的情况, 是否升级成功</p> <p>步骤4: 通过 AT+CGMR 命令查看终端固件</p>
预期结果	<ol style="list-style-type: none"> 1、 在 IoT 网关批量固件升级任务列表中查看到执行的情况, 终端升级成功 
实际结果	<ol style="list-style-type: none"> 2、 命令查询到的固件版本与实际版本一致

3.3 终端与网络对接验证

3.3.1 下行控制时延优化（提前寻呼）

预置条件	<ol style="list-style-type: none"> 1、 已申请测试账号 2、 已在 IoT 网关上创建应用并已设置 NB-IoT 参数为 DRX 模式 3、 终端已接入 IoT 网关 4、 应用侧已针对手机 APP 业务进行优化，打开 APP 后支持提前激活终端进入连接态 5、 终端网络覆盖等级为 0,1 6、 终端已关闭 eDRX 和 PSM 模式
测试步骤	步骤1： 使用手机 APP 下发控制命令，比较优化前后时延信息
预期结果	时延明显小于未优化前 (<10s)
实际结果	

3.3.2 终端功耗优化（快速释放连接）

预置条件	<ol style="list-style-type: none"> 1、 已申请测试账号 2、 终端已接入 IoT 网关 3、 芯片版本：海思 Boudica 芯片 V120 B657SP5 以上版本
测试步骤	<p>步骤1： 使用终端上报消息，且消息中携带 RELEASE 释放辅助指示 (0x0001) (样例：AT+MLWULDATAEX =3,AA34BB,0x0001)</p> <p>步骤2： 终端上报消息，且消息中携带 RELEASE_AFTER_REPLY 释放辅助指示 (0x0101) (样例：AT+MLWULDATAEX =3,AA34BB,0x0101)</p>
预期结果	消息上报后，RRC 立即释放 (0x0001) 或收到平台确认消息后 (0x0101)立即释放链路，查询芯片处于 IDLE 状态：AT+CSCON? (第二个参数为 0 表示已释放，1 表示未释放)
实际结果	

4 附录

4.1 AT+MLWULDATAEX 命令（以海思为例，具体详见对应模组厂家指令集）

4.1.1 命令语法

Command	Response	Example
+MLWULDATAEX	OK +CME ERROR: <err>	AT+MLWULDATAEX X =3,AA34BB,0x0001 OK

4.1.2 命令描述

终端设备通过该命令将数据发送给 NB 模组，模组在向 NB-IoT 网关发送 CON 或 NON 消息时携带释放辅助指示（RAI）

4.1.3 参数描述

Parameter	Description
< length >	发送数据的长度
<data>	16 进制形式的字符串

Parameter	Description
<mode>	<p>0x0000: 发送 NON 消息</p> <p>0x0001: 发送 NON 消息, 且携带 RELEASE 释放辅助指示</p> <p>0x0100: 发送 CON 消息</p> <p>0x0101: 发送 CON 消息, 且携带 RELEASE_AFTER_REPLY 释放辅助指示</p>

 说明

如果发送 NON 消息, 且携带 RELEASE 释放辅助指示 (0x0001), 在 NON 消息发送成功后立马释放 RRC 连接; 如果发送 CON 消息, 且携带 RELEASE_AFTER_REPLY 释放辅助指示 (0x0101) 会在收到平台的 ACK 消息后立马释放 RRC 连接;

4.2 AT 命令返回结果

4.2.1 华为海思芯片

- 模组上电

AT+CFUN=1

- 模组进行网络注册

模组默认自动注册网络; 检验注册状态 AT+CEREG?, 如下图第二个参数 1 表示注册。

[09:40:52.999]发→AT+CEREG?

[09:40:53.015]收←

+CEREG:0,1

OK

- 关闭 eDRX 模式

关闭 eDRX: AT+CEDRXS=0,5;

检验 eDRX 状态 AT+CEDRXRDP，如下图第一个参数为 0 表示关闭，5 表示打开：

[09:41:30.516]发→AT+CEDRXRDP

[09:41:30.531]收←

+CEDRXRDP:0

OK

- 关闭 PSM 模式

关闭 PSM: AT+CPSMS=0;

检验 PSM 状态 AT+CPSMS?，如下图第一个参数为 0 表示关闭，1 表示打开：

[09:41:48.806]发→AT+CPSMS?

[09:41:48.821]收←

+CPSMS:1,,01000001,00000101

OK

- 检查模组接入的网络状态

AT+NUESTATS，查看 RSRP/SINR/ECL，ECL 在 0 或 1 说明网络信号正常，2 标示信号不好)

[09:44:03.938]发→AT+NUESTATS

[09:44:03.954]收←

Signal power:-945

Total power:-765

TX power:110

TX time:654

RX time:29795

Cell ID:8454866

ECL:1

SNR:-33

EARFCN:2506

PCI:155

RSRO:-166

OK

4.3 设备升级能力定义

4.3.1 Profile 定义

测试 FOTA 升级之前，需要确保设备 Profile 包含 omCapabilities 能力，否则平台无法对该设备创建固件升级任务。

Profile 样例：

```
{
  "devices": [
    {
      "manufacturerId": "Huawei",
      "manufacturerName": "Huawei",
      "model": "NBloTDevice",
      "protocolType": "CoAP",
      "deviceType": "SmartDevice",
      "omCapability": {
        "upgradeCapability": {
          "supportUpgrade": false
        },
        "fwUpgradeCapability": {
          "supportUpgrade": true,
          "upgradeProtocolType": "LWM2M",
          "downloadProtocolType": "CoAP"
        }
      }
    }
  ]
}
```


4.4 FOTA 升级与 MCU 交互

升级前，请认真阅读[《FOTA 升级与 MCU 交互》](#)文档，避免升级异常；

4.5 编解码插件开发指导

编写网络质量参数上报数据 profile 及编解码插件时，请参考如下指导：

4.5.1 Profile 模板开发

用户可以根据提供的 profile 模板对参数进行修改，模板中定义的设备为一个公共类型设备，提供 Connectivity 服务和 Transmission 服务。Connectivity 服务可用于命令下发，接收命令响应以及数据上报。命令名为 SET_CONNECTIVITY_STATE，下发的属性为 string 类型，以 json 结构体的形式下发给设备，通过编解码插件将 json 结构体转为 byte 数组最终转为 16 进制码流下发给设备，设备返回给平台的响应数据也是 string 类型，实际上设备返回的是一串 16 进制码流，但是编解码插件并不对其进行解析，实现透传。数据上报一共上报 signalPower, totalPower, cellId, signalEcl, signalSnr, signalPci, linkQuality, signalIndicator, batteryVoltage, length, rawData 共 11 个属性，前 10 个属性编解码插件会对其进行解析形成属性所对应的真实数据，但是并不会对 rawData 进行解析，实现半透传。用户可以根据实际需要可对插件参数进行修改。

如下为设备上报的数据说明：

signalPower: 信号强度（2 字节）

totalPower: 总信号强度（2 字节）

cellId: 小区号（4 字节）

signalEcl: 信号覆盖等级（1 字节）

signalSnr: 信噪比（2 字节）

signalPci: 小区 PCI（2 字节）

linkQuality: 连接质量（1 字节）

signalIndicator: 接收的信号强度指示 (2 字节)

batteryVoltage: 电压 (1 字节)

length: 指定需要透传数据的数据长度 (2 字节)

rawData: 需要透传的数据

其中 profile 模板中的 manufacturerId, manufacturerName, model, deviceType 四个属性是用户必须要修改的属性, 解压 profile 模板压缩文件后进入 profile 目录, 打开 devicetype-capability.json 文件对其进行修改, 位置如下图所示:

```

{
  "devices": [
    {
      "manufacturerId": "Huawei",
      "manufacturerName": "Huawei",
      "model": "NBIoTDevice",
      "protocolType": "CoAP",
      "deviceType": "GlobalTemplate",
      "serviceTypeCapabilities": [
        {
          "serviceId": "Connectivity",
          "serviceType": "Connectivity",
          "option": "Mandatory"
        },
        {
          "serviceId": "Transmission",
          "serviceType": "Transmission",
          "option": "Mandatory"
        }
      ]
    }
  ]
}

```

同时在使用编解码插件包的时候, 也需要修改以上属性, 解压编解码插件包, 打开 package-info.json 文件, 对属性进行修改, 位置如下图所示:

```

"deviceType": "GlobalTemplate", "manufacturerName": "Huawei", "model": "NBIoTDevice",

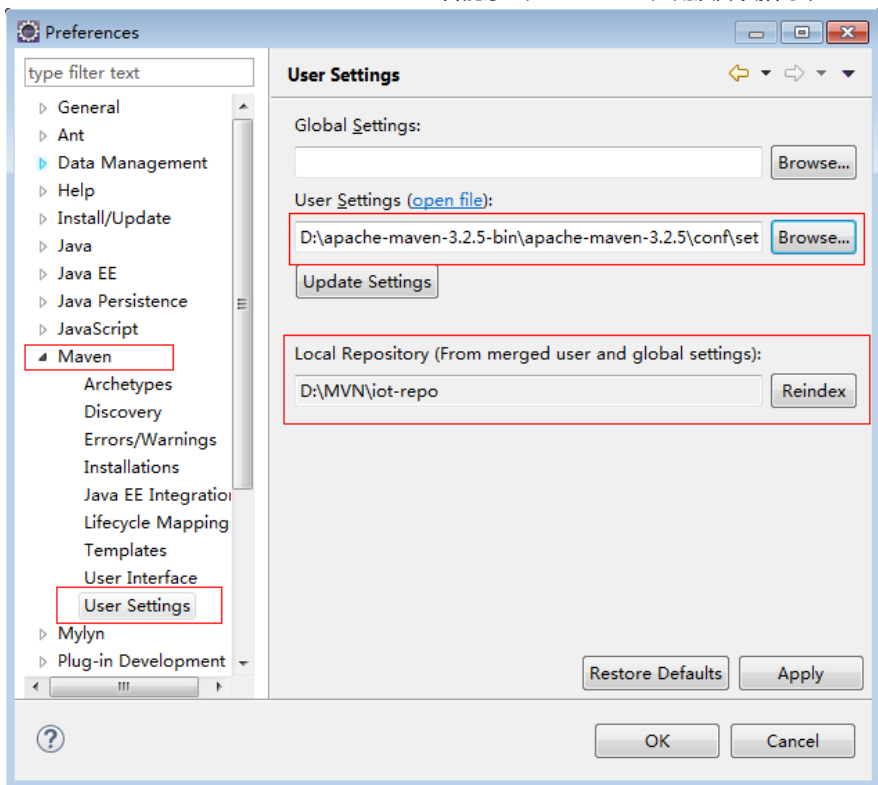
```

4.5.2 编解码插件开发

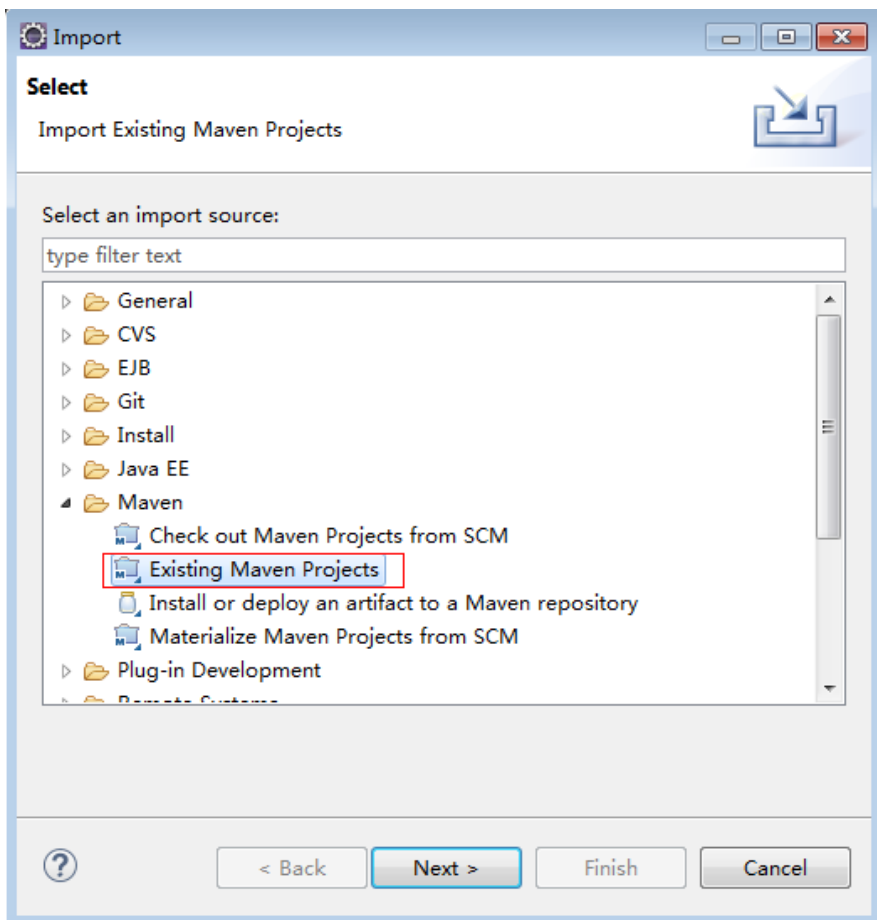
解压 apache-maven-3.2.5.zip，在解压后得到的文件夹中找到 conf 目录下的 settings.xml 配置文件，打开配置文件找到<localRepository>标签，设置自己的本地仓库路径用来存储工程所依赖的 jar 包，如下图所示：

```
<localRepository>D:\MVN\iot-repo</localRepository>-->
```

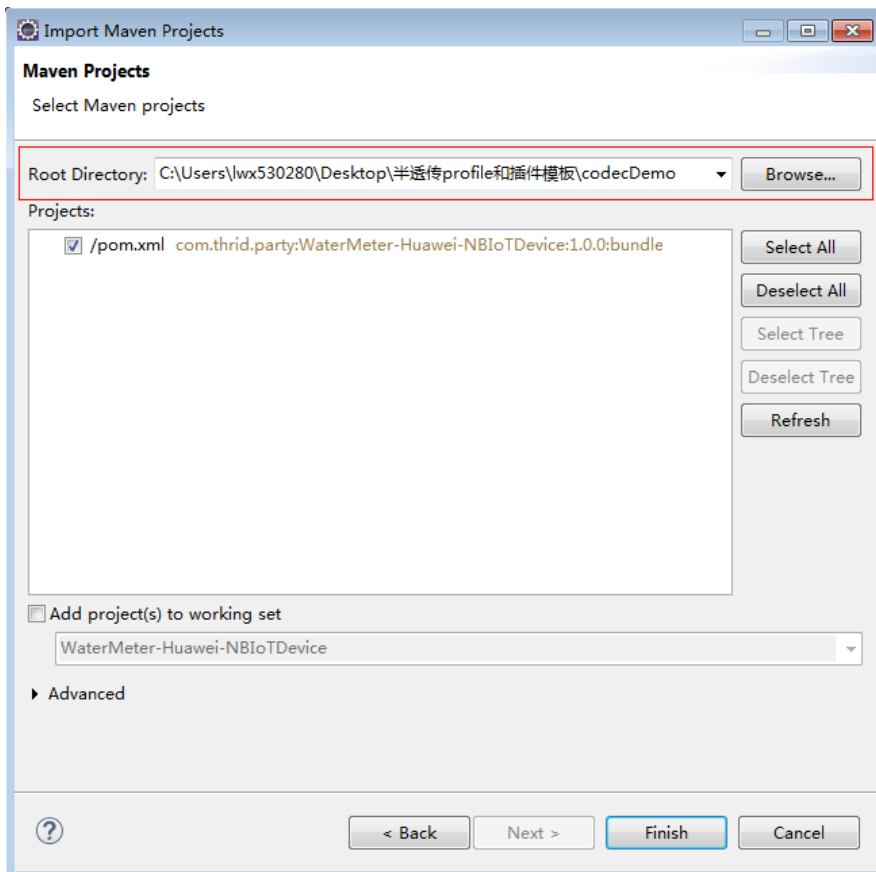
以 eclipse 开发环境为例，打开 eclipse，先配置 maven 插件，点击上方导航栏的 Window->Preferences 弹出如下窗口，在左侧导航栏中选择 Maven->User Settings，将 User Settings 设置为 maven 插件 settings.xml 配置文件的所在路径，此例为 D:\apache-maven-3.2.5-bin\apache-maven-3.2.5\conf\settings.xml，选择后 Local Repository 会自动变为你在 settings.xml 配置文件中设置的本地仓库路径。配置完毕后点击 Apply 后再点击 OK。



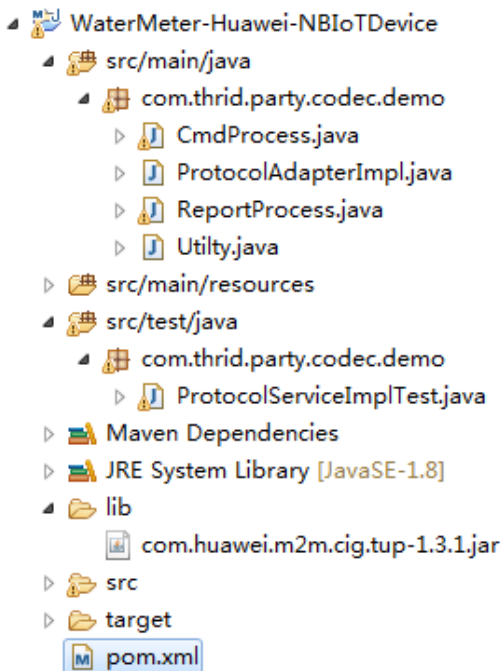
解压 codecDemo.zip, 将工程导入到 eclipse, 点击上方导航栏的 File->Import 弹出如下窗口, 选择 Maven 目录下的 Existing Maven Projects 后点击 Next。



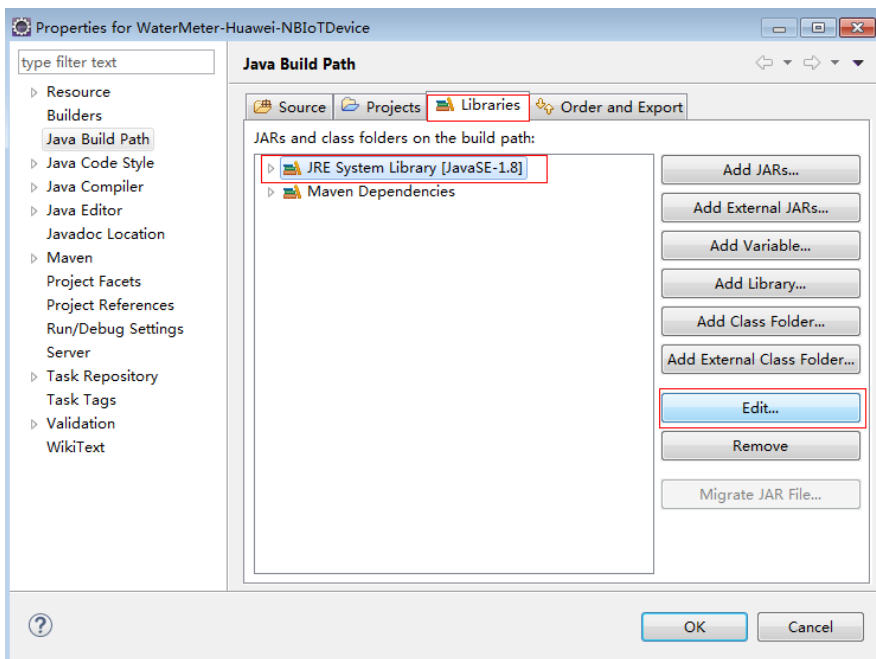
之后弹出如下窗口，选择自己所解压出的 codecDemo 工程后点击 Finish。



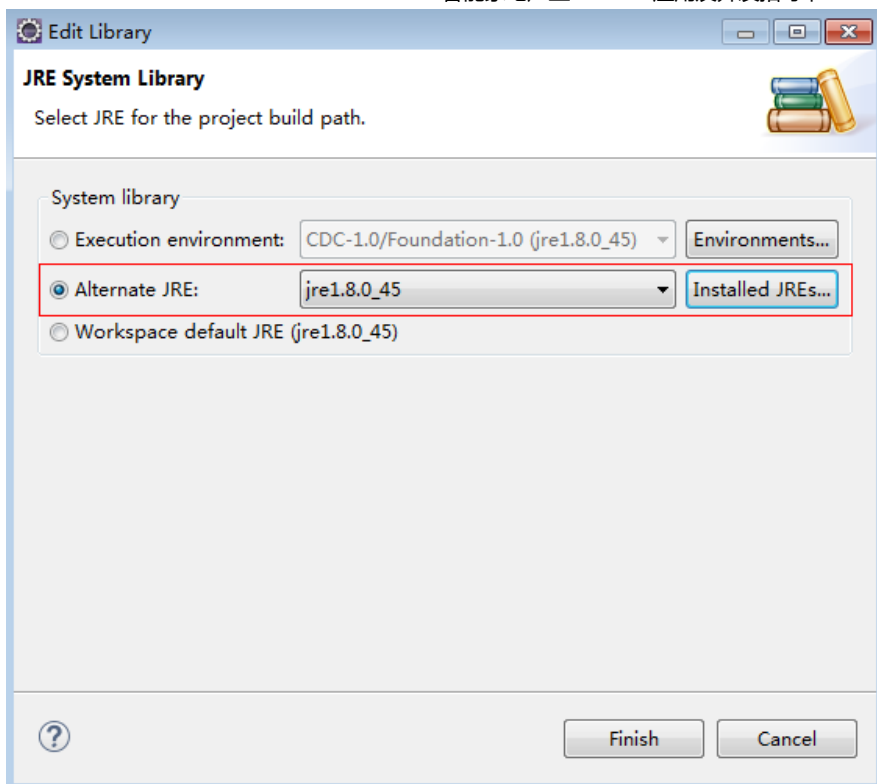
导入成功后，如下为 maven 工程的目录结构：



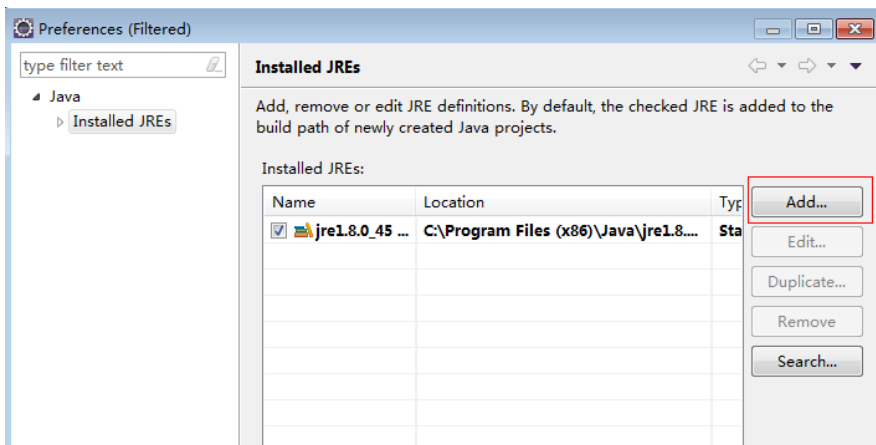
在这里需要多配置一步，因为该工程默认使用的 JRE 是 JavaSE-1.8，在打包成 jar 时由于缺少 tools.jar 工具包会打包失败，因此这里将 JRE 配制成自己的 JDK。右键点击工程，选择 Build Path->Configure Build Path 弹出如下窗口，选择 Libraries 页签下的 JRE System Library 后点击 Edit。



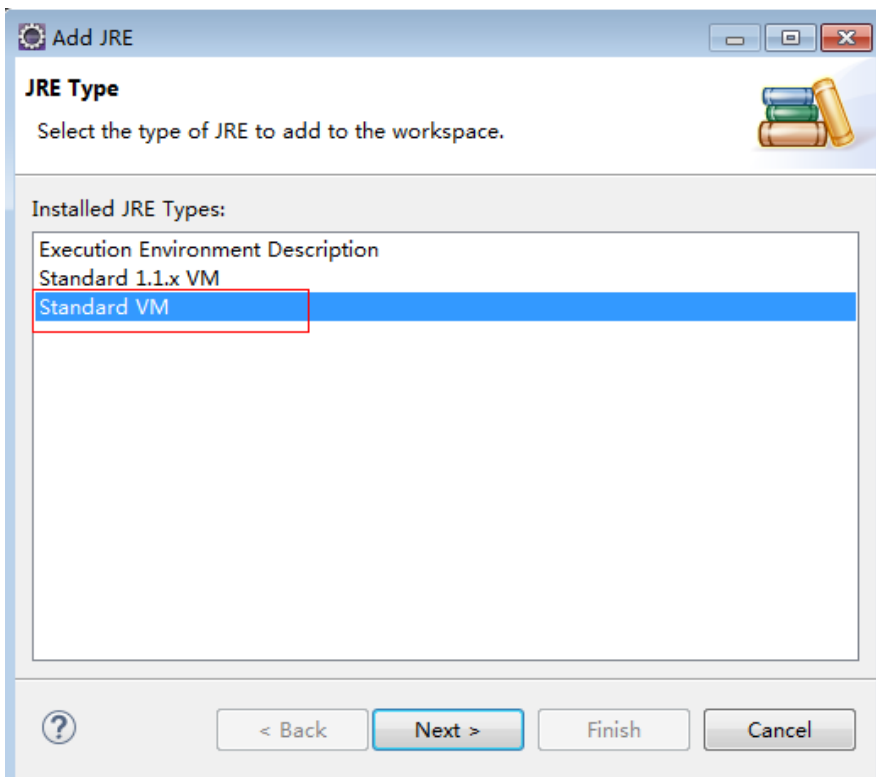
选择 Alternate JRE，点击 Installed JREs。



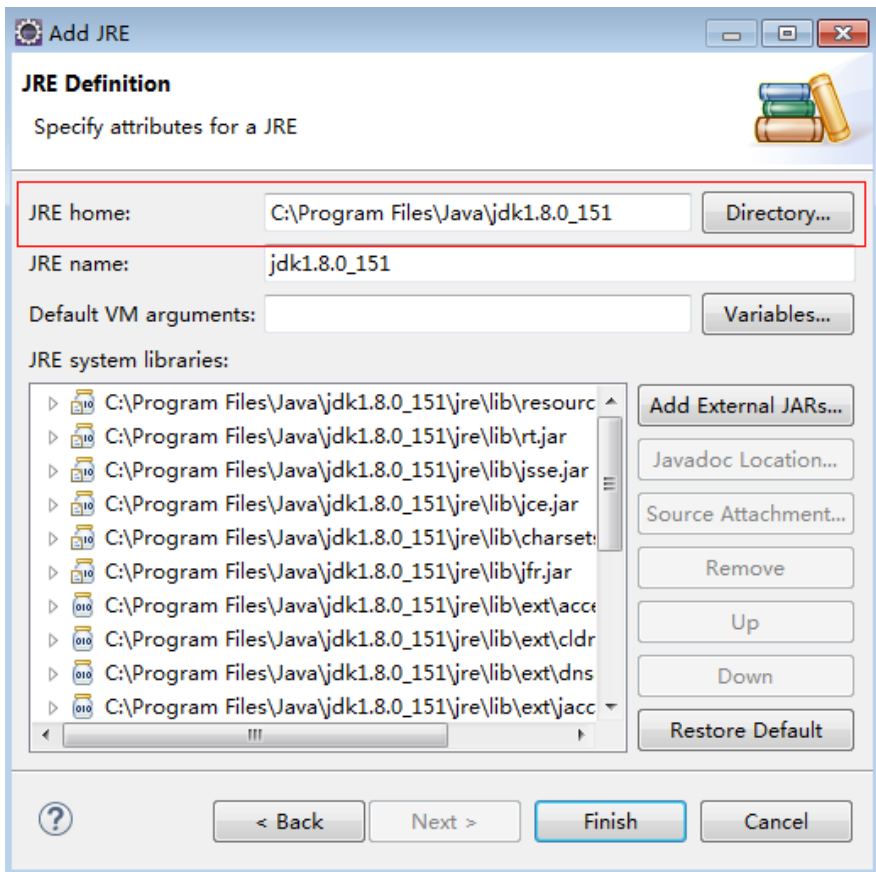
在弹出的窗口中点击 Add。



选择 Standard VM 后点击 Next。



选择 JDK 后点击 **Finish**，最后选择自己刚才添加的 JDK 后点击 **OK** 即可。



ProtocolAdapterImpl 类中包含了 encode 和 decode 方法,为编码和解码的具体实现。该类中以下的两个属性是要求用户必须要修改的,修改的值要与 profile 中设置的值保持一致。

```
// 厂商名称
private static final String MANU_FACTURERID = "Huawei";
// 设备型号
private static final String MODEL = "NBIoTDevice";
```

ReportProcess 类为设备给平台上报数据的处理，有两种消息类型，分别为设备上报的数据以及设备对平台命令的响应。用户可以根据代码中提供的参数注释对参数进行修改。

CmdProcess 类为平台给设备下发数据的处理，有两种消息类型，分别为平台下发的命令以及平台收到设备上报数据的响应。用户可以根据代码中提供的参数注释对参数进行修改。

ProtocolServiceImplTest 类为测试类，包含了四种消息类型的测试，双击 ProtocolServiceImplTest. 打开该类，用户可以通过右键该类选择 Run As-> 1 JUnit Test 执行。

执行结果如下：

```
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
{"msgType": "deviceReq", "data": [{"serviceId": "Connectivity", "serviceData": {"signalPower": 12808, "totalPower": -29437, "cellId": 543986393, "signalIsc": 1, "signalIsn": 257, "signalPci": {"msgType": "deviceRsp", "errcode": 0, "mid": 2816, "body": {"length": 3, "rawData": "Zmdo"}}}]}
cloudRsp output:0287E95A6D646F
cloudRsp output:AAAA0800
```

用户开发完毕后，可以右键工程名，选择 Run As -> Maven clean，然后再右键工程名 Run As -> Maven install 来生成对应的 jar 包，jar 包的生成位置在该工程的 target 目录下。

4.5.3 输入数据（码流，JSON 结构体）格式定义

用户可以通过登陆中国电信物联网开放平台开发者门户导入 profile 模板，注册对应的设备，添加 cig 插件，绑定设备后通过模拟器模拟设备向平台上报数据。

设备上报的数据为 16 进制码流，码流再上报到平台前 cig 会将其转换为字节数组并通过调用编解码插件的 decode 方法对其解码形成 JSON 格式的结构体，一个字节为 8 个 bit 位，16 进制的码流每两位对应一个字节也就是 8 个 bit 位，码流的前两位也就是第一个字节代表消息类型，如果为 00，则代表该条消息为设备上报的数据；如果为 01，则代表设备对平台命令的响应。

(1) 设备上报数据

智能家电产业 NB-IoT 应用及开发指导书

如果是设备上报数据的话，消息类型字段加上需要解析的字段一共 20 个字节也就是 40 位，所以设备上报数据的码流至少要保证有 40 位，否则平台无法接收到消息，可以参照下表。

字节数	1	2	3	4	5	6	7	8	9	10	11	12	13	14
名称	msgType	signalPower	totalPower	cellId				signalEcl	signalSnr	signalPci				
定义	消息类型	信号强度	总信号强度	小区号				信号覆盖等级	信噪比	小区 PCI				
字节数	15	16	17	18			19	20	...					
名称	linkQuality	signalIndicator	batteryVoltage	length		rawData								
定义	连接质量	接收的信号强度指示	电压	指定透传数据长度		需要透传的数据								

数据上报的码流格式如下所示，最后的 666768 为 rawdata，需要透传的数据，不做解析：

0000020002000000040100020002010002010003666768

一个字节的最高位为符号位，无论上报的数据值是正数还是负数，都要通过数据值的二进制补码来转换成对应的 16 进制码流，正数的原码和补码是相同的，负数的补码等于原码的符号位不变，其余各位取反再加一。例如-1 的二进制原码为 10000001，补码为 11111111，对应的 16 进制码流就为 FF。上报的 16 进制码流经过编解码插件的处理会给应用端返回一个 ObjectNode，ObjectNode 中数据的封装如下代码所示：

```

public ObjectNode toJsonNode() {
    try {
        //组装body体
        ObjectMapper mapper = new ObjectMapper();
        ObjectNode root = mapper.createObjectNode();

        // root.put("identifier", this.identifier);
        root.put("msgType", this.msgType);

        //根据msgType字段组装消息体
        if (this.msgType.equals("deviceReq")) {
            ArrayNode arrynode = mapper.createArrayNode();

            //serviceId=Connectivity 数据组装
            ObjectNode ConnectivityNode = mapper.createObjectNode();
            ConnectivityNode.put("serviceId", "Connectivity");
            ObjectNode ConnectivityData = mapper.createObjectNode();
            ConnectivityData.put("signalPower", this.signalPower);
            ConnectivityData.put("totalPower", this.totalPower);
            ConnectivityData.put("cellId", this.cellId);
            ConnectivityData.put("signalEcl", this.signalEcl);
            ConnectivityData.put("signalSnr", this.signalSnr);
            ConnectivityData.put("signalPci", this.signalPci);
            ConnectivityData.put("linkQuality", this.linkQuality);
            ConnectivityData.put("signalIndicator", this.signalIndicator);
            ConnectivityData.put("batteryVoltage", this.batteryVoltage);
            ConnectivityNode.put("serviceData", ConnectivityData);
            arrynode.add(ConnectivityNode);

            //serviceId=Transmission 数据组装
            ObjectNode TransmissionNode = mapper.createObjectNode();
            TransmissionNode.put("serviceId", "Transmission");
            ObjectNode TransmissionData = mapper.createObjectNode();
            TransmissionData.put("length", this.length);
            TransmissionData.put("rawData", this.rawData);
            TransmissionNode.put("serviceData", TransmissionData);
            arrynode.add(TransmissionNode);

            root.put("data", arrynode);
        } else {
            root.put("errcode", this.errcode);
            root.put("mid", this.mid); //mid

            //组装body体, 只能为ObjectNode对象
            ObjectNode body = mapper.createObjectNode();
            body.put("length", this.length);
            body.put("rawData", this.rawData);
            root.put("body", body);
        }
    }
    return root;
}

```


由于 rawdata 没有进行解析，所以数据传递到应用端需要用户自己解析，解析的代码如下所示：

```
byte[] data = objectNode.get("data").get(1).get("serviceData").get("rawData").binaryValue();
String s = new String(data,0,data.length);
System.out.println(s);
```

通过以上代码就可以将未解析的数据转换为字符串。输入以上码流通过编解码插件处理后所返回的结果如下，rawdata 没有解析，为一个 base64 格式的字符串：

```
{
  "msgType": "deviceReq",
  "data": [
    {
      "serviceId": "Connectivity",
      "serviceData": {
        "signalPower": 2,
        "totalPower": 2,
        "cellId": 4,
        "signalEcl": 1,
        "signalSnr": 2,
        "signalPci": 2,
        "linkQuality": 1,
        "signalIndicator": 2,
        "batteryVoltage": 1
      }
    },
    {
      "serviceId": "Transmission",
      "serviceData": {
        "length": 3,

```

```

        "rawData": "Zmdo"
    }
}
]
}

```

(2) 平台对上报数据的响应

平台对上报数据的响应是固定格式的，会给设备返回 4 个字节长度的 16 进制码流。如下为平台下发给设备的响应数据所对应的 16 进制码流，对数据的封装和解析同上：

```
cloudRsp output:AAAA0000
```

(3) 命令下发

命令下发时平台向设备发送的是一个 ObjectNode，并将 ObjectNode 经过编解码插件处理获得对应的 16 进制码流下发给设备。下发的数据是一个字符串，并以键值对的形式封装到 ObjectNode 中，并且必须要传递 mid 号。具体实现如下所示：

```

private static ObjectNode initCloudReqObjectNode() throws IOException {
    ObjectMapper mapper = new ObjectMapper();
    ObjectNode cloudReqObjectNode = mapper.createObjectNode();
    ObjectNode paras = mapper.createObjectNode();
    paras.put("rawData", "Zmdo");
    cloudReqObjectNode.put("identifier", "123");
    cloudReqObjectNode.put("msgType", "cloudReq");
    cloudReqObjectNode.put("cmd", "SET_CONNECTIVITY_STATE");
    cloudReqObjectNode.put("paras", paras);
    cloudReqObjectNode.put("mid", 2016);
    return cloudReqObjectNode;
}

```

下发的命令格式如下：

```

{
    "msgType": "cloudReq",
    "serviceId": "Connectivity",
    "cmd": "SET_CONNECTIVITY_STATE",

```

```
"paras":{"rawData":"Zmdo"},
  "mid": 2016,
  "hasMore": 0
}
```

以上传递的数据为字符串“Zmdo”，如下为经过编解码插件的 encode 方法对 ObjectNode 数据进行处理所形成的 16 进制码流，02 为消息类型，代表该消息为命令下发，07E0 为 mid 号所对应的 16 进制码流，5A6D646F 为下发的字符串数据所对应的 16 进制码流：

cloudReq output:0207E05A6D646F

(4) 设备对平台命令的响应

如果是设备对平台命令的响应，第一个字节为消息类型，第二个字节为 errcode 状态码，当 errcode 为 00 时，表示命令发送成功；当 errcode 为 01 时，表示命令发送失败。第三和第四个字节为 mid 号值，第五和第六个字节为 length 值，用来指定需要透传数据的长度，余下的字节为 rawdata，需要透传的数据，不做解析。mid 号用来匹配命令和响应数据，平台下发命令的时候会带一个 mid 号，设备对命令响应时会根据该 mid 号来确定是对哪一条命令的响应。消息类型字段加上需要解析的字段一共 6 个字节，所以设备上报数据的码流至少要保证有 6 个字节，否则平台无法接收到消息，可以参照下表。

字节数	1	2	3	4	5	6	...
名称	msgType	errcode	mid		length		rawData
定义	消息类型	状态码	用于匹配命令与对应的响应		指定透传数据长度		需要透传的数据

命令响应的码流格式如下所示，最后的 666768 为 Rawdata，需要透传的数据，不做解析：010007E00003666768

对于 rawdata 数据的解析同上。输入以上码流通过编解码插件处理后所返回的结果如下，rawdata 没有解析，为一个 base64 格式的字符串：

```
{
  "msgType": "deviceRsp",
  "errcode": 0,
  "mid": 2016,
  "body": {
    "length": 3,
    "rawData": "Zmdo"
  }
}
```

