

设备接入

最佳实践

文档版本 1.0
发布日期 2022-08-30



版权所有 © 华为云计算技术有限公司 2024。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HUAWEI和其他华为商标均为华为技术有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受华为云计算技术有限公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，华为云计算技术有限公司对本文档内容不做任何明示或暗示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

目录

1 概述	1
2 设备接入	6
2.1 在线开发 MQTT 协议的模拟智慧路灯	6
2.2 设备模拟器快速接入	23
2.3 基于 NB-IoT 小熊派开发智慧路灯	31
2.4 基于 NB-IoT 小熊派的智慧烟感	46
2.5 模拟 NB 设备智慧路灯的接入与调试	62
2.6 通过协议转换网关实现泛协议设备接入	86
2.7 基于 MQTT.fx 的 X.509 证书接入	89
2.8 基于 BearPi-HM_Nano 开发板和 OpenHarmony3.0 对接 IoTDA	108
2.9 基于 Jmeter 工具进行 MQTT 性能测试	116
3 设备管理	122
3.1 预置设备影子自动调节空调温度	122
3.2 使用自定义 Topic 管理室内空调	125
3.3 MQTT 协议设备 OTA 固件升级	129
3.4 设备自定义 TOPIC 迁移上云	140
4 数据转发	151
4.1 指标数据推送至 DMS Kafka 最佳实践	151
5 设备联动	154
5.1 设备触发告警并邮件或短信通知	154
5.2 温度过高时自动关闭设备	160
5.3 煤气浓度过高自动打开推窗器	163
5.4 监测设备状态变化情况并发送通知	169

1 概述

现在您应该对物联网平台有了初步的认识，那也许您会问，如何给您带来价值？哪些业务场景可以通过平台来实现？怎样接入到平台？接下来，通过最佳实践，我们把平台的业务流转串起来，把物模型等概念引进来，通过业务场景示例来呈现平台能力和价值，供您参考。

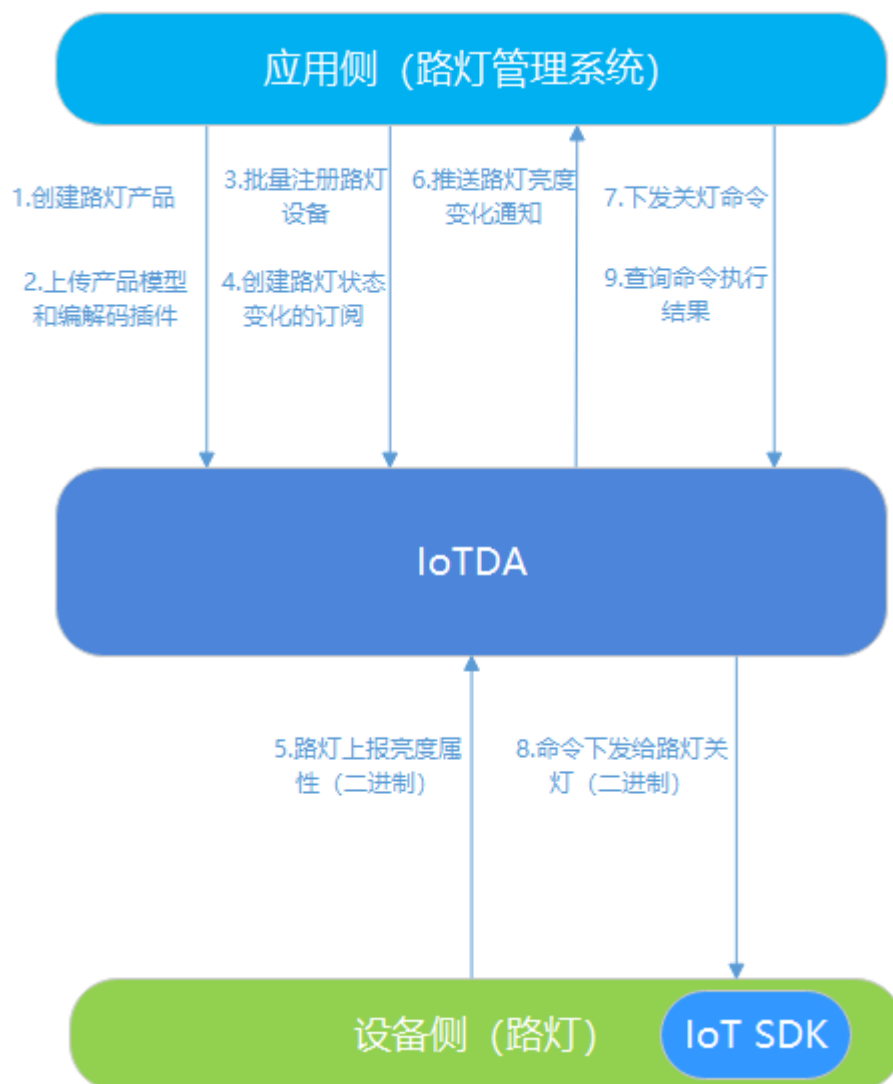
场景示例—智慧路灯

场景简介：路灯管理系统通过接入物联网平台，监控路灯状态，控制路灯开关（集成了NB模组的智慧路灯）。

该场景主要描述的是设备可以通过LWM2M协议与物联网平台进行交互，应用侧可以到物联网平台订阅设备侧变化的通知，也可以给设备侧下发命令。

核心知识点：产品模型、编解码插件、订阅推送、属性上报、命令下发。

最佳实践参考：[基于NB-IoT小熊派开发智慧路灯](#)。



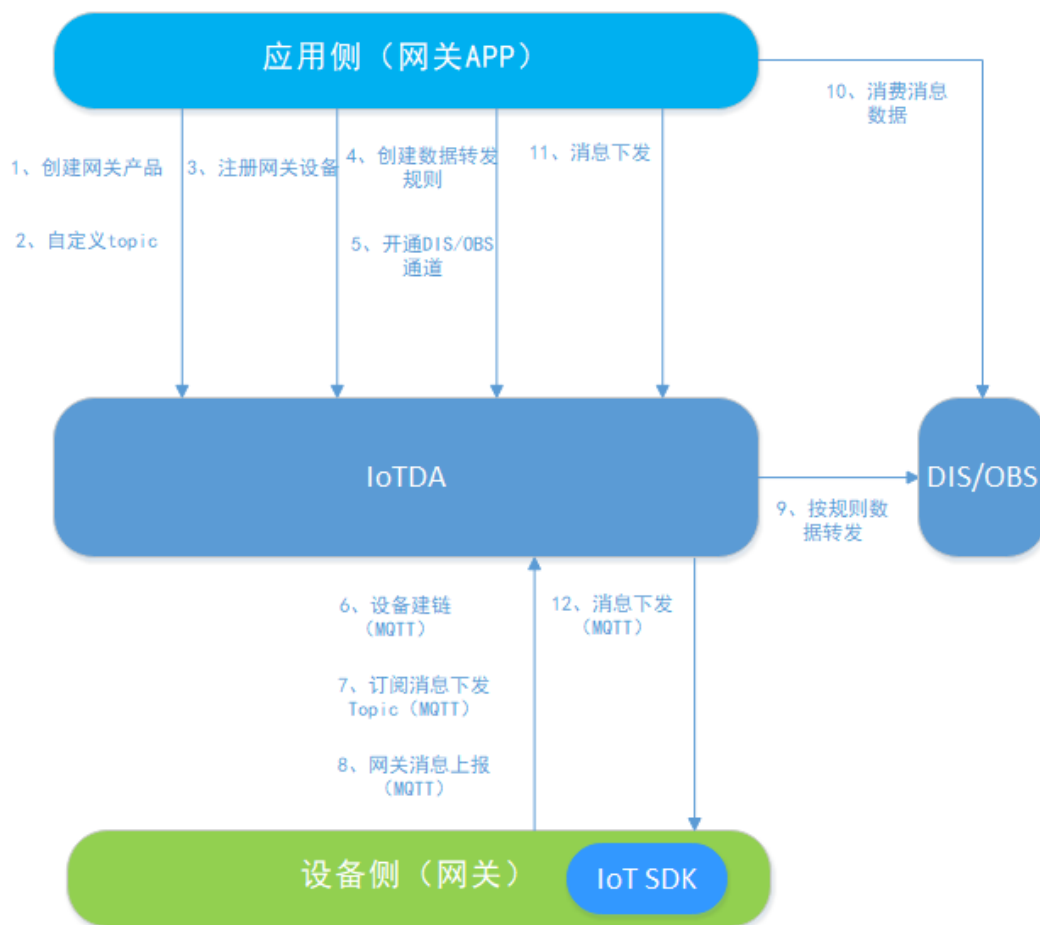
场景示例—智能网关

场景简介：已对接存量设备的网关，不用迁移就可以实现对既有设备和新设备的兼容控制。

该场景主要描述的是设备可以通过MQTT协议与物联网平台进行交互，用户可以在控制台产品详情中自定义Topic，通过应用侧接口或控制台创建数据转发规则，把设备上报的消息转发给其他华为云服务，供应用侧用户主动去消费。

核心知识点：产品模型、消息上报、消息下发、MQTT协议、数据转发规则、自定义Topic。

最佳实践参考：[使用自定义Topic管理室内空调](#)



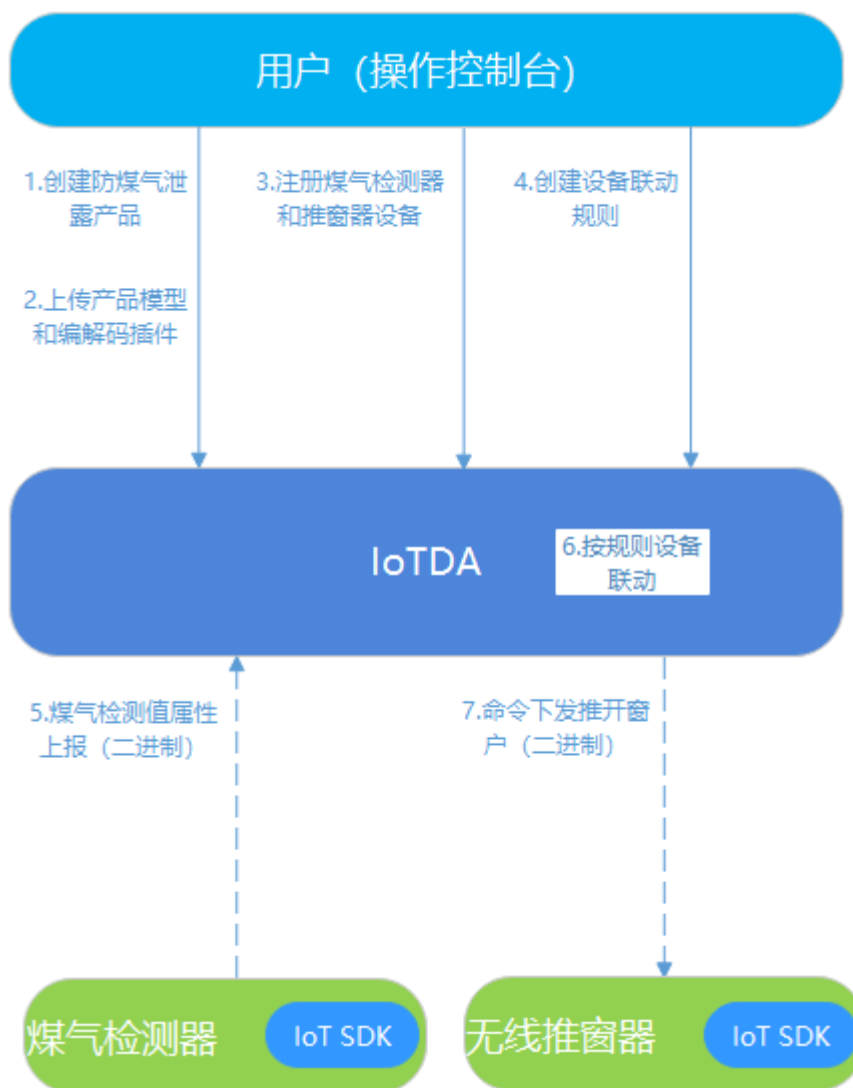
场景示例—智能家居煤气检测联动

场景简介：煤气检测器发现煤气浓度超标时，自动关联无线推窗器把窗户打开，起到通风的作用。

该场景主要描述的是设备可以通过MQTT协议与物联网平台进行交互，用户可以在控制台或通过应用侧接口创建设备联动规则，把设备上报的属性转发，通过物联网平台规则引擎转变成命令下发给其他指定设备。

核心知识点：产品模型、属性上报、命令下发、MQTT协议、设备联动规则。

最佳实践参考：[煤气浓度过高自动打开推窗器](#)。



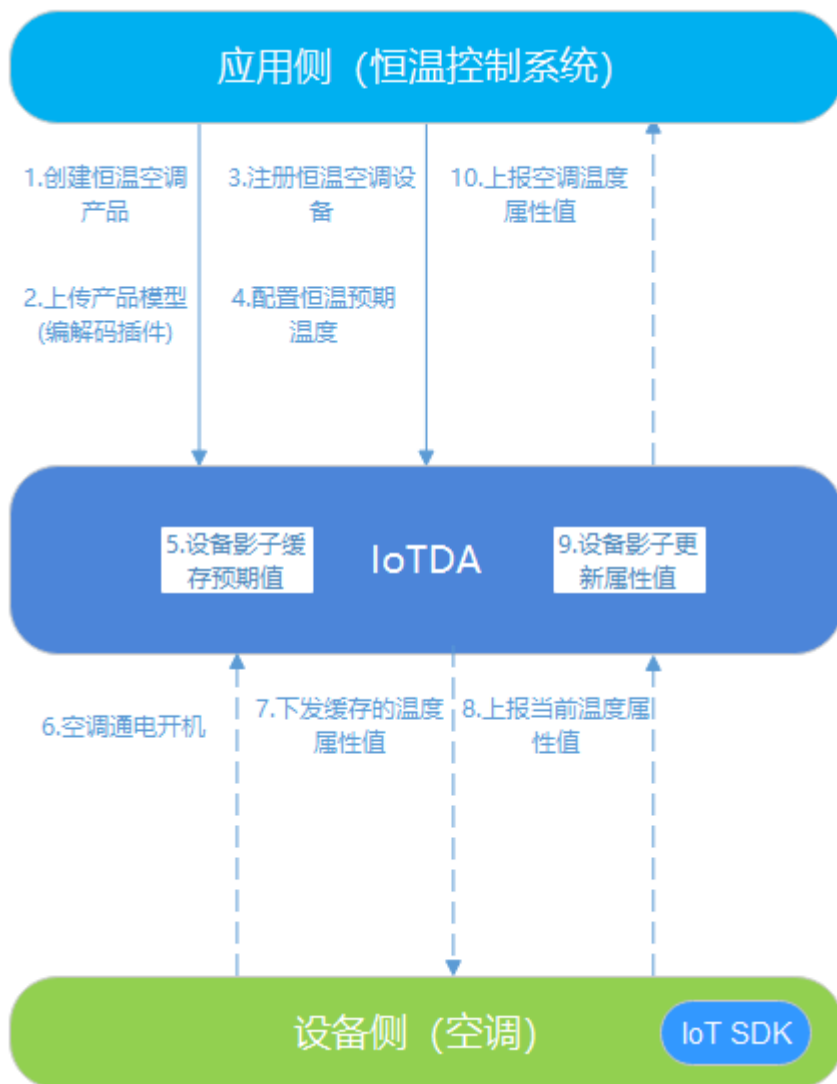
场景示例一恒温空调

场景简介：通过恒温控制系统，不论空调是否开机，都可以调整空调默认温度，待空调上电开机后，自动按默认温度调节。

该场景主要描述的是应用侧或控制台可以给离线设备下发缓存命令，待设备上线后执行，控制台通过设备影子配置属性，待设备上线后上报的属性不匹配的话，会自动下发属性修改的指令，直至修改成功。

核心知识点：产品模型、编辑码插件、设备影子、属性上报、属性修改。

最佳实践参考：[预置设备影子自动调节空调温度。](#)



2 设备接入

2.1 在线开发 MQTT 协议的模拟智慧路灯

场景说明

本文以“智慧路灯”为示例，通过设备模拟器替代真实的设备，带您快速体验设备上报数据到物联网平台和远程下发控制命令到设备的全过程。您可以选择MQTT.fx工具或MQTT_Simulator两种设备模拟器，推荐使用MQTT.fx工具进行模拟。

假设：路灯设备上报一条光照强度（luminance）消息；支持远程控制路灯开关状态的命令（switch），上报数据的格式为JSON格式。

前提条件

- 已注册华为云官方账号。未注册可单击[注册页面](#)完成注册。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，单击“立即使用”后开通该服务。

业务流程

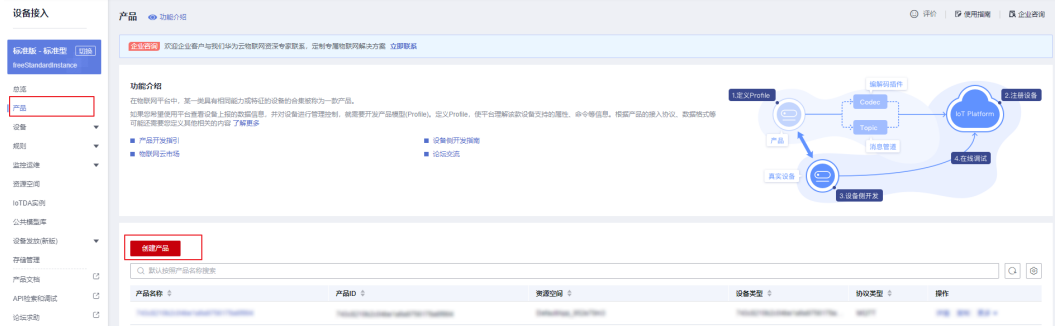
基于MQTT.fx体验平台功能是指使用MQTT.fx模拟器工具，进行数据上报、命令下发等业务的体验。

- 步骤1 创建产品。**创建一个MQTT协议的产品。
 - 步骤2 开发产品模型。**通过定义产品模型，在物联网平台构建一款路灯设备，支持上报光照强度、下发路灯开关状态命令。
 - 步骤3 注册设备。**注册一个MQTT协议的设备，体验数据上报业务。
 - 步骤4 连接鉴权。**使用MQTT.fx激活在物联网平台上注册的设备。
 - 步骤5 数据上报。**使用MQTT.fx向物联网平台上报数据。
 - 步骤6 命令下发。**在管理控制台下发命令，远程控制设备。
- 结束

创建产品

步骤1 登录**管理控制台**，单击左侧导航栏“产品”，单击页面左侧的“创建产品”。

图 2-1 产品-创建产品



步骤2 创建一个协议类型为MQTT协议、设备类型为StreetLamp的产品，参考页面提示填写参数后，单击“确定”。

图 2-2 创建产品-MQTT



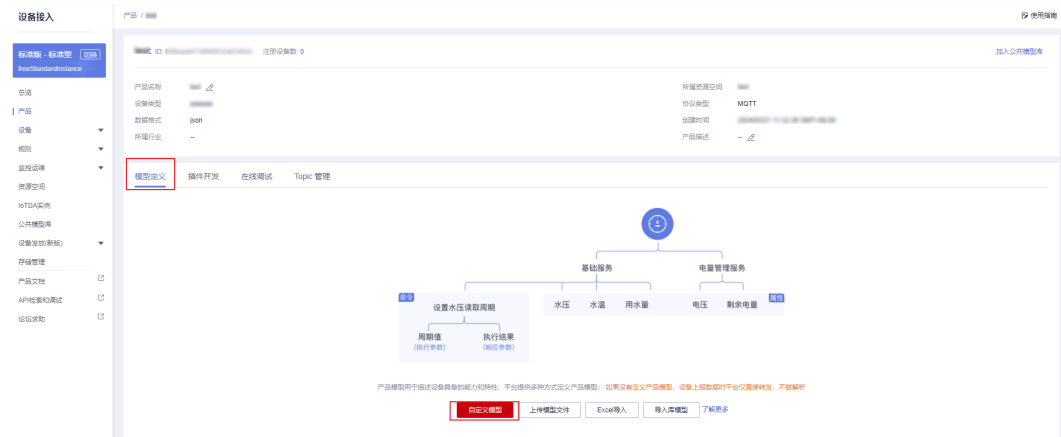
----结束

开发产品模型

步骤1 找到新增的产品，单击产品进入产品界面。

步骤2 在产品详情“模型定义”页面，单击“自定义模型”，配置产品的服务。

图 2-3 自定义模型-MQTT



步骤3 新增服务类型“BasicData”。

1. 在“添加服务”页面，根据页面提示填写“服务ID”、“服务类型”和“服务描述”，单击“确定”。

图 2-4 添加服务-BasicData

添加服务

* 服务ID

服务类型 ?

服务描述

6/128

确定

取消

2. 在“BasicData”服务列表右侧区域，单击“新增属性”，填写相关信息后，单击“确定”。

图 2-5 新增属性-luminance

×

新增属性

* 属性名称

属性描述 4/128

* 数据类型

* 访问权限

* 取值范围 -

步长

单位

步骤4 新增服务类型“LightControl”。

1. 在“模型定义”下单击“添加服务”，根据页面提示填写后，单击“确定”。
 - “服务ID”：LightControl
 - “服务类型”：建议和服务ID保持一致
 - “服务描述”：路灯开关控制
2. 在“LightControl”的下拉菜单下单击“添加命令”，输入命令名称“Switch”。

图 2-6 新增命令-Switch

新增命令
×

* 命令名称

下发参数

参数名称	数据类型	描述	操作
<p>暂无表格数据</p> <p>当前无下发参数数据，请先新增输入参数</p> <input style="width: 60px;" type="button" value="新增输入参数"/>			

5

总条数: 0

<

1

>

响应参数

参数名称	数据类型	描述	操作
<p>暂无表格数据</p> <p>当前无响应参数数据，请先新增响应参数</p> <input style="width: 60px;" type="button" value="新增响应参数"/>			

- 在“新增命令”页面，单击“新增输入参数”，填写相关信息后，单击“确定”。

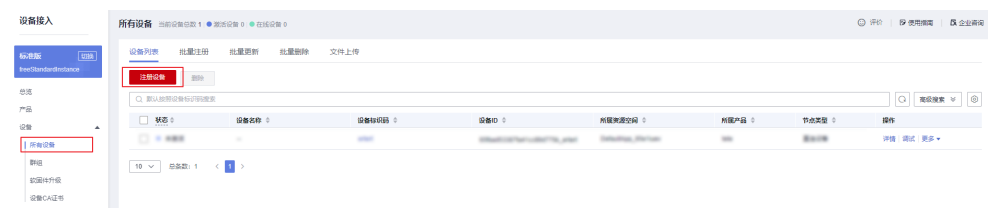
图 2-7 新增命令参数-value

----结束

注册设备

步骤1 在设备接入控制台页面，选择左侧导航栏“设备 > 所有设备”，单击页面右上角的“注册设备”。

图 2-8 所有设备-注册设备



步骤2 根据页面提示信息填写参数，然后单击“确定”。

参数名称	说明
所属资源空间	确保和所属产品归属在同一个资源空间。
所属产品	选择对应产品。
设备标识码	即nodeID，设备唯一物理标识。可自定义，由英文字母和数字组成。

参数名称	说明
设备名称	即device_name，可自定义。
设备认证类型	选择“密钥”。
密钥	此处如不填写，物联网平台会自动生成。

图 2-9 单设备注册-MQTT

×

单设备注册

* 所属资源空间 ?

* 所属产品

* 设备标识码 ?

设备ID ?

设备名称

设备描述 0/2,048

设备认证类型 ? 密钥 X.509证书

密钥

确认密钥

确定
取消

步骤3 成功注册设备后，平台会自动生成设备ID和密钥，请妥善保管好设备ID（deviceID）和密钥（deviceSecret），用于设备接入。

图 2-10 设备-注册设备成功



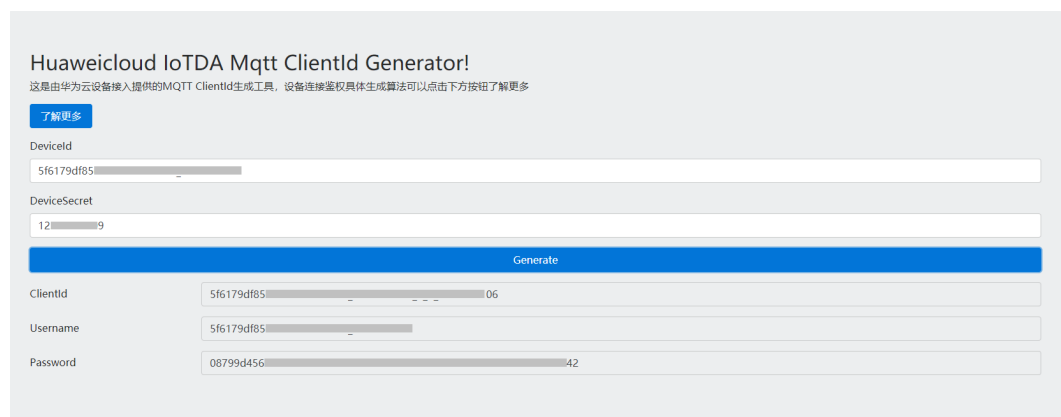
----结束

连接鉴权

使用MQTT.fx工具激活在物联网平台上注册的设备。

步骤1 下载**MQTT.fx**（默认是64位操作系统，如果是32位操作系统，单击此处下载**MQTT.fx**），安装MQTT.fx工具。

步骤2 访问[这里](#)，填写注册设备后生成的设备ID（DeviceId）和密钥（DeviceSecret），生成连接信息（ClientId、Username、Password）。



参数	必选/ 可选	类型	参数描述
ClientId	必选	String(256)	<p>一机一密的设备clientId由4个部分组成：设备ID、设备身份标识类型、密码签名类型、时间戳，通过下划线“_”分隔。</p> <ul style="list-style-type: none"> ● 设备ID：指设备在平台成功注册后生成的唯一设备标识，通常由设备的产品ID和设备的NodeId通过分隔符“_”拼装而来。 ● 设备身份标识类型：固定值为0，表示设备ID。 ● 密码签名类型：长度1字节，当前支持2种类型： <ul style="list-style-type: none"> - “0”代表HMACSHA256不校验时间戳。 - “1”代表HMACSHA256校验时间戳。 ● 时间戳：为设备连接平台时的UTC时间，格式为YYYYMMDDHH，如UTC时间2018/7/24 17:56:20 则应表示为2018072417。
Username	必选	String(256)	设备ID。
Password	必选	String(256)	<p>加密后的设备密钥。Password的值为使用“HMACSHA256”算法以时间戳为密钥，对secret进行加密后的值。</p> <p>secret为注册设备时平台返回的secret。</p>

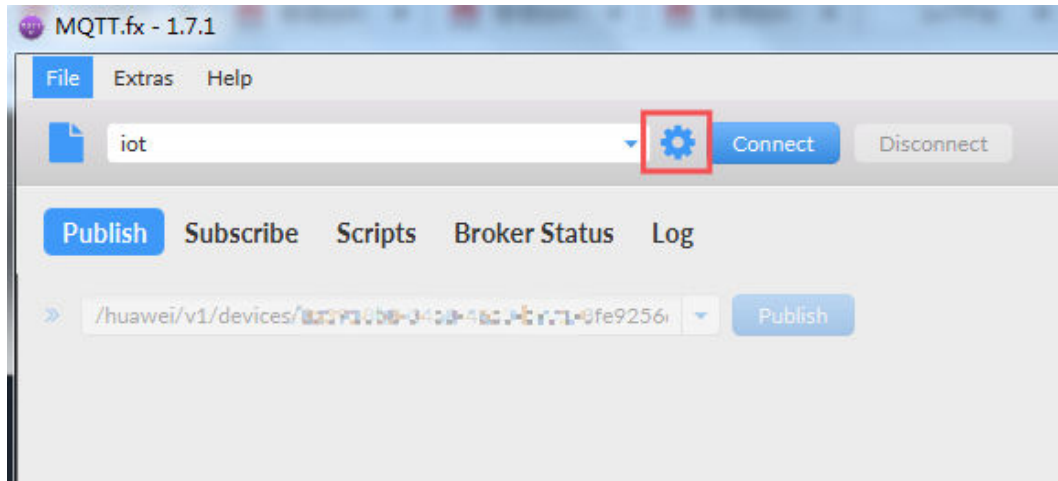
设备通过MQTT协议的connect消息进行鉴权，对于构造clientId的各个部分信息都必须包括进去，平台收到connect消息时，会判断设备的鉴权类型和密码摘要算法。

使用生成工具生成的clientId格式，默认不校验时间戳：设备ID_0_0_时间戳。

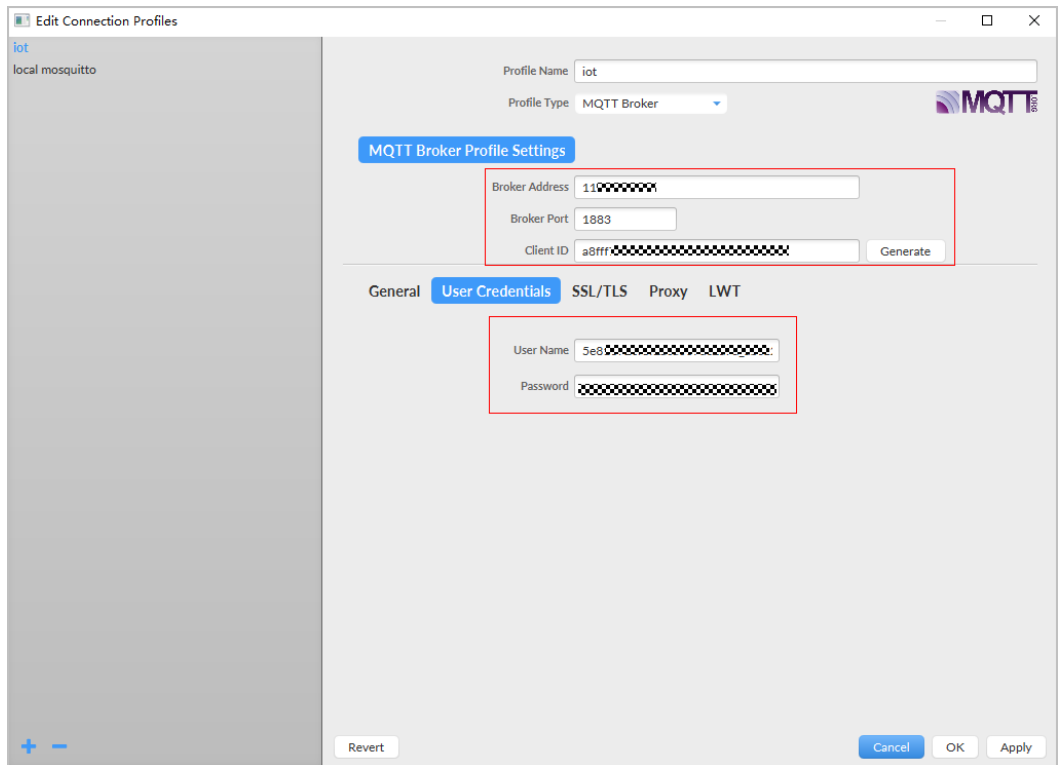
- 当采用“HMACSHA256”校验时间戳方式时，会先校验消息时间戳与平台时间是否一致，再判断密码是否正确。
- 当采用“HMACSHA256”不校验时间戳方式时，鉴权消息也必须带时间戳，但不检验时间是否准确，仅判断密码是否正确。

connect消息鉴权失败时，平台会返回错误，并自动断开MQTT链路。

步骤3 打开MQTT.fx软件，单击设置图标。



步骤4 参考下表配置鉴权参数，然后单击“Apply”。



参数名称	说明
Broker Address	填写从设备接入服务控制台获取的 平台对接信息 ，此接入地址为域名信息。不能通过域名接入的设备，通过在cmd命令框中执行“ping 域名”获取IP地址，用IP地址接入平台。由于IP地址不固定，您需要将IP地址做成可配置项。
Broker Port	默认为1883。
Client ID	设备cliendID，请参考2中获取。
User Name	即设备ID，请参考2中获取。

参数名称	说明
Password	加密后的设备密钥，请参考2中获取。

步骤5 单击“Connect”，设备鉴权成功后，在物联网平台可以看到设备处于在线状态。

图 2-11 设备在线

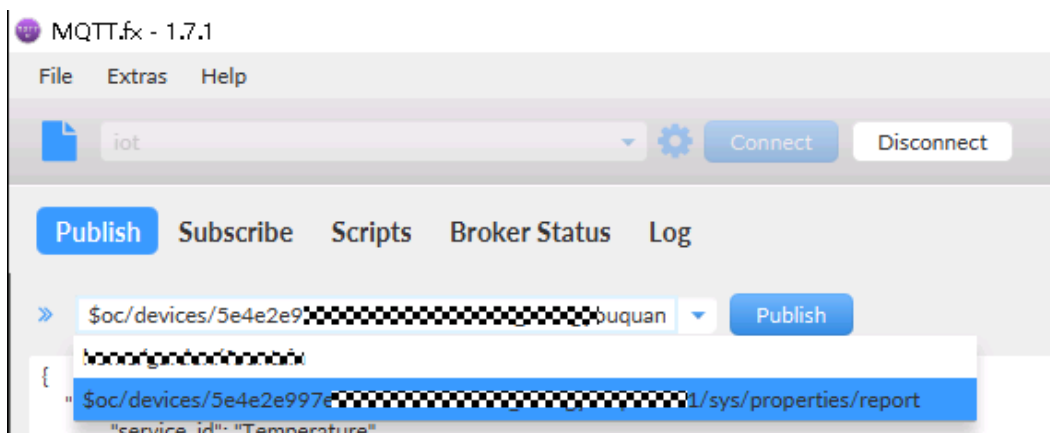


----结束

数据上报

使用MQTT.fx工具向物联网平台上报数据。设备若通过MQTT通道上报数据，需要发给指定的topic，上报消息的topic格式为：`$oc/devices/{device_id}/sys/properties/report`，其中对于一机一密设备，使用deviceId接入时需填写为设备注册成功后返回的deviceId值。

步骤1 填写接口地址，此处以`$oc/devices/{device_id}/sys/properties/report`为例，如：`$oc/devices/5e4e2e92ac-164aefa8fouquan1/sys/properties/report`。



步骤2 在工具中间的空白处填写上报的数据后，单击“Publish”。

表 2-1 设备服务数据列表

字段名	必选/可选	类型	参数描述
services	必选	List<ServiceProperty>	设备服务数据列表（具体结构参考下表ServiceProperty定义表）

表 2-2 ServiceProperty 定义表

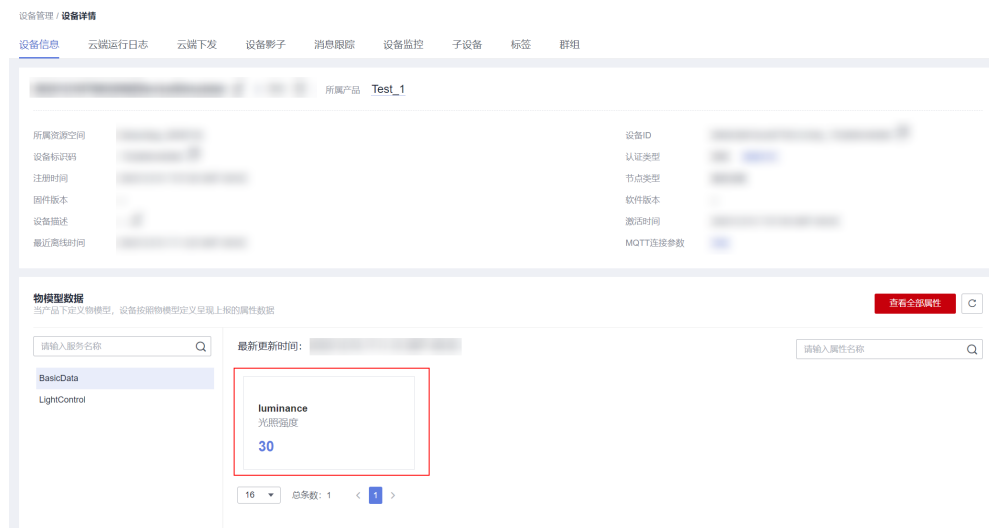
字段名	必选/可选	类型	参数描述
service_id	必选	String	设备服务的ID。
properties	必选	Object	设备服务的属性列表，具体字段在设备关联的产品模型中定义。
eventTime	可选	String	设备采集数据UTC时间（格式：yyyyMMddTHHmssZ），如：20161219T114920Z。 设备上报数据不带该参数或参数格式错误时，则数据上报时间以平台时间为准。

请求示例

```
{
  "services": [{
    "service_id": "BasicData",
    "properties": {
      "luminance": 30
    }
  }
]
```

步骤3 可以在设备详情页中查看设备是否成功上报数据。

图 2-12 查看上报数据-MQTT



----结束

命令下发

在管理控制台下发命令，远程控制设备。

步骤1 选择“设备 > 所有设备”，找到新建的设备，单击“详情”进入设备详情页面。

步骤2 单击“云端下发”页签，下发同步命令。

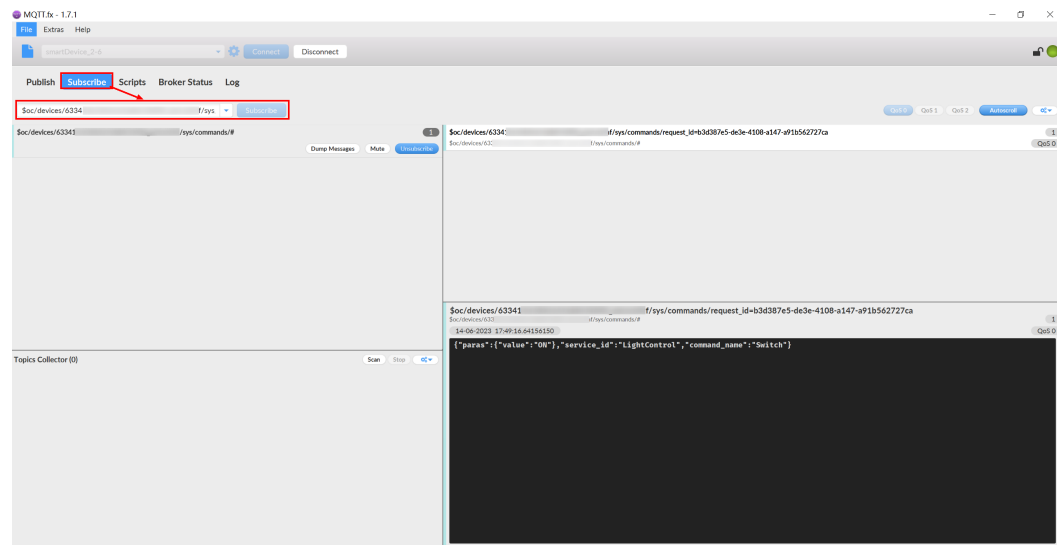
图 2-13 命令下发-MQTT



说明

MQTT协议设备仅支持同步命令下发，NB-IoT设备仅支持异步命令下发。

步骤3 MQTT.fx模拟器中选择“Subscribe”，输入命令下发Topic，订阅后则可以查看到下发的命令参数。



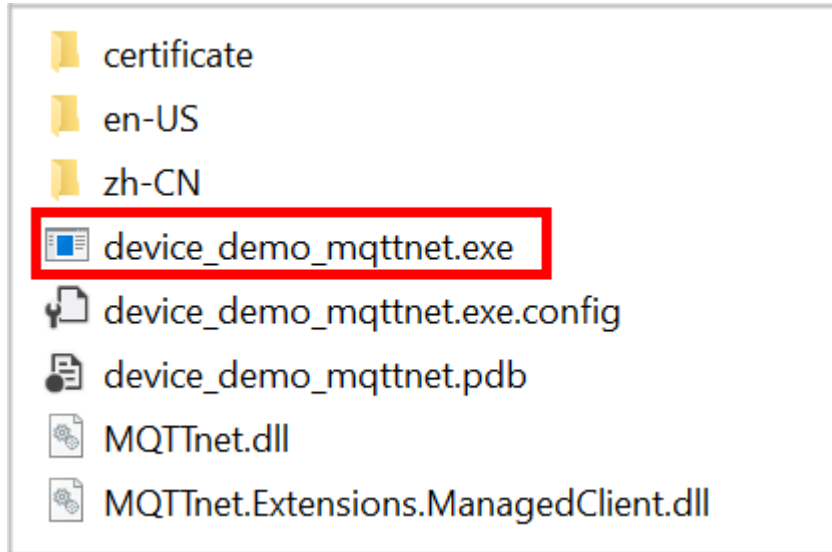
说明

- 通过MQTT.fx模拟器查看下发的命令参数，命令下发Topic格式为：\$oc/devices/{device_id}/sys/commands/#，其中{device_id}为设备注册成功后返回的deviceId值。
- 界面提示命令下发失败，是因为下发的同步命令需要设备侧及时响应，响应内容请参考[平台命令下发](#)。

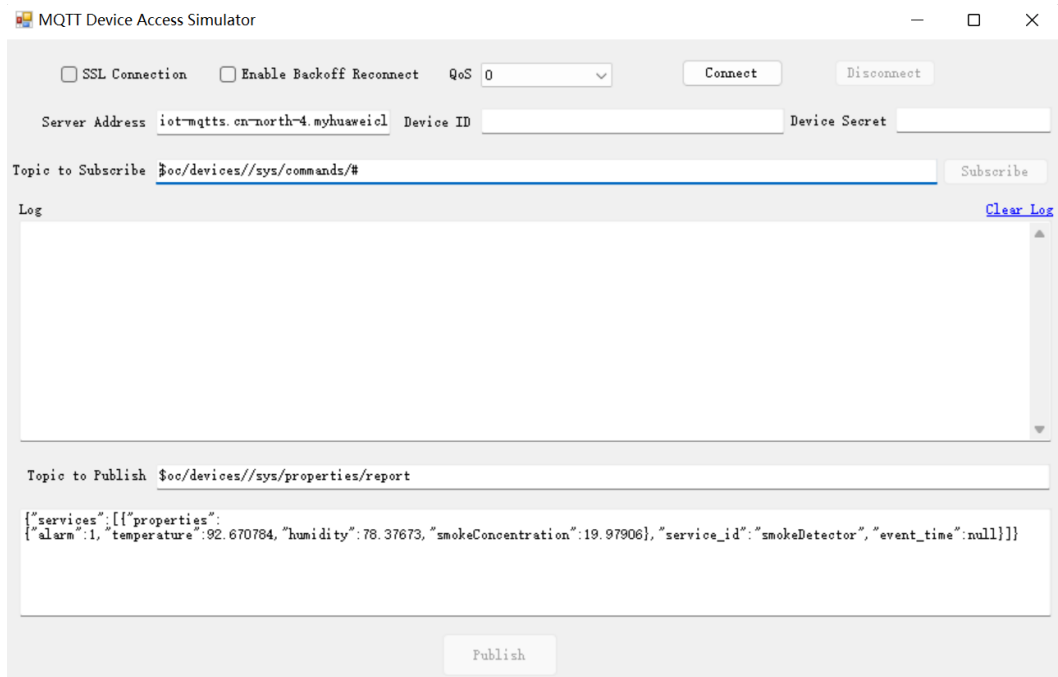
----结束

使用 MQTT_Simulator 接入

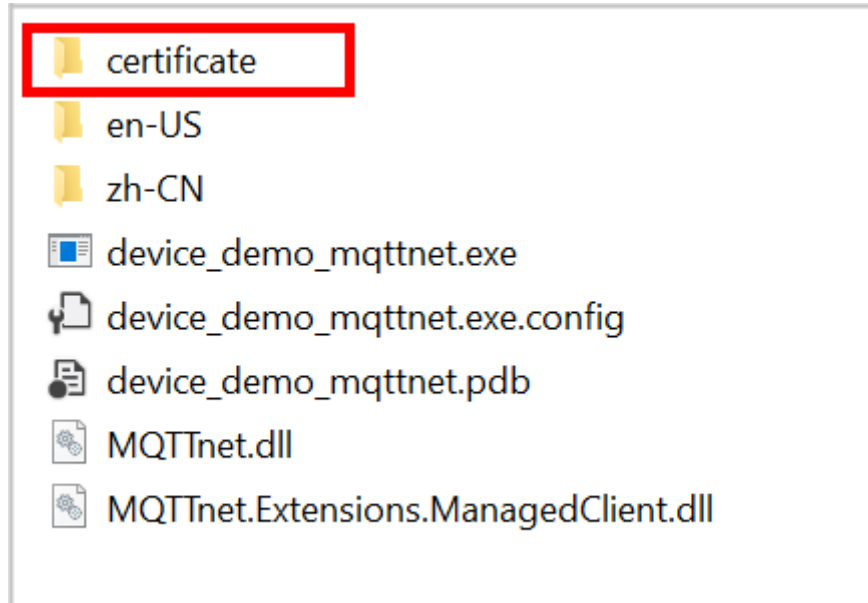
步骤1 下载[MQTT_Simulator模拟器](#)（默认是64位操作系统），并启动模拟器，如下图：



步骤2 界面展示



1. 模拟器提供了界面显示，填写Server地址、设备ID、设备密钥。请根据实际创建的设备信息填写。
 - Server地址：即域名，参考[平台对接信息](#)获取；
 - 设备ID和设备密钥：在[这里](#)获取。
2. 不同的Server地址SSL加密接入时要与对应的[证书文件匹配使用](#)，证书获取请参考[获取资源](#)，需要替换certificate文件夹下的证书，如下图：



3. 用户可以选择设备侧建链时是否为SSL加密，选择Qos方式是0还是1，当前不支持Qos2，可参考[使用限制](#)。

步骤3 新建连接

设备或网关在接入物联网平台时首先需要和平台建立连接，从而将设备或网关与平台进行关联。开发者通过传入设备信息，将设备或网关连接到物联网平台。单击新建连接按钮，域名、设备ID和密钥正确的情况下，可以看到日志设备连接成功，可在平台查看设备状态，如下图：

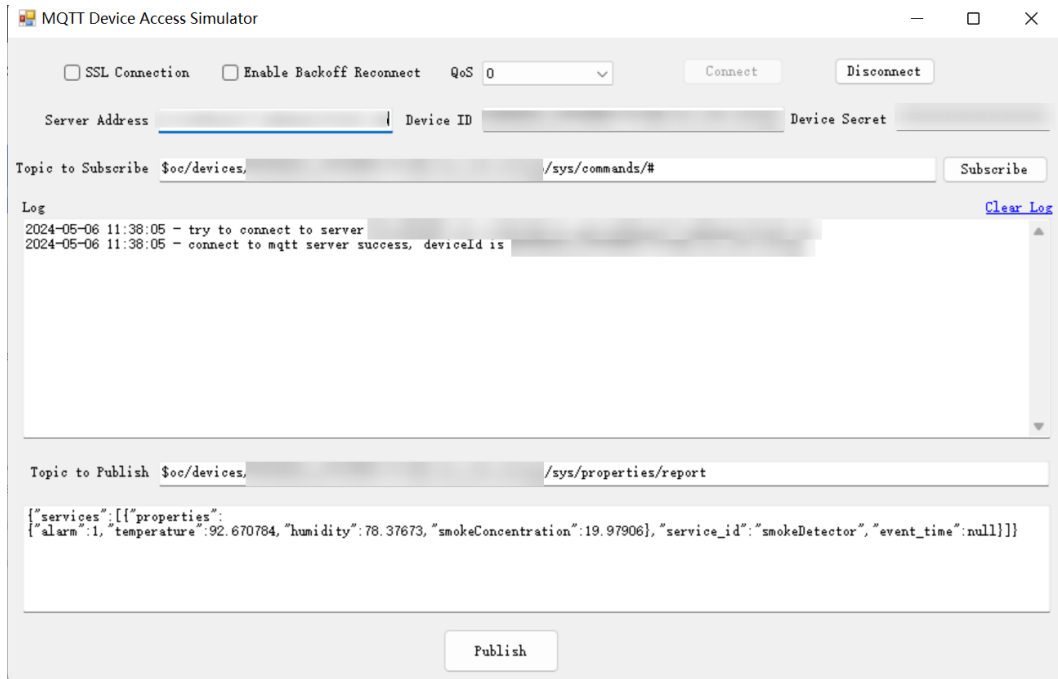
图 2-14 设备列表-设备在线



步骤4 订阅Topic

订阅某topic的设备才能接收broker发布的关于该topic的消息，关于平台预置topic可参考[Topic定义](#)。

建链后，如果成功订阅Topic，主界面日志栏显示如下信息：

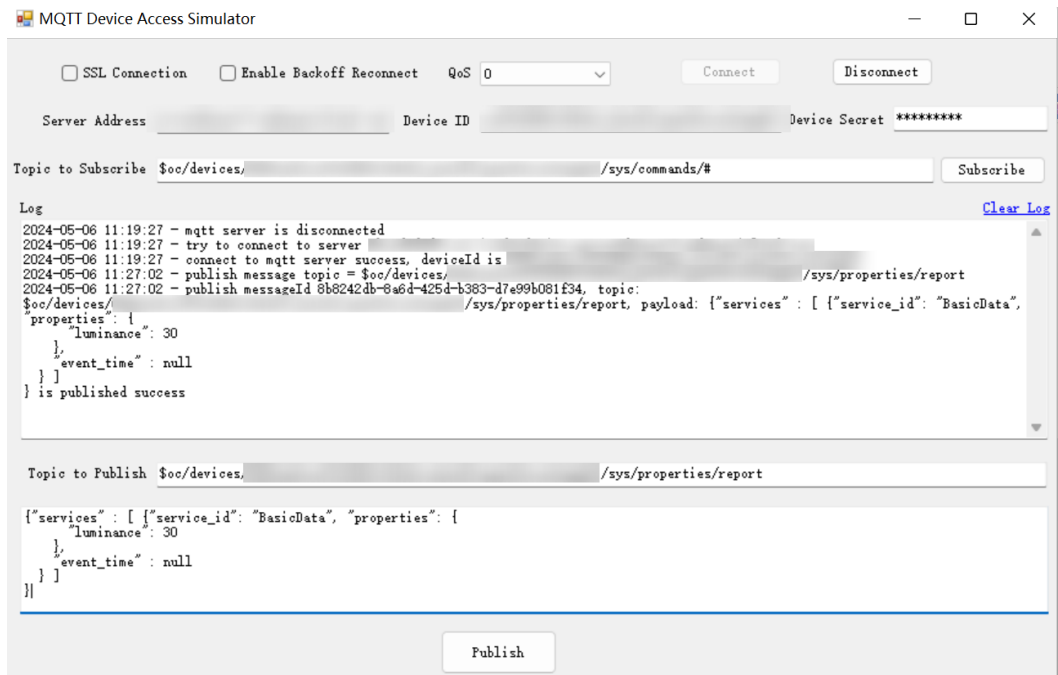


步骤5 发布Topic

发布Topic是指设备主动向平台上报自己的属性或消息，详细见[设备模拟器快速接入文档](#)。

在模拟器中实现了上报Topic、属性上报功能。

输入要上报的JSON message，上报luminance属性为30。发布Topic后，Demo界面显示如下：



设备上报属性成功后可在“设备详情”页面查看到上报的属性：

图 2-15 查看上报数据-MQTT



步骤6 接收下发命令

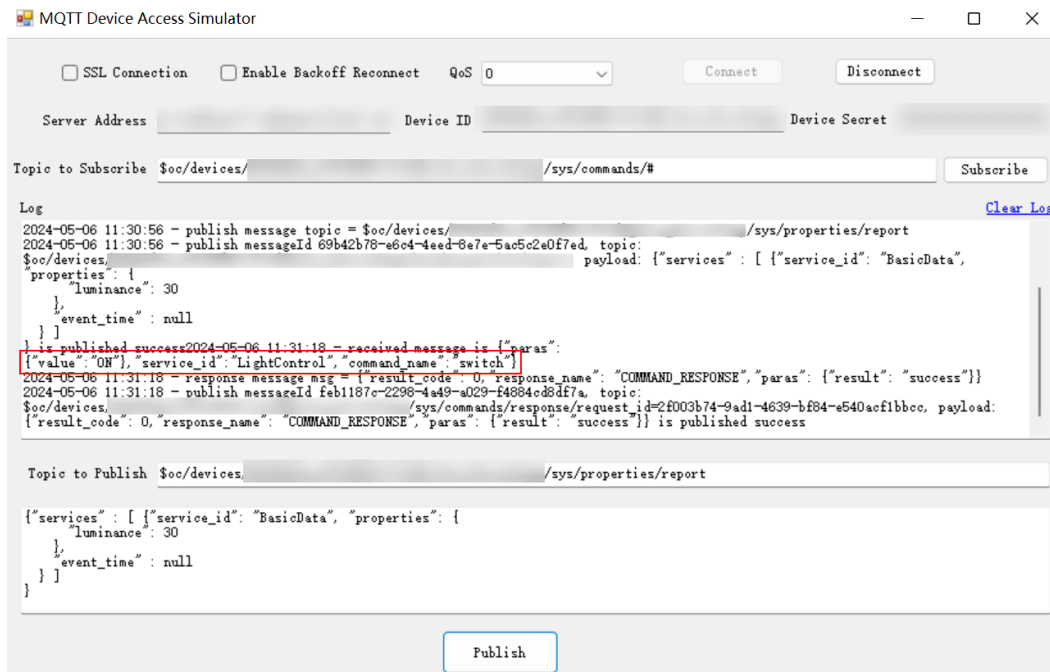
模拟器支持接收平台下发命令的功能，在MQTT建链完成并成功订阅Topic后，可以在[控制台](#)设备详情中对该设备进行同步命令下发。下发成功后，在MQTT的回调函数中接收到平台下发给设备的命令。

例如用户想远程开灯，下发参数名为LightControl: switch，参数值为ON的命令。

图 2-16 命令下发-同步命令下发



同步命令下发成功后，Demo界面显示如下：



----结束

进阶体验

按照本页面的指导，模拟MQTT设备接入物联网平台后，您应该已经基本了解MQTT设备调用物联网平台开放的接口与平台交互。

若您想要进一步体验设备接入服务，可参考[开发指南](#)开发真实应用和真实设备，并接入物联网平台，体验更多功能。

2.2 设备模拟器快速接入

本文档以设备接入模拟器为例，介绍以MQTT原生协议接入物联网平台。此模拟器是MQTT客户端，可以快速验证是否可以与物联网平台服务交互发布或订阅消息。

前提条件

- 已注册华为云官方账号。未注册可单击[注册页面](#)完成注册。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，单击“立即使用”后开通该服务。

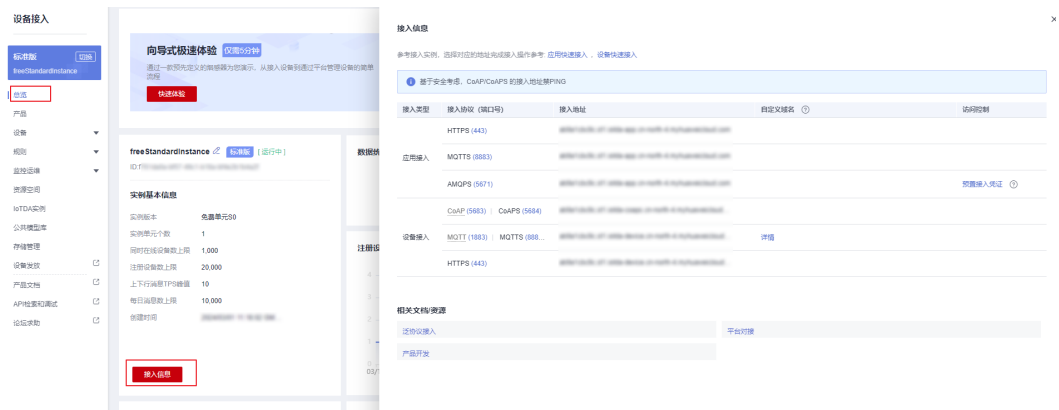
获取设备接入信息

在设备接入服务控制台获取设备接入平台所需的信息。

步骤1 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。

步骤2 单击“立即使用”进入控制台，单击左侧导航栏的“总览”，点击“实例基本信息 - 接入信息”查看设备接入信息，记录域名和端口。

图 2-17 总览-获取接入信息



说明

针对不支持用域名接入的设备，通过在cmd命令框中执行“ping 域名”获取IP地址，用IP地址接入平台。由于IP地址不固定，您需要将IP地址做成可配置项。

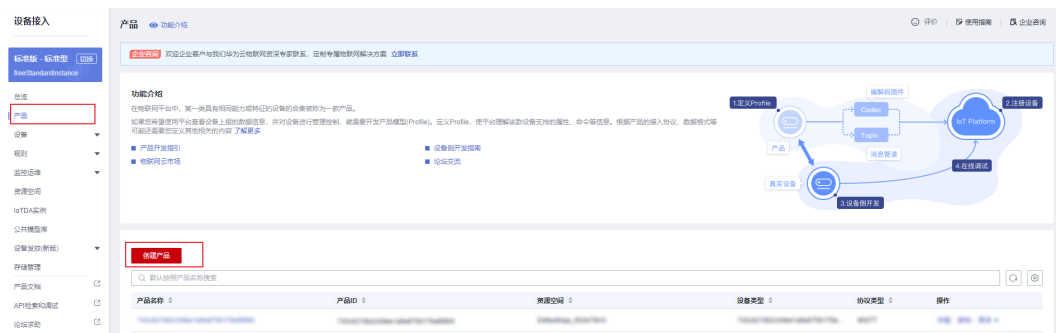
----结束

创建产品

步骤1 创建MQTT协议产品（如果已有MQTT协议产品，可跳过此步骤）。

步骤2 登录**管理控制台**，单击左侧导航栏“产品”，单击页面左侧的“创建产品”。

图 2-18 产品-创建产品



步骤3 创建一个协议类型为MQTT协议、设备类型为StreetLamp的产品，参考页面提示填写参数后，单击“确定”。

图 2-19 创建产品-MQTT

创建产品

* 所属资源空间 ?

如需创建新的资源空间，您可前往当前实例详情创建

* 产品名称

协议类型 ? MQTT

* 数据格式 ? JSON

设备类型选择 标准类型 自定义类型

* 设备类型 ?

高级配置 ▲ 定制ProductID | 备注信息

产品ID ?

产品描述

0/128

确定 取消

---结束

注册设备

步骤1 在设备接入控制台页面，选择左侧导航栏“设备 > 所有设备”，单击页面右上角的“注册设备”。

图 2-20 所有设备-注册设备



步骤2 根据页面提示信息填写参数，然后单击“确定”。

参数名称	说明
所属资源空间	确保和所属产品归属在同一个资源空间。
所属产品	选择对应产品。

参数名称	说明
设备标识码	即nodeID，设备唯一物理标识。可自定义，由英文字母和数字组成。
设备名称	即device_name，可自定义。
设备认证类型	选择“密钥”。
密钥	此处如不填写，物联网平台会自动生成。

图 2-21 单设备注册-MQTT

单设备注册
×

* 所属资源空间 ?

* 所属产品

* 设备标识码 ?

设备ID ?

设备名称

设备描述 0/2,048

设备认证类型 ? 密钥 X.509证书

密钥

确认密钥

确定
取消

步骤3 成功注册设备后，平台会自动生成设备ID和密钥，请妥善保管好设备ID（deviceID）和密钥（deviceSecret），用于设备接入。

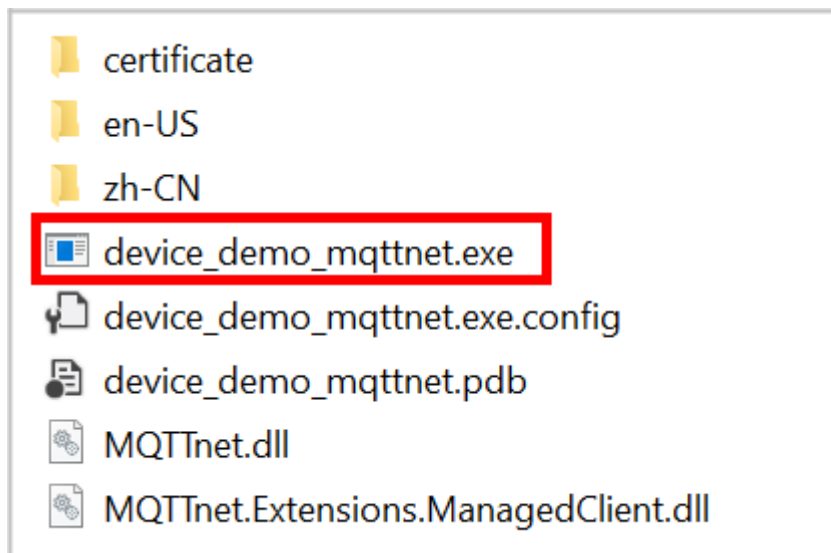
图 2-22 设备-注册设备成功



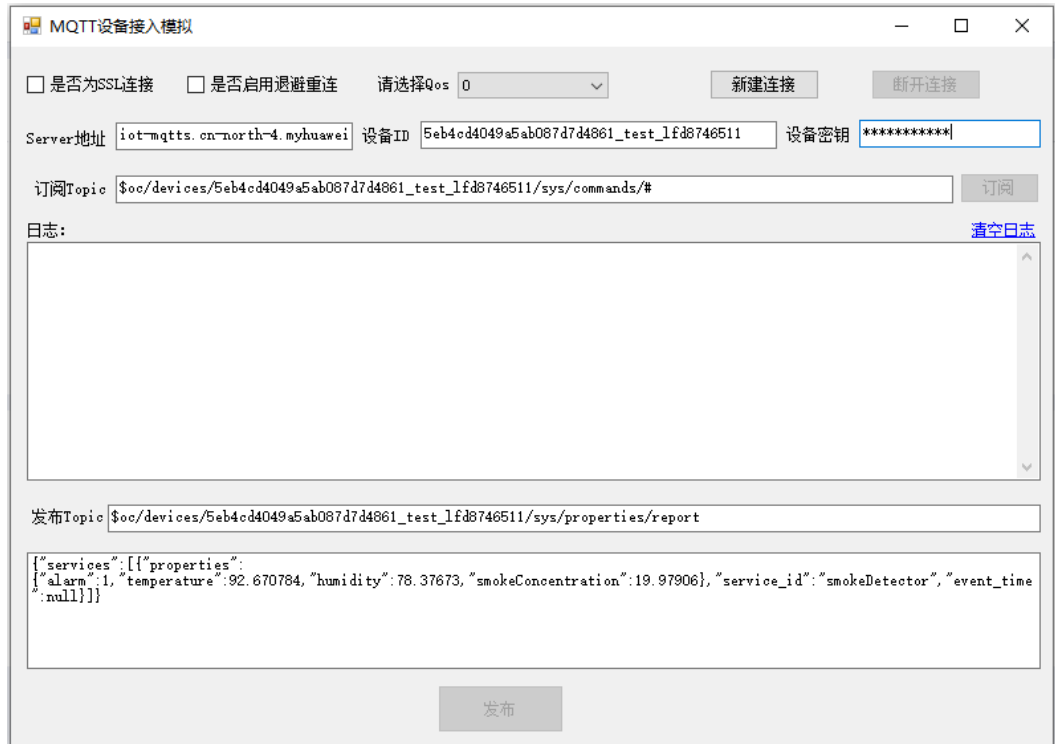
----结束

使用模拟器接入

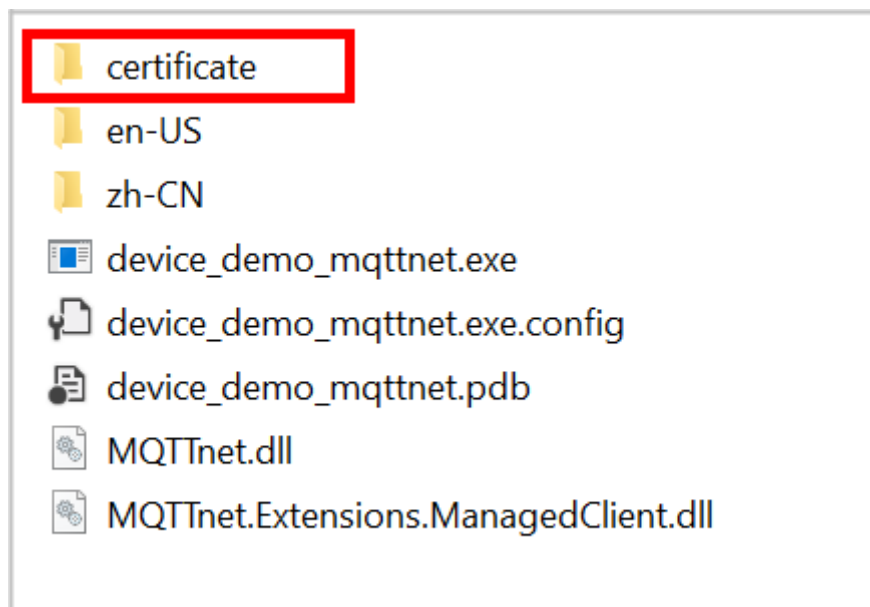
步骤1 下载**模拟器**（默认是64位操作系统），并启动模拟器，如下图：



步骤2 界面展示



1. 模拟器提供了界面显示，填写Server地址、设备ID、设备密钥。请根据实际创建的设备信息填写。
 - Server地址：即域名，参考[平台对接信息](#)获取；
 - 设备ID和设备密钥：在[这里](#)获取。
2. 不同的Server地址SSL加密接入时要与对应的[证书文件匹配使用](#)，证书获取请参考[获取资源](#)，需要替换certificate文件夹下的证书，如下图：

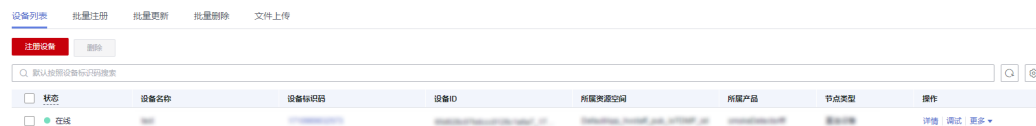


3. 用户可以选择设备侧建链时是否为SSL加密，选择Qos方式是0还是1，当前不支持Qos2，可参考[使用限制](#)。

步骤3 新建连接

设备或网关在接入物联网平台时首先需要和平台建立连接，从而将设备或网关与平台进行关联。开发者通过传入设备信息，将设备或网关连接到物联网平台。单击新建连接按钮，域名、设备ID和密钥正确的情况下，可以看到日志设备连接成功，可在平台查看设备状态，如下图：

图 2-23 设备列表-设备在线



步骤4 订阅Topic

订阅某topic的设备才能接收broker发布的关于该topic的消息，关于平台预置topic可参考[Topic定义](#)。

建链后，如果成功订阅Topic，主界面日志栏显示如下信息：



步骤5 发布Topic

发布Topic是指设备主动向平台上报自己的属性或消息，详细见[设备模拟器快速接入文档](#)。

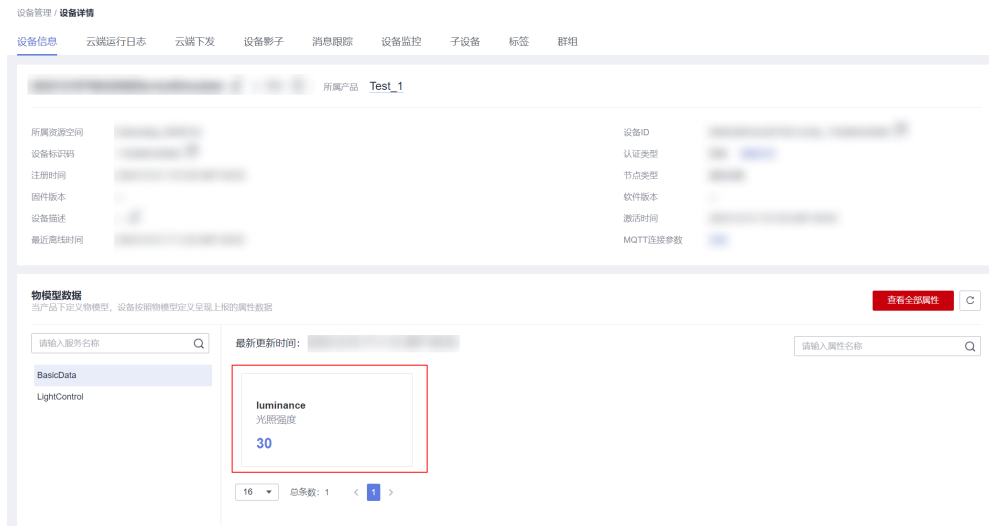
在模拟器中实现了上报Topic、属性上报功能。

发布Topic后，Demo界面显示如下：



设备上报属性成功后可在“设备详情”页面查看到上报的属性：

图 2-24 查看上报数据-MQTT

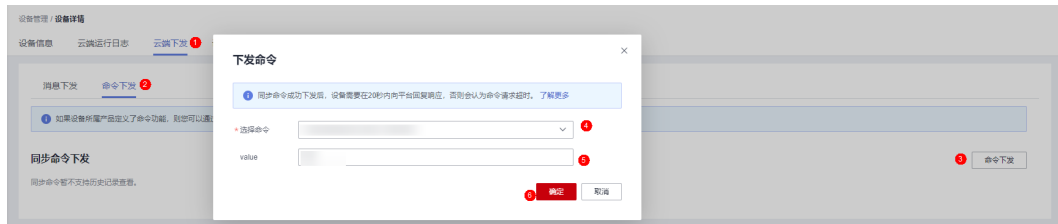


步骤6 接收下发命令

模拟器支持接收平台下发命令的功能，在MQTT建链完成并成功订阅Topic后，可以在**控制台**设备详情中对该设备进行同步命令下发。下发成功后，在MQTT的回调函数中接收到平台下发给设备的命令。

例如下发参数名为smokeDetector: SILENCE，参数值为50的命令。

图 2-25 命令下发-同步命令下发



同步命令下发成功后，Demo界面显示如下：



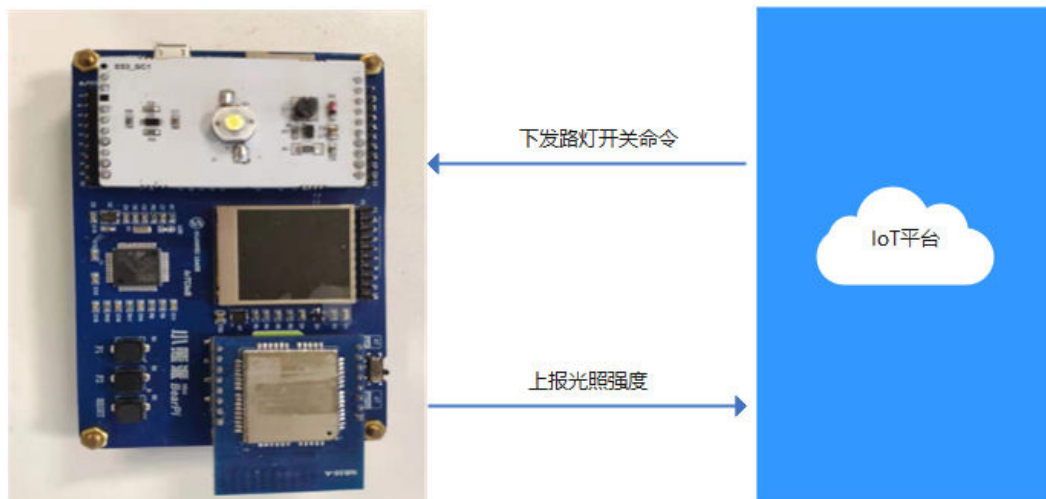
----结束

2.3 基于 NB-IoT 小熊派开发智慧路灯

场景说明

智慧路灯是城市智能化道路上重要一环，智慧路灯的实施具有节约公共照明能耗、减少因照明引起的交通事故等多种社会意义。路灯也是大家在日常生活中可以强烈感知到的公共设施，更易理解其智能化的场景。

在该文档中，我们基于华为一站式开发工具平台（VS code工具IoTlink插件），从设备（小熊派开发套件）、平台（IoTDA华为物联网平台）、端到端构建一款智慧路灯解决方案样例，带您体验十分钟快速上云。实现智慧路灯检测并上报光照强度，显示在IoTDA控制台，并在IoTDA控制台远程控制LED灯开关的功能。

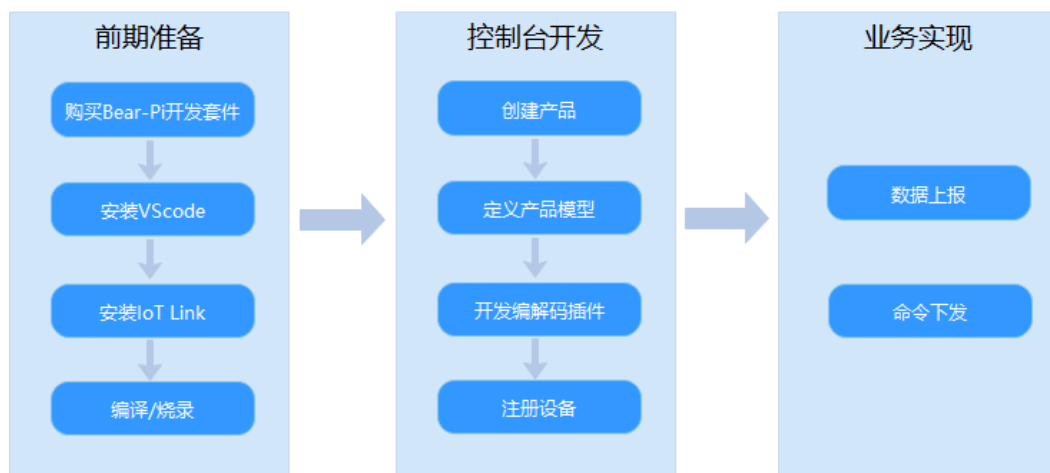


开发环境

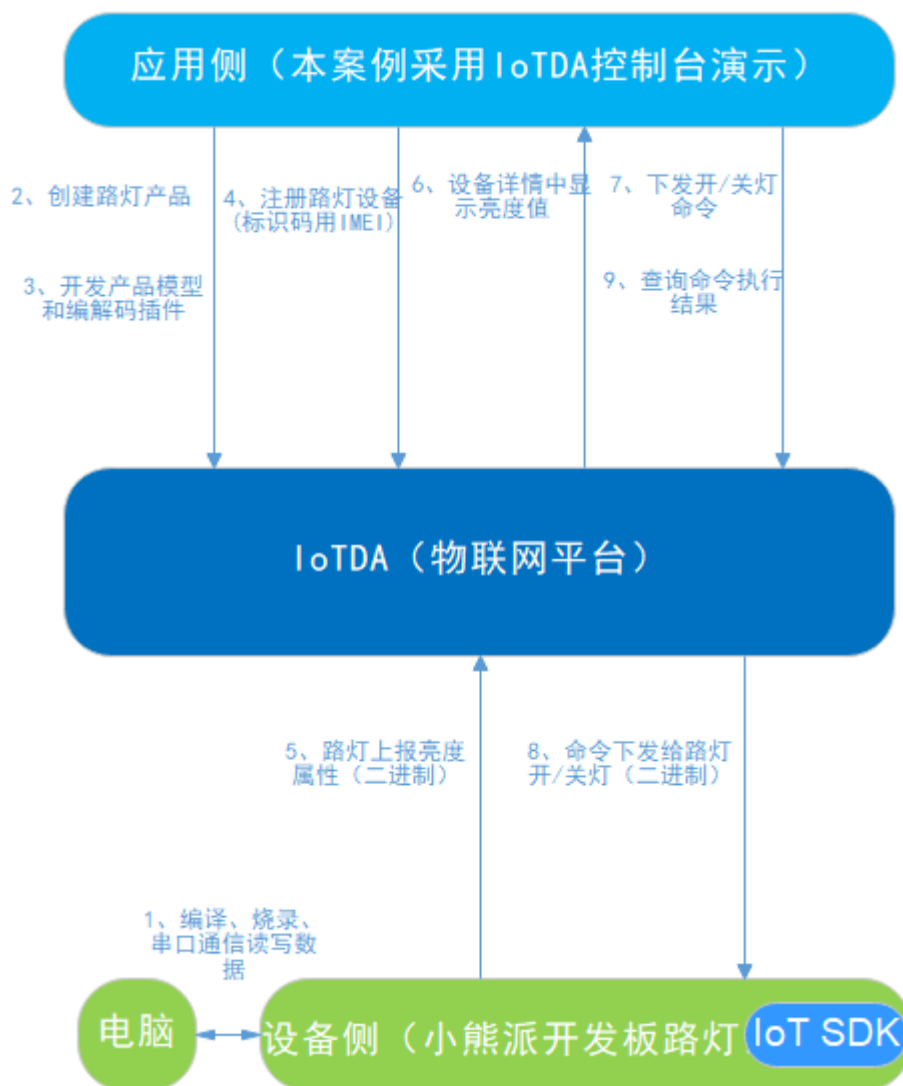
- 硬件：小熊派BearPi-LoT开发套件（包含NB卡、NB模组、智慧路灯功能模块、USB数据线等）。
- 软件：VScode, IoT Link插件、华为云（已开通**设备接入服务**）、win7及以上版本的64位操作系统（本文以windows 10 64-bit系统调试）。

整体流程

智慧路灯案例的端到端整体流程如下图所示。



该场景主要描述的是设备可以通过LWM2M协议（NB卡）与物联网平台进行交互，应用侧可以查看设备侧属性变化，也可以给设备侧下发命令。



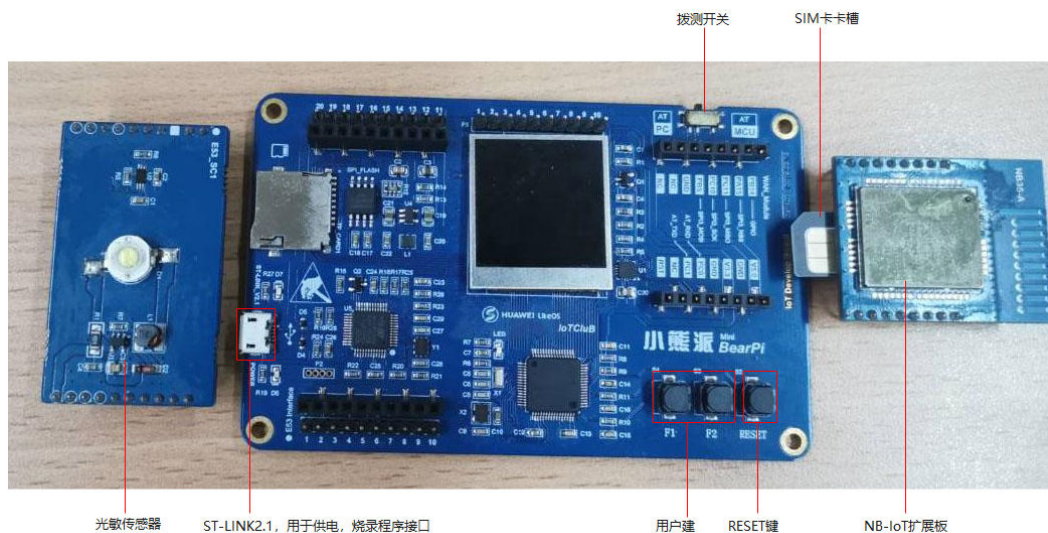
小熊派开发板简介

开发板在物联网系统架构中属于感知设备，该类设备通常由传感器、通信模组、芯片以及操作系统组成。为增加开发板的可扩展性，小熊派开发板没有采用传统的板载设计，而是使用了可更换传感器扩展板以及可更换通信模组扩展板设计，通信模组是数据传输的出入口，常用的通信模组包括NB-IoT, Wifi以及4G等，芯片是设备的主控设备，该开发板内置了一个低功耗的STM32L431单片机作为主控芯片，即MCU。操作系统使用的是华为的LiteOS操作系统，其提供了丰富的端云互通组件。

为了便于开发调试，如图1所示，该开发板板载了2.1版本的ST-Link，它具有在线调试烧录，拖拽下载以及虚拟串口等功能。开发板中间板载一块分辨率为240 * 240的LCD屏幕，其主要用于显示传感器数据以及调试日志。LCD屏幕下方是主控芯片。

开发板右上角具有一个拨码开关，将其拨至左侧AT-PC模式，通过电脑端的串口助手，发送AT指令调试通信模组。右侧AT-MCU模式，通过MCU发送AT指令与通信模组进行交互，将采集到的传感器数据通过通信模组发送到云端。

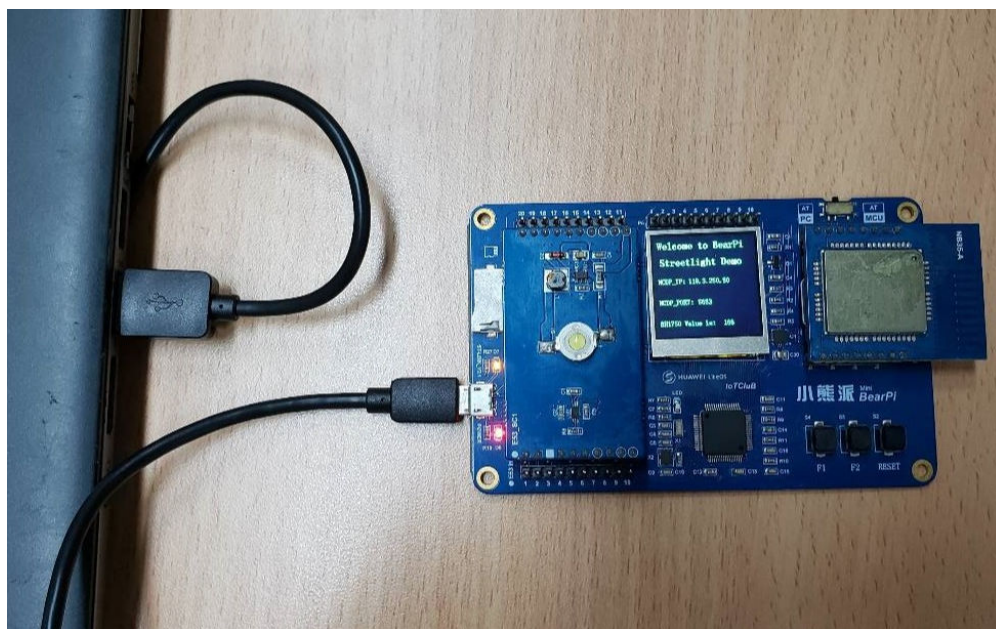
图 2-26 小熊派开发板



硬件连接

1. 将NB卡插入到NB-LoT扩展板的SIM卡卡槽，确保插卡的时候卡的缺口朝外插入，如图2所示。
2. 将光敏传感器以及NB-LoT扩展板插入到开发板上，注意安装方向，然后用USB数据线将小熊派开发板与电脑连接起来。显示屏有显示，电源灯被点亮，说明开发板通电成功。

图 2-27 硬件连接



安装 IoT Link Studio 插件

IoT Link Studio是针对IoT物联网端侧开发的IDE环境，提供了编译、烧录、调试等一站式开发体验，支持 C、C++、汇编等多种开发语言，让您快速、高效地进行物联网开发。

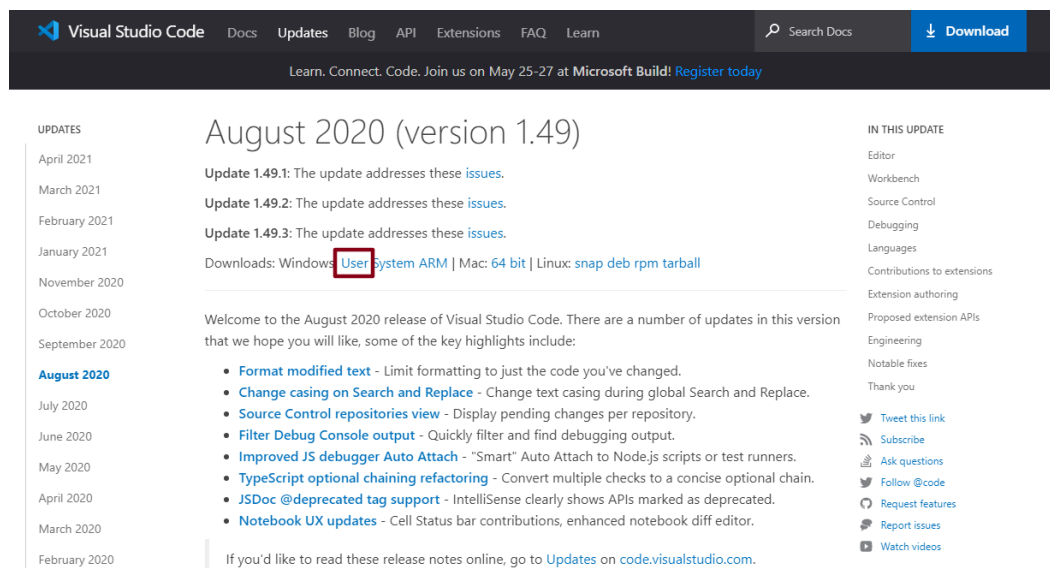
步骤1 以win10为例，获取电脑的系统配置：在运行窗口中输入pc，然后单击“属性”，查看系统配置，因为后续安装软件要选择操作系统。

图 2-28 获取系统配置



步骤2 访问 [这里](#)，根据您电脑系统配置，下载匹配的Visual Studio Code并安装。（本文以 windows 10 64-bit系统Visual Studio Code为例。请下载1.49版本，其他版本不支持 IoT Link）。

图 2-29 下载 Visual Studio Code




注：Visual Studio Code不支持苹果Mac系统。

步骤3 安装成功后，打开VSCode 插件应用商店，搜索 IoT Link 找到IoT Link，然后单击安装。

步骤4 首次启动配置。

IoT Link Studio 首次启动时会自动从网络下载最新的SDK包以及gcc依赖环境，请确保您的网络可用。安装过程中请不要关闭窗口，耐心等待。安装完成后重启VSCode使插件生效。

📖 说明

若您的网络需要配置代理，请在VSCode主页，选择左下角-设置-应用程序-代理服务器，在Proxy Support下拉框中选择“on”。

步骤5 假如下载SDK包及gcc依赖环境失败，请手动下载**SDK包**，放到C:\Users\\${用户名}\.iotlink\sdk目录下,文件名修改为IoT_LINK。放置完后，重新打开VSCode即可。目录格式如下图：

----结束

配置 IoT link Studio 工程

步骤1 单击VSCode底部工具栏的“Home”。

- Home：管理IoT Link工程；
- Serial：输入AT指令检查开发板状态；
- Build：编译示例代码（步骤3后可见）；
- Download：把编译后的代码烧录到MCU（步骤3后可见）。

步骤2 配置交叉编译工具链。在弹出界面中单击“IoT Link设置”，选择工具链，若GCC工具目录或文件不存在，单击**下载安装**。

📖 说明

小熊派STM32431下载的编译工具链版本为win32.zip。

步骤3 在弹出界面中单击“创建 IoT 工程”，输入工程名称，工程目录，并选择开发板的硬件平台和示例工程模板。

- 工程名称：自定义，如QuickStart。
- 工程目录：可以使用工具安装的默认路径，也可以选择系统盘以外的其他盘，如D:\。
- 硬件平台：当前提供的demo只适配STM32L431_BearPI硬件平台，请选择STM32L431_BearPI。
- 示例工程：本示例中，我们以智慧路灯为例，请选择oc_streetlight_template，否则烧录的demo样例和在控制台定义的产品模型不匹配，无法上报数据。如果您需要适配智慧烟感等其他场景，请选择对应的oc_smoke_template demo样例。

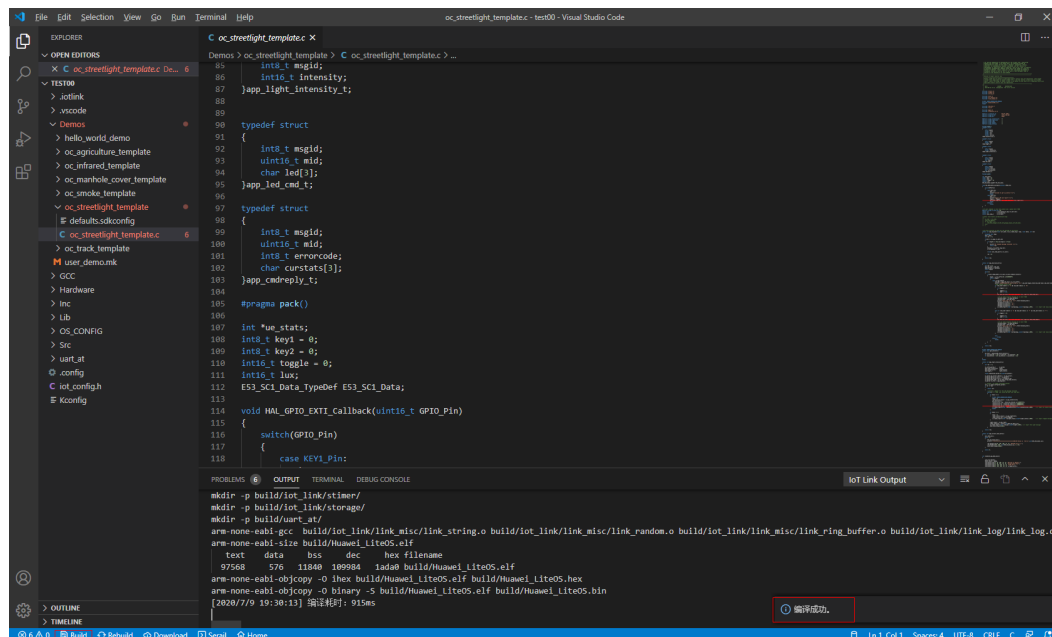
步骤4 单击“确定”，导入完成。

----结束

编译并烧录代码

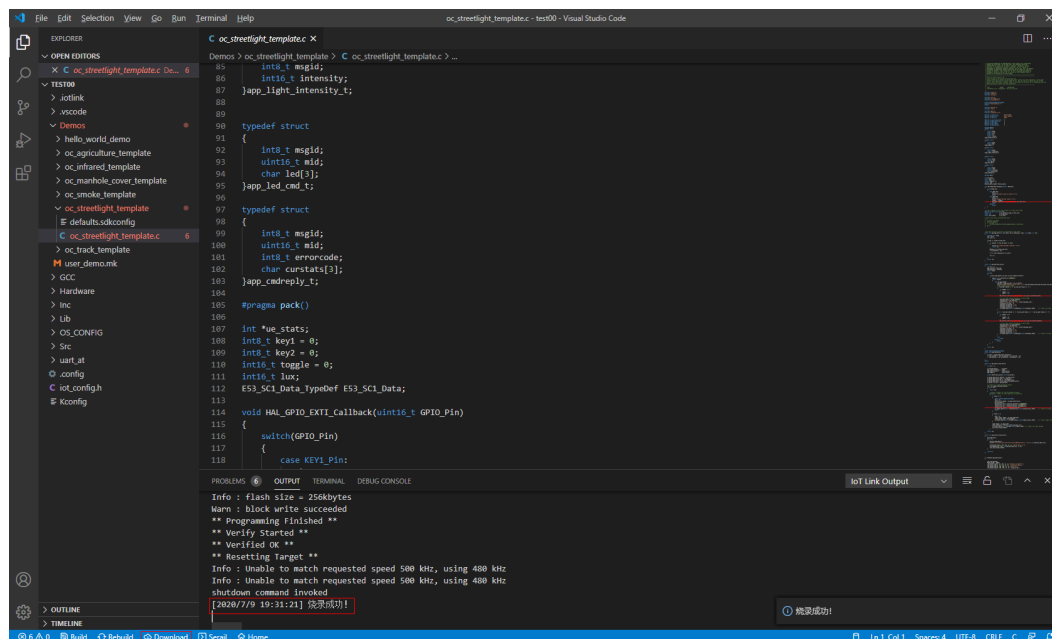
由于提供的demo样例中已配置好和华为云物联网平台的对接信息，您可以直接编译（代码不用做任何修改），并烧录到小熊派开发板MCU，节省开发时间。

步骤1 单击VSCode底部工具栏的“Build”，等待系统编译完成。编译成功后，界面显示“编译成功”。



步骤2 使用USB数据线，将小熊派开发与电脑连接，开发板右上角的拨测开关拨到右侧“AT-MCU模式”。

步骤3 单击VSCode底部工具栏的“Download”，等待系统烧录完成。烧录成功后，界面显示“烧录成功”。



说明

如果显示烧录失败，可能是开发板没有驱动导致与电脑无法串口通信，请参考[2](#)检查ST-Link驱动是否安装成功，如果驱动未安装，则参考[这里](#)下载并安装ST-Link驱动。

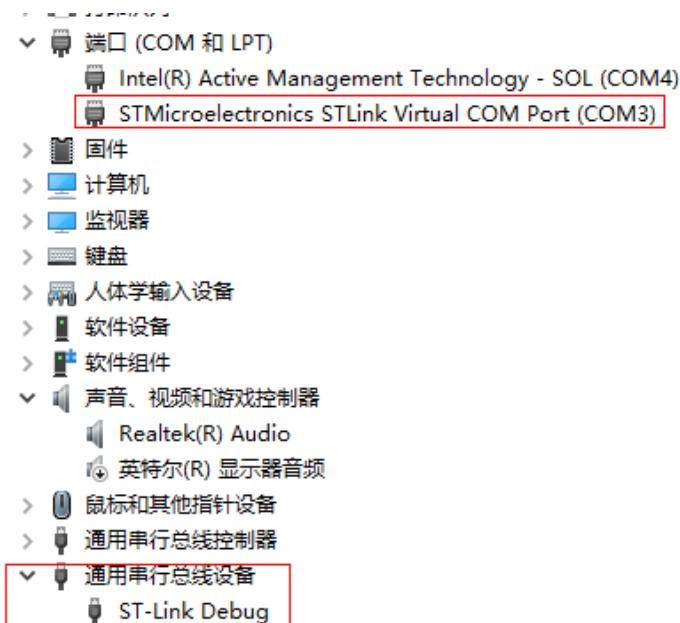
步骤4 （可选）安装ST-Link驱动。

1. 访问[ST官网](#)，下载ST-Link驱动，双击stlink_winusb_install.bat文件进行自动安装。（本文以Windows10-64bit ST-Link 2.0.1为例）。

Name	Size	Packed	Type
..			文件夹
amd64			文件夹
x86			文件夹
dpinst_amd64.exe	680,440	242,269	应用程序
dpinst_x86.exe	552,328	231,230	应用程序
readme.txt	391	250	文本文档
stlink_bridge_winusb.inf	2,853	1,093	安装信息
stlink_dbg_winusb.inf	4,373	1,347	安装信息
stlink_VCP.inf	2,467	871	安装信息
stlink_winusb_install.bat	412	259	Windows 批处理...
stlinkbridgewinusb_x64.cat	11,004	5,890	安全目录
stlinkbridgewinusb_x86.cat	11,004	5,892	安全目录
stlinkdbgwinusb_x64.cat	10,997	5,891	安全目录
stlinkdbgwinusb_x86.cat	10,998	5,892	安全目录
stlinkvcp_x64.cat	9,248	5,474	安全目录
stlinkvcp_x86.cat	9,247	5,470	安全目录

注：您也可以使用适配您系统版本的exe文件进行安装。

2. 打开PC设备管理器可查看对应的驱动是否安装成功。若下图所示，表明驱动安装成功。

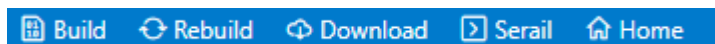


----结束

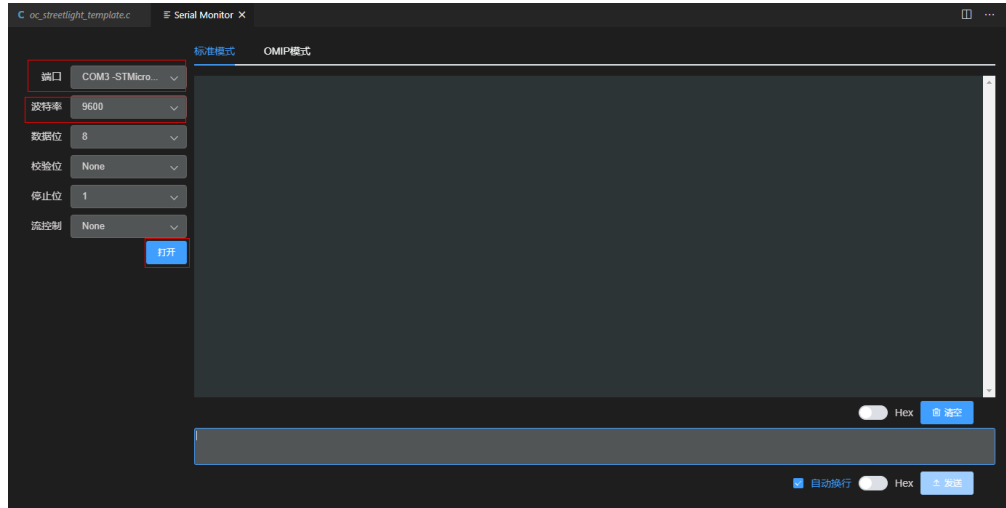
使用 AT 指令定位模组通信问题

IoT Link在与物联网平台连接使用时，可使用AT指令快速定位模组与云端连通性问题，提高开发效率。本节以“小熊派开发板”为例，介绍如何使用AT指令检测通信模组常见问题，如设备无法上线，数据上报不成功等。

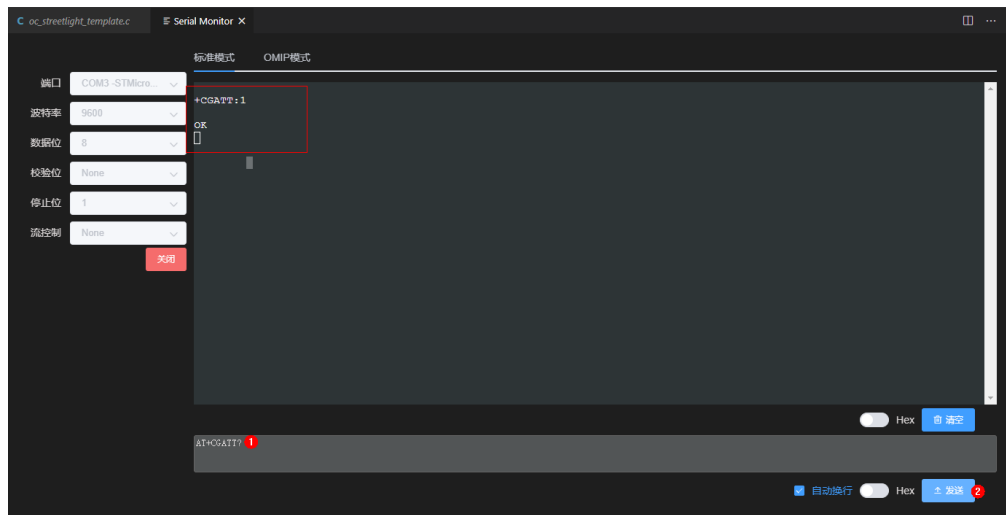
1. 小熊派开发板和电脑已正常连接（确保驱动已安装），并将开发板右上角的拨测开关切换到**AT-PC**模式。
2. 单击VSCode底部工具栏的“Serial”。



3. 选择**2**中查看的端口号，波特率设置为9600，然后单击“打开”。



4. 输入“AT+CGATT?”，然后单击“发送”，若返回“+CGATT:1”，表示网络附着成功（附着成功代表NB-IoT联网正常），返回“+CGATT:0”表示网络附着失败（附着失败代表NB-IoT联网异常），请查看SIM卡是否插入正确，或联系运营商检查网络状态。



📖 说明

使用AT指令检测完模组通信后，请将拨测开关拨到**AT-MCU**模式，以便完成控制台的开发后，将采集到的传感器数据通过通信模组发送到平台。

PC模式是开发板与电脑串口通信，AT指令读写开发板的状态等数据；MCU模式是开发板通过模组上插的SIM连接网络，实现NB-IoT通信。

5. “AT+CSQ<CR>”指令用于检查网络信号强度和SIM卡情况。输入“AT+CSQ<CR>”，然后单击“发送”，返回“+CSQ:**,##”。其中**应在10到31之间，数值越大表明信号质量越好，##为误码率，值在0到99之间。若返回的值不在这个范围，应检查天线或SIM卡是否正确安装。

注：本文仅列举两个通用的AT指令，用于检测模组的网络状况。更多AT指令，您可以参考小熊派模组的说明书，进行调测。

控制台操作概览

在完成实体设备硬件连接和代码编译烧录之后，我们需要在设备接入服务控制台创建产品、定义产品模型、开发编解码插件和注册设备。

- **创建产品**：在物联网平台上规定好某一款产品的协议类型、数据格式、厂商名称、设备类型。此处我们需要按照智慧路灯的特征，在控制台创建智慧路灯的产品。
- **定义产品模型**：产品模型是用来描述设备能力的文件，通过JSON的格式定义了设备的基本属性、上报数据和下发命令的消息格式。定义产品模型，即在物联网平台构建一款设备的抽象模型，使平台理解该款设备支持的属性信息。此处，我们需要在控制台上，定义开关灯控制、光照强度、信号质量等。
- **编解码插件开发**：编解码插件是供物联网平台调用，完成二进制格式和JSON格式相互转换的。它将设备上报的二进制数据解码为JSON格式供应用服务器“阅读”，将应用服务器下行的JSON格式命令编码为二进制格式数据供终端设备（UE）“理解执行”。智慧路灯的数据格式是二进制，因此我们需要开发编解码插件，让物联网能够理解智慧路灯上报的数据，智慧路灯也能理解物联网平台下发的命令。
- **注册设备**：将小熊派智慧路灯注册到物联网平台。

创建产品

某一类具有相同能力或特征的设备的集合称为一款产品。除了设备实体，产品还包含该类设备在物联网能力建设中的产品信息、产品模型（Profile）、编解码插件等资源。下面，我们按照小熊派智慧路灯的基本特征，在控制台上创建智慧路灯产品。

步骤1 登录**管理控制台**，单击左侧导航栏“产品”，单击页面左侧的“创建产品”。

图 2-30 产品-创建产品



步骤2 创建一个协议类型为LwM2M/CoAP协议、设备类型为StreetLamp的产品，参考页面提示填写参数后，单击“确定”。

图 2-31 创建产品-CoAP

---结束

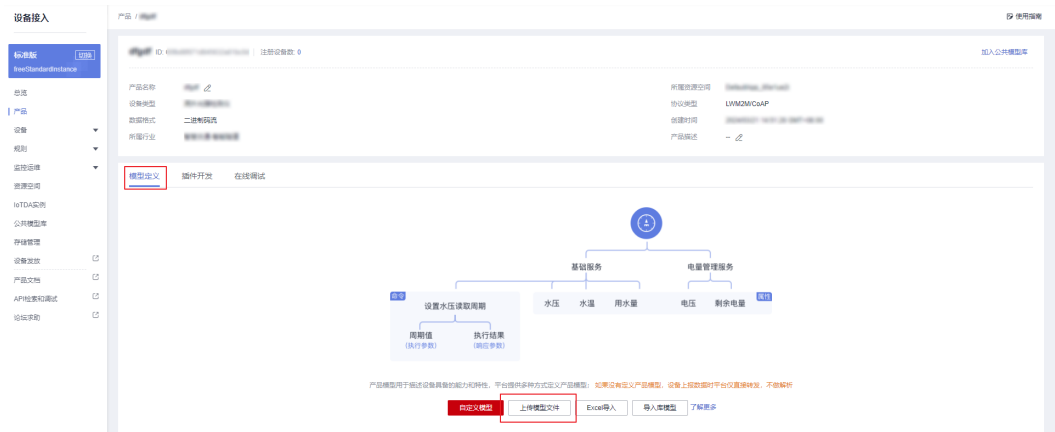
上传产品模型

产品模型是用来描述设备能力的文件，通过JSON的格式定义了设备的基本属性、上报数据和下发命令的消息格式。定义产品模型，即在物联网平台构建一款设备的抽象模型，使平台理解该款设备的功能。我们已提供开发好的产品模型，帮助您快速体验上云流程，如果您想体验产品模型的开发流程，可参考[开发产品模型](#)。

操作步骤：

- 步骤1 找到新增的产品，单击产品进入产品界面。
- 步骤2 在产品详情“模型定义”页面，单击“上传模型文件”。

图 2-32 上传产品模型-CoAP



步骤3 在弹出的页面中，上传提供的产品模型（[本示例的产品模型](#)），然后单击“确认”。

图 2-33 上传模型文件-CoAP

上传模型文件



您可以通过了解产品模型格式规范，在本地进行开发、打包并上传。[点击了解产品模型](#)

StreetLight_58f383040b1e2... (1.93KB) X

添加文件

确定

取消

----结束

注册设备

本文介绍集成NB模组设备的注册方法，将小熊派智慧路灯在物联网平台注册。

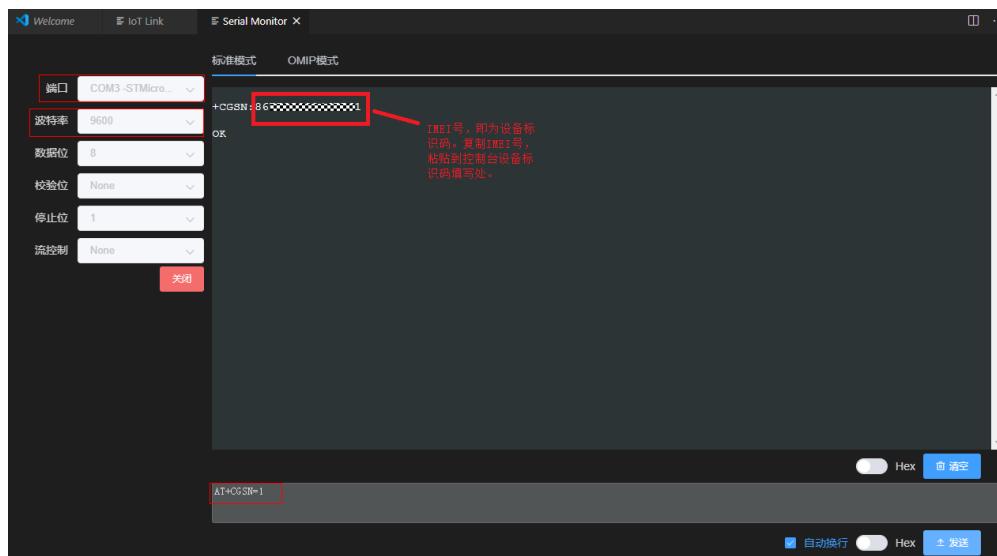
步骤1 在产品详情页面，选择“在线调试”，单击“新增测试设备”，此处新增的是非安全的NB-IoT设备。

步骤2 在新增测试设备页面，选择“真实设备”，完成设备参数的填写后，单击“确定”。

图 2-34 在线调试-新增测试设备

- 设备名称：自定义。
- 设备标识码：设备的IMEI号，用于设备在接入物联网平台时携带该标识信息完成接入鉴权，可在NB模组上查看。您也可以将拨测开关拨到**AT-PC模式**，选择STM的端口，波特率设置为9600，输入指令“AT+CGSN=1”获取IMEI号。

注：获取IMEI注册完设备后，需要将开发板的拨测开关拨到“AT-MCU”模式，因为开发板在MCU模式下才会通过NB卡连接网络。



- 设备注册方式：不加密。

步骤3 设备创建成功，可在页面看到创建的设备。

----结束

数据上报

平台和开发板建立连接后，根据烧录至开发板的代码，小熊派智慧路灯每隔2秒（上报频率可根据业务需要在demo中自行设置）上报光照传感器数据。您可以通过用手遮挡光线改变光照强度，查看设备上报给平台的光照强度数据的实时变化。

注：先检查确认开发板的拨测开关是“AT-MCU”模式。

步骤1 登录**设备接入服务控制台**，选择“设备 > 所有设备”。

步骤2 选择此处注册的设备，单击“查看”进入设备详情，查看上报到平台的数据。

图 2-35 查看上报数据-智慧路灯



----结束

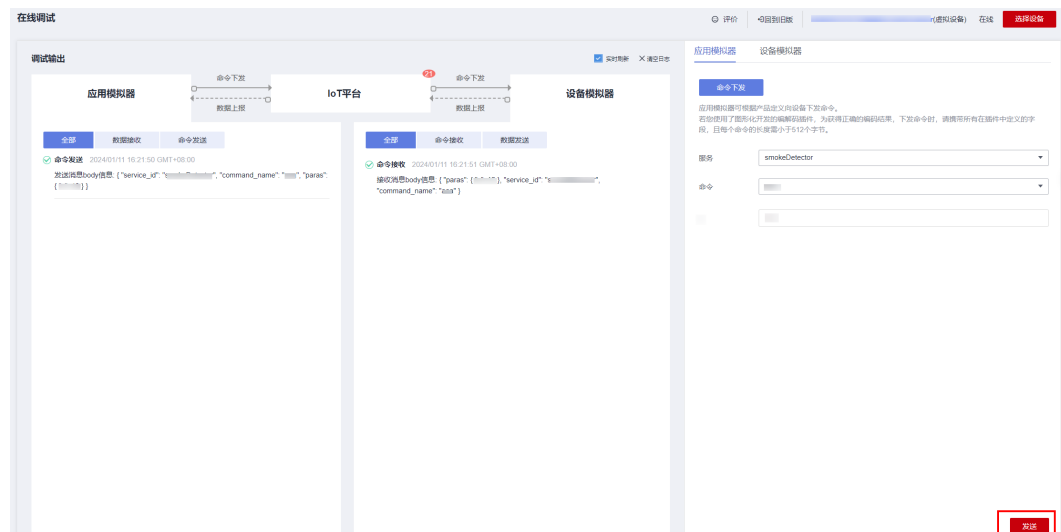
命令下发

步骤1 登录**设备接入服务控制台**，选择之前创建的产品，单击产品进入产品界面。

步骤2 选择“在线调试”，单击注册的设备进入调试页面。

步骤3 设置命令参数后，单击“命令发送”。

图 2-36 命令下发调试-SmokeDetector



步骤4 小熊派板子灯被点亮，下发OFF命令后，灯熄灭。



----结束

至此，基于NB-IoT小熊派开发智慧路灯的最佳实践，端到端流程就全部体验完成了。

更多参考

- 开发产品模型**
 在产品详情“模型定义”页面，单击“自定义模型”，配置产品的服务。
 产品模型设计思路：

表 2-3 设备服务列表

服务类型 (ServiceID)	服务描述
Button	实时按键检测
LED	LED灯控制
Sensor	实时检测光照强度
Connectivity	实时检测信号质量

服务能力如下表所示。

表 2-4 Button

能力描述	属性名称	数据类型	数据范围
属性列表	toggle	int	0 ~ 65535

表 2-5 LED 命令列表

能力描述	命令名称	命令字段	字段名称	类型	数据长度	枚举
命令列表	Set_LED	下发命令	LED	string	3	ON、OFF
		响应命令	Light_state	string	3	ON、OFF

表 2-6 Sensor

能力描述	属性名称	数据类型	数据范围
属性列表	luminance	int	0 ~ 65535

表 2-7 Connectivity

能力描述	属性名称	数据类型	数据范围
属性列表	SignalPower	int	-140 ~ -44
	ECL	int	0 ~ 2
	SNR	int	-20 ~ 30
	CellID	int	0 ~ 65535

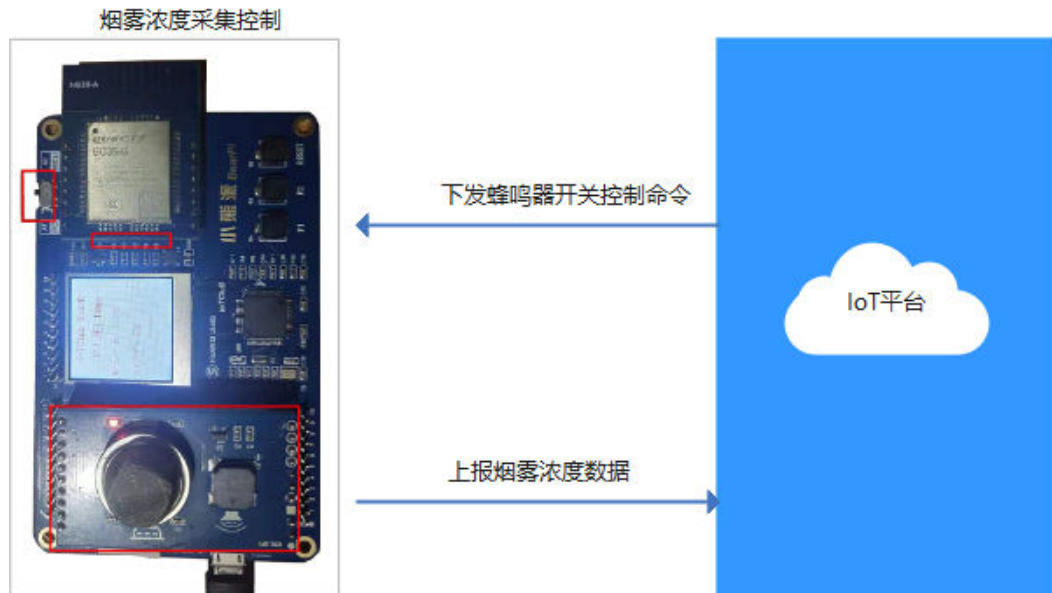
2.4 基于 NB-IoT 小熊派的智慧烟感

场景说明

近年来，火灾频发，造成很大的人员及财产损失，国家高度重视，陆续出台相关文件，明确相关要求，落实消防安全责任。同时，人民消防意识逐渐提高，独立烟感得到一定程度的普及，在防火减灾方面起到一定的作用。但由于独立烟感产品存在一定的局限，例如人不在场收不到报警信息，工作状态不能实时掌握，独立烟感没有完全解决这些问题。

NB-IoT智慧烟感解决了传统烟感器布线难、电池使用周期短、维护成本高、无法与业主及消防机构交互等缺点。智慧烟感采用无线通信，具备即插即用、无需布线、易于安装等特点。

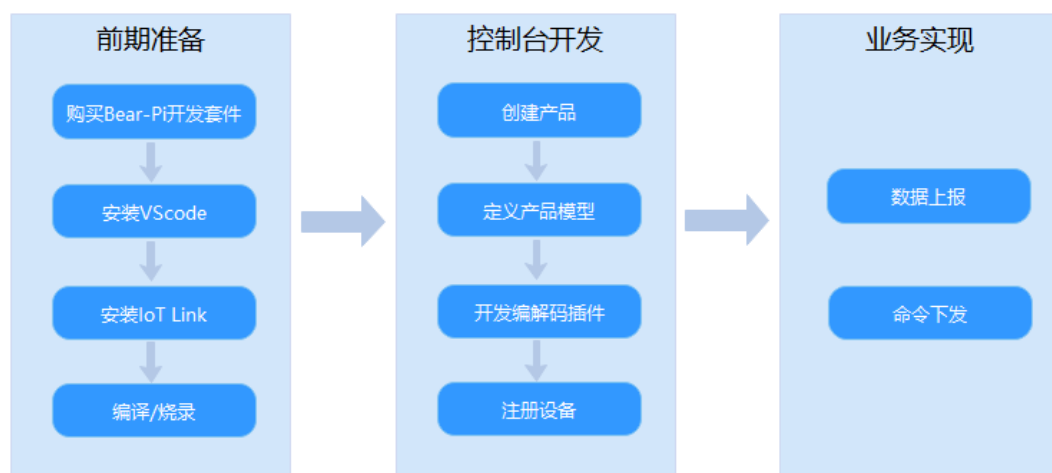
在该文档中，我们基于华为一站式开发工具平台（VS code工具IoTlink插件），从设备（小熊派开发套件）、平台（IoTDA华为物联网平台）、端到端构建一款智慧烟感解决方案样例，带您体验十分钟快速上云。实现智慧烟感检测并上报烟雾浓度值，显示在IoTDA控制台，并在IoTDA控制台远程控制蜂鸣器开关的功能。



开发环境

- 硬件：小熊派BearPi-IoT开发套件（包含NB卡、NB模组、智慧路灯功能模块、USB数据线等）。
- 软件：VScode，IoT Link插件、华为云（已开通**设备接入服务**）、win7及以上版本的64位操作系统（本文以windows 10 64-bit系统调试）。

整体流程



小熊派开发板简介

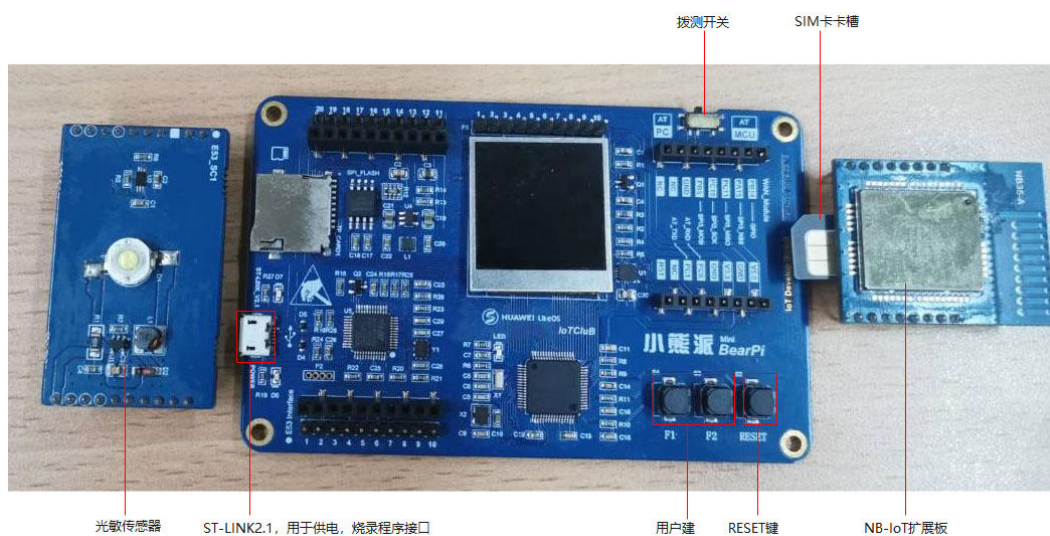
开发板在物联网系统架构中属于感知设备，该类设备通常由传感器、通信模组、芯片以及操作系统组成。为增加开发板的可扩展性，小熊派开发板没有采用传统的板载设

计，而是使用了可更换传感器扩展板以及可更换通信模组扩展板设计，通信模组是数据传输的出入口，常用的通信模组包括NB-IoT，Wifi以及4G等，芯片是设备的主控设备，该开发板内置了一个低功耗的STM32L431单片机作为主控芯片，即MCU。操作系统使用的是华为的LiteOS操作系统，其提供了丰富的端云互通组件。

为了便于开发调试，如图1所示，该开发板板载了2.1版本的ST-Link，它具有在线调试烧录，拖拽下载以及虚拟串口等功能。开发板中间板载一块分辨率为240 * 240的LCD屏幕，其主要用于显示传感器数据以及调试日志。LCD屏幕下方是主控芯片。

开发板右上角具有一个拨码开关，将其拨至左侧AT-PC模式，通过电脑端的串口助手，发送AT指令调试通信模组。右侧AT-MCU模式，通过MCU发送AT指令与通信模组进行交互，将采集到的传感器数据通过通信模组发送到云端。

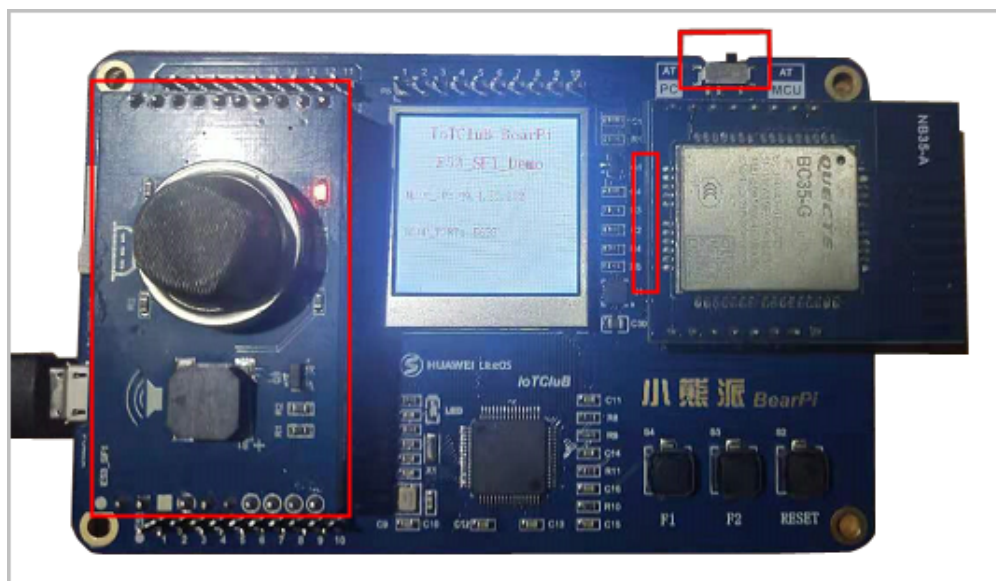
图 2-37 小熊派开发板



硬件连接

1. 将NB卡插入到NB-IoT扩展板的SIM卡卡槽，确保插卡的时候卡的缺口朝外插入，如图2所示。
2. 将烟雾浓度采集控制板和NB-IoT扩展板插入到开发板上，注意安装方向，然后用USB数据线将小熊派开发板与电脑连接起来。显示屏有显示，电源灯被点亮，说明开发板通电成功。

图 2-38 硬件连接



安装 IoT Link Studio 插件

IoT Link Studio是针对IoT物联网端侧开发的IDE环境，提供了编译、烧录、调试等一站式开发体验，支持 C、C++、汇编等多种开发语言，让您快速、高效地进行物联网开发。

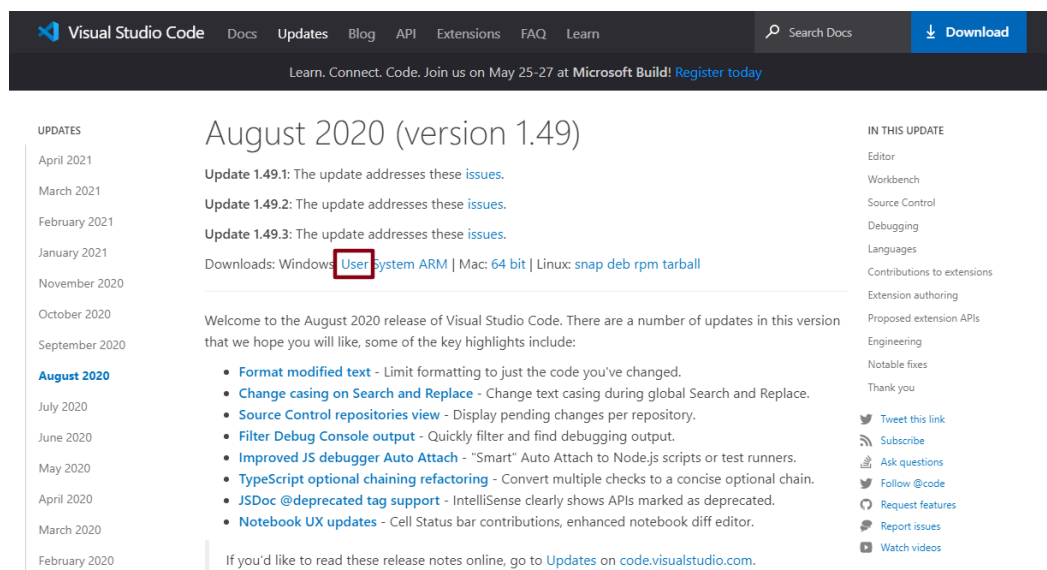
步骤1 以win10为例，获取电脑的系统配置：在运行窗口中输入pc，然后单击“属性”，查看系统配置，因为后续安装软件要选择操作系统。

图 2-39 获取系统配置



步骤2 访问 [这里](#)，根据您电脑系统配置，下载匹配的Visual Studio Code并安装。（本文以 windows 10 64-bit系统Visual Studio Code为例。请下载1.49版本，其他版本不支持 IoT Link）。

图 2-40 下载 Visual Studio Code




注：Visual Studio Code不支持苹果Mac系统。

步骤3 安装成功后，打开VSCode 插件应用商店，搜索 IoT Link 找到IoT Link，然后单击安装。

步骤4 首次启动配置。

IoT Link Studio 首次启动时会自动从网络下载最新的SDK包以及gcc依赖环境，请确保您的网络可用。安装过程中请不要关闭窗口，耐心等待。安装完成后重启VSCode使插件生效。

📖 说明

若您的网络需要配置代理，请在VSCode主页，选择左下角-设置-应用程序-代理服务器，在Proxy Support下拉框中选择“on”。

步骤5 假如下载SDK包及gcc依赖环境失败，请手动下载**SDK包**，放到C:\Users\\${用户名}\.iotlink\sdk目录下,文件名修改为IoT_LINK。放置完后，重新打开VSCode即可。目录格式如下图：

----结束

配置 IoT link Studio 工程

步骤1 单击VSCode底部工具栏的“Home”。

- Home：管理IoT Link工程；
- Serial：输入AT指令检查开发板状态；
- Build：编译示例代码（步骤3后可见）；
- Download：把编译后的代码烧录到MCU（步骤3后可见）。

步骤2 配置交叉编译工具链。在弹出界面中单击“IoT Link设置”，选择工具链，若GCC工具目录或文件不存在，单击**下载安装**。

📖 说明

小熊派STM32431下载的编译工具链版本为win32.zip。

步骤3 在弹出界面中单击“创建 IoT 工程”，输入工程名称，工程目录，并选择开发板的硬件平台和示例工程模板。

- 工程名称：自定义，如Smoke。
- 工程目录：可以使用工具安装的默认路径，也可以选择系统盘以外的其他盘，如D:\。
- 硬件平台：当前提供的demo只适配STM32L431_BearPI硬件平台，请选择STM32L431_BearPI。
- 示例工程：本示例中，我们以智慧烟感为例，请选择oc_smoke_template demo，否则烧录的demo样例和在控制台定义的产品模型不匹配，无法上报数据。单击“确定”，导入完成。

步骤4 单击“确定”，导入完成。

----结束

编译并烧录代码

由于提供的demo样例中已配置好和华为云物联网平台的对接信息，您可以直接编译（代码不用做任何修改），并烧录到小熊派开发板MCU，节省开发时间。

步骤1 单击VSCode底部工具栏的“Build”，等待系统编译完成。编译成功后，界面显示“编译成功”。

步骤2 使用USB数据线，将小熊派开发与电脑连接，开发板右上角的拨测开关拨到右侧“AT-MCU模式”。

步骤3 单击VSCode底部工具栏的“Download”，等待系统烧录完成。烧录成功后，界面显示“烧录成功”。

📖 说明

如果显示烧录失败，可能是开发板没有驱动导致与电脑无法串口通信，请参考[2](#)检查ST-Link驱动是否安装成功，如果驱动未安装，则参考[这里](#)下载并安装ST-Link驱动。

步骤4 （可选）安装ST-Link驱动。

1. 访问[ST官网](#)，下载ST-Link驱动，双击stlink_winusb_install.bat文件进行自动安装。（本文以Windows10-64bit ST-Link 2.0.1为例）。

注：您也可以使用适配您系统版本的exe文件进行安装。

2. 打开PC设备管理器可查看对应的驱动是否安装成功。若下图所示，表明驱动安装成功。

----结束

创建产品

某一类具有相同能力或特征的设备的集合称为一款产品。除了设备实体，产品还包含该类设备在物联网能力建设中的产品信息、产品模型（Profile）、插件等资源。

步骤1 登录**管理控制台**，单击左侧导航栏“产品”，单击页面左侧的“创建产品”。

图 2-41 产品-创建产品



步骤2 创建一个协议类型为LwM2M/CoAP协议、设备类型为SmokeSensors的产品，参考页面提示填写参数后，单击“确定”。

图 2-42 创建产品-CoAP



----结束

定义产品模型

在产品详情，“模型定义”页面，单击“自定义模型”，配置产品的服务。具体创建方法可见：

产品模型设计思路：

表 2-8 设备服务列表

服务类型 (ServiceID)	服务描述
Smoke	实时检测烟雾状况

服务能力如表2、表3所示。

表 2-9 Smoke

能力描述	属性名称	数据类型	数据范围
属性列表	Smoke value	int	0 ~ 65535

表 2-10 Smoke 命令列表

能力描述	命令名称	命令字段	字段名称	类型	数据长度	枚举
命令列表	Smoke control beep	下发命令	beep	string	3	ON、OFF
		响应命令	beep_state	int	/	/

新增服务类型Smoke

1. 进入“添加服务”页面，填写“服务ID”、“服务类型”和“服务描述”，单击“确认”。
 - “服务ID”：Smoke
 - “服务类型”：建议和服务ID保持一致
 - “服务描述”：实时检测烟雾状况

图 2-43 添加服务-Smoke

添加服务

* 服务ID

服务类型 ?

服务描述

8/128

确定 取消

2. 在“Smoke”的下拉菜单下单击“添加属性”，填写相关信息，单击“确认”。
 - 属性名称：Smoke_Value
 - “数据类型”：int（整型）
 - “访问权限”：可读、可写
 - “取值范围”：0~65535
 - “步长”：0
 - “单位”：不填写

图 2-44 新增属性-Smoke_value

×

新增属性

* 属性名称

属性描述
0/128

* 数据类型

* 访问权限

* 取值范围 -

步长

单位

3. 在“Smoke”的下拉菜单下单击“添加命令”，填写命令名称“Smoke_Control_Beep”。

图 2-45 新增命令-Smoke_Control_Beep

新增命令

* 命令名称:

下发参数:

参数名称	数据类型	描述	操作
暂无表格数据 当前无下发参数数据, 请先新增输入参数			

5 | 总条数: 0 | < 1 >

响应参数:

参数名称	数据类型	描述	操作
暂无表格数据 当前无响应参数数据, 请先新增响应参数			

- 在“新增命令”弹框，分别单击“新增输入参数”和“新增响应参数”，填写相关信息后，单击“确认”。

图 2-46 新增命令参数-Beep

×

新增参数

* 参数名称

参数描述
0/128

* 数据类型

* 长度

枚举值
6/1,024

图 2-47 新增命令参数-Beep_State

×

新增参数

* 参数名称

参数描述
0/128

* 数据类型

* 取值范围 -

步长

单位

注册设备

本文介绍集成NB模组设备的注册方法。

- 步骤1** 在产品详情页面，选择“在线调试”，单击“新增测试设备”，此处新增的是非安全的NB-IoT设备。
- 步骤2** 在新增测试设备页面，选择“真实设备”，完成设备参数的填写后，单击“确定”。

图 2-48 在线调试-新增测试设备



新增测试设备

设备类型: 真实设备 虚拟设备

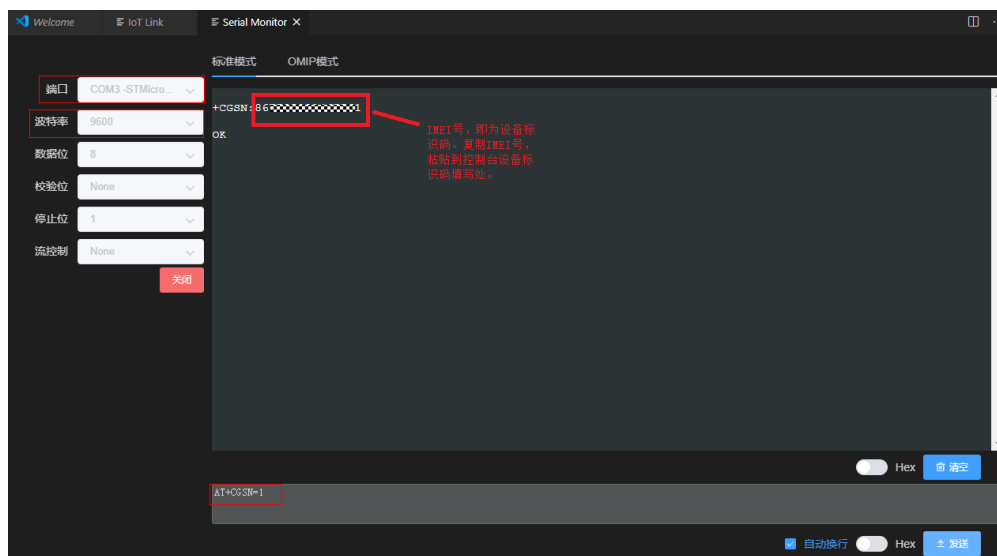
设备名称:

* 设备标识码:

设备注册方式: 不加密 加密

- 设备名称：自定义。
- 设备标识码：设备的IMEI号，用于设备在接入物联网平台时携带该标识信息完成接入鉴权，可在NB模组上查看。您也可以将拨测开关拨到**AT-PC模式**，选择STM的端口，波特率设置为9600，输入指令“AT+CGSN=1”获取IMEI号。

注：获取IMEI注册完设备后，需要将开发板的拨测开关拨到“AT-MCU”模式，因为开发板在MCU模式下才会通过NB卡连接网络。



- 设备注册方式：不加密。

步骤3 设备创建成功，可在页面看到创建的设备。

----结束

数据上报

平台和开发板建立连接后，上报烟雾浓度值。

步骤1 登录[设备接入服务控制台](#)，选择“设备 > 所有设备”。

步骤2 选择注册的设备，单击“查看”进入设备详情，查看上报到平台的数据。

图 2-49 查看上报数据-Smoke



----结束

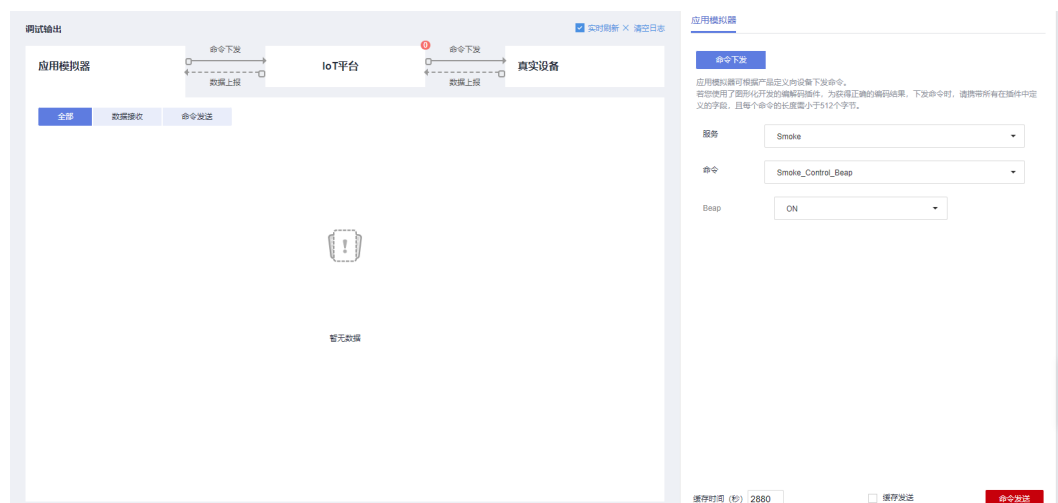
命令下发

步骤1 登录[设备接入服务控制台](#)，选择创建的产品，单击产品进入产品详情页。

步骤2 选择“在线调试”，单击注册的设备进入调试页面。

步骤3 设置命令参数后，单击“命令发送”。

图 2-50 命令下发调试-Smoke



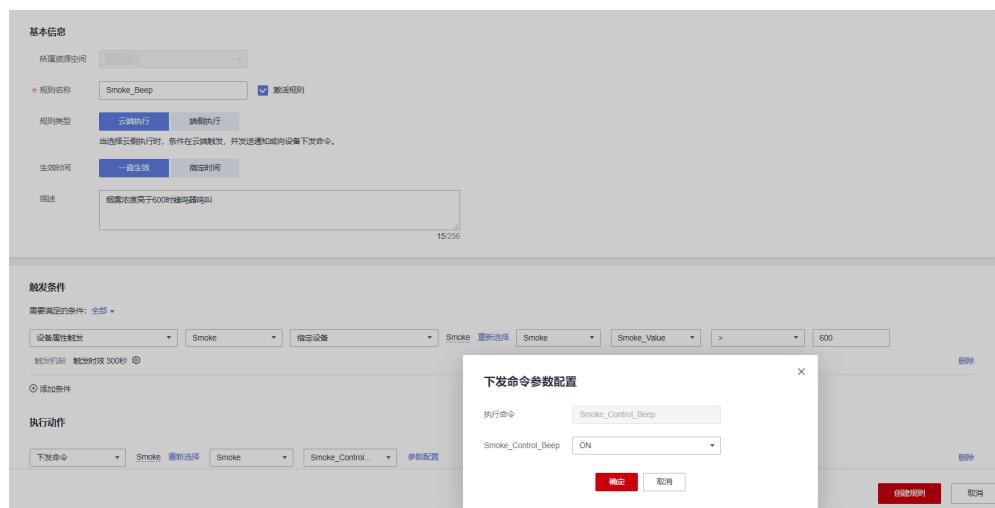
----结束

(可选) 配置设备联动规则

步骤1 选择左侧导航栏的“规则 > 设备联动”，单击页面右上角的“创建规则”。

步骤2 参考下表参数说明，创建设备联动规则。

图 2-51 新建联动规则-Smoke_Beep



参数名称	说明
规则名称	创建的规则名称，例如“Smoke_Beep”。
激活规则	勾选“激活规则”
生效时间	选择“一直生效”。
描述	对该规则的描述，如“烟雾浓度高于600时蜂鸣器鸣叫”。
触发条件	<ol style="list-style-type: none"> 单击“添加条件”； 在下拉框中选择“设备属性触发”，选择注册设备中添加的设备； “服务类型”选择“smoke”，“属性”选择“Smoke_Value”，“操作”选择“>”，“值”填写“600”，单击“触发机制”，“触发策略”选择“重复抑制”，数据时效”填写“300”，然后单击“确认”。
执行动作	<ol style="list-style-type: none"> 单击“添加动作”； “动作类型”选择“下发命令”，下发设备选择注册设备创建的设备； “服务类型”选择“smoke”，“命令”选择“Smoke_Control_Beep”，单击“参数配置”，“Beep”的值选择“ON”，然后单击“确认”。

----结束

(可选) 验证操作

将降温剂喷在烟雾传感器周围，模拟烟雾浓度。当烟雾浓度大于600时，蜂鸣器鸣叫。

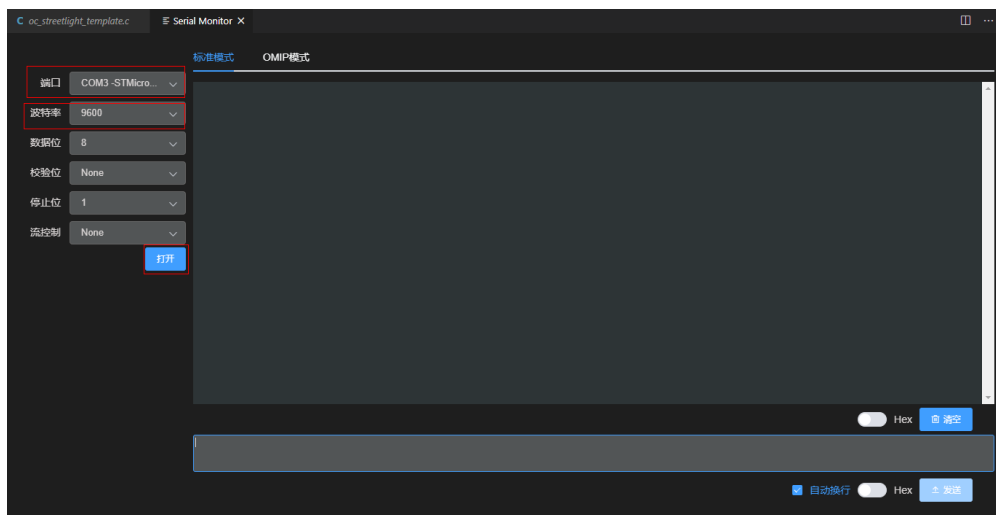
使用 AT 指令定位模组通信问题

IoT Link在与物联网平台连接使用时，可使用AT指令快速定位模组与云端连通性问题，提高开发效率。本节将以“小熊派开发板”为例，介绍如何使用AT指令检测通信模组常见问题，如设备无法上线，数据上报不成功等。

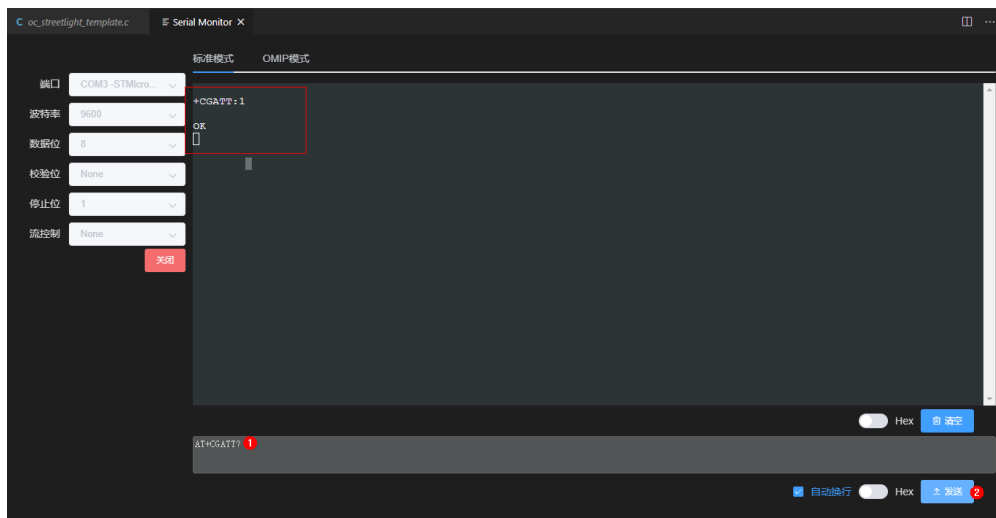
1. 小熊派开发板和电脑已正常连接（确保驱动已安装），并将开发板右上角的拨测开关切换到AT-PC模式。
2. 单击VSCode底部工具栏的“Serial”。



3. 选择步骤4.2中查看的端口号，波特率设置为9600，然后单击“打开”。



4. 输入“AT+CGATT?”，然后单击“发送”，若返回“+CGATT:1”，表示网络附着成功（附着成功代表NB-IoT联网正常），返回“+CGATT:0”表示网络附着失败（附着失败代表NB-IoT联网异常），请查看SIM卡是否插入正确，或联系运营商检查网络状态。



📖 说明

使用AT指令检测完模组通信后，请将拨测开关拨到AT-MCU模式，以便完成控制台的开发后，将采集到的传感器数据通过通信模组发送到平台。

PC模式是开发板与电脑串口通信，AT指令读写开发板的状态等数据；MCU模式是开发板通过模组上插的SIM连接网络，实现NB-IoT通信。

2.5 模拟 NB 设备智慧路灯的接入与调试

场景说明

本文以“智慧路灯”为示例，通过设备接入控制台提供的设备模拟器和应用模拟器替代真实的设备和应用，带您快速体验设备上报数据到物联网平台和远程下发控制命令到设备的全过程。

假设：

路灯设备上报一条数据消息，包含路灯的光照强度（light_intensity）和路灯的开关状态（light_status）；支持远程控制路灯开关状态的命令（SWITCH_LIGHT），上报数据的格式为二进制格式。

前提条件

- 已注册华为云官方账号。未注册可单击[注册页面](#)完成注册。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，单击“立即使用”后开通该服务。

（可选）创建资源空间

资源空间是物联网平台提供物联网应用和设备的调测空间，您可以根据场景的不同创建不同资源空间分别调测。

系统为您预置了一个资源空间，您可以直接在预置的资源空间下，在线开发设备的产品模型文件和编解码插件；也可以新建一个资源空间，具体步骤如下。

步骤1 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。

步骤2 选择左侧导航栏的“资源空间”，单击右上角“新建资源空间”。

步骤3 在弹出的页面填写“空间名称”，然后单击“确定”。

----结束

创建产品

基于开发的产品，用户可以在线开发设备的产品模型文件和编解码插件；另外，物联网平台提供了设备和应用模拟器，可以便捷地调测开发的产品模型文件和编解码插件的正确性。

步骤1 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。

步骤2 选择左侧导航栏的“产品”。

图 2-52 产品-创建产品



步骤3 单击左侧的“创建产品”，创建一个基于LwM2M/CoAP协议的产品，填写参数后，单击“确定”。

图 2-53 创建产品-CoAP

创建产品 ✕

*** 所属资源空间** ?

如需创建新的资源空间，您可前往当前实例详情创建

*** 产品名称**

协议类型 ?

*** 数据格式** ?

设备类型选择

标准类型
自定义类型

*** 设备类型** ?

高级配置 ▲ 定制ProductID | 备注信息

产品ID ?

产品描述

0/128

确定
取消

基本信息	
所属资源空间	选择新建产品所属的资源空间。
产品名称	自定义，如Test_1。

协议类型	选择“LwM2M/CoAP”。
数据格式	选择“二进制码流”。 说明 当“数据格式”配置为“二进制码流”时，该产品下需要进行编解码插件开发；当“数据格式”配置为“JSON”时，该产品下不需要进行编解码插件开发。
所属行业	无
设备类型	streetlamp

----结束

定义产品模型

步骤1 找到步骤3创建的产品，单击产品进入产品界面。

步骤2 在产品详情模型定义页面，单击“自定义模型”，配置产品的服务，然后单击“确定”。

- “服务ID”：StreetLight
- “服务类型”：建议和服务ID保持一致
- “服务描述”：路灯上报的环境光强度和路灯开关状态的属性。

步骤3 单击步骤2新增的服务ID，在展开的页面单击“添加属性”，定义一条属性为路灯采集的当前环境的光照强度。

- “属性名称”：light_intensity
- “数据类型”：int（整型）
- “访问权限”：可读、可写
- “取值范围”：0~100，光照强度范围。

图 2-54 新增属性-light_intensity

新增属性

* 属性名称

属性描述
0/128

* 数据类型

* 访问权限

* 取值范围 -

步长

单位

步骤4 单击“添加属性”，定义一条属性为路灯当前的开关灯状态。

- “属性名称”：light_status
- “数据类型”：int（整型）
- “访问权限”：可读、可写
- “取值范围”：0~1，0代表关闭，1代表打开状态

图 2-55 添加属性-light_status

新增属性

* 属性名称

属性描述
0/128

* 数据类型

* 访问权限 可读 可写

* 取值范围 -

步长

单位

步骤5 接下来定义远程控制开关灯状态的命令。

1. 单击“添加命令”，定义命令名称：SWITCH_LIGHT。
2. 单击“新增输入参数”，参数名称：SWITCH_LIGHT，数据类型为：string，长度为：3个字符，枚举值为：ON,OFF。

图 2-56 新增命令参数-SWITCH_LIGHT

新增参数

* 参数名称 SWITCH_LIGHT

参数描述 0/128

* 数据类型 string(字符串)

* 长度 3

枚举值 ON.OFF 6/1,024

确定 取消

- 单击“新增响应参数”，参数名称：result，数据类型：int。

图 2-57 新增响应参数-result

步骤6 单击“确定”，完成该路灯的产品模型文件开发。

----结束

在线开发插件

通常情况下设备为了省电，设备会采用“二进制”格式上报数据，编解码插件的作用就是将设备上报的“二进制”格式数据，按照产品模型文件的定义的属性转换为“JSON”格式数据，便于物联网平台和应用服务器识别。同时，用户远程下发控制命令时，物联网平台会将“JSON”格式的命令转换为“二进制”格式数据下发给设备。

📖 说明

如果设备本身上报的是JSON格式数据，则不需要定义编解码插件。

步骤1 在智慧路灯产品的开发空间，选择“插件开发”，单击“图形化开发”。

步骤2 单击“新增消息”，配置数据上报消息，上报路灯的数据消息。

- 消息名：LightData
- 消息类型：数据上报
- 添加响应字段：是。添加响应字段后，物联网平台在收到设备上报的数据后，会下发用户设置的响应数据到设备。
- 响应数据：AAAA0000（默认）

图 2-58 插件开发-新增消息

新增消息 ×

基本信息

*消息名


*消息类型 数据上报 命令下发

添加响应字段

描述

0/1,024

偏移值	名字	描述	数据类型	长度	是否地址域	操作
-----	----	----	------	----	-------	----


暂无表格数据

响应数据

1. 单击“添加字段”，添加messageId字段，表示消息种类。

图 2-59 插件开发-添加字段 messaged

×

添加字段

i 只有标记为地址域时，名字固定为messaged；其他字段名字不能设置为messaged。

标记为地址域 ?

* 字段名称	<input type="text" value="messaged"/>
描述	<input type="text" value="输入字段描述"/> 0/1,024
数据类型 (大端模式)	<input type="text" value="int8u"/>
偏移值	<input type="text" value="0-1"/> ?
* 长度	<input type="text" value="1"/> ?
默认值	<input type="text" value="0x0"/> ?

确定 取消

2. 添加**LightIntensity**字段，表示光照强度，数据类型：int8u（8位无符号整型），长度：1字节。

图 2-60 插件开发-添加字段 LightIntensity

✕

添加字段

标记为地址域 ?

* 字段名称

描述
0/1,024

数据类型 (大端模式)

偏移值 ?

* 长度 ?

默认值 ?

确定
取消

3. 添加**LightStatus**字段，表示路灯上报的路灯开关状态数据。数据类型：int8u（8位无符号整型），长度：1字节。

图 2-61 插件开发-添加字段 LightStatus

×

添加字段

标记为地址域 ?

*** 字段名称**

描述 0/1,024

数据类型 (大端模式)

偏移值 ?

*** 长度** ?

默认值 ?

确定 取消

步骤3 再次单击“新增消息”，定义远程控制路灯开关的命令对应的编解码插件消息。

- 消息名：SwitchStatus
- 消息类型：命令下发
- 添加响应字段：是。添加响应字段后，设备在接收命令后，会上报命令执行结果。

图 2-62 插件开发-新增消息 SwitchStatus

新增消息 ×

基本信息

***消息名**

描述

消息描述

0/1,024

***消息类型**

数据上报
 命令下发

添加响应字段

字段 添加字段

偏移值	名字	描述	数据类型	长度	是否地址域	操作
<div style="text-align: center;">  <p>暂无表格数据</p> </div>						

响应字段 添加响应字段

偏移值	名字	描述	数据类型	长度	是否地址域	操作
<div style="text-align: center;">  </div>						

确定
取消

1. 单击“添加字段”，添加messageId字段，表示消息种类。

图 2-63 插件开发-添加字段 messaged

×

添加字段

i 只有标记为地址域时，名字固定为messaged；其他字段名字不能设置为messaged。

标记为地址域 ?

* 字段名称	messaged
描述	输入字段描述 0/1,024
数据类型 (大端模式)	int8u ▼
偏移值	0-1 ?
* 长度	1 ?
默认值	0x0 ?

确定
取消

2. 添加**mid**字段，用于将下发的命令和命令执行结果进行关联。

图 2-64 插件开发-添加命令字段 mid

×

添加字段

i 只有标记为响应标识字段时，名字固定为mid；其他字段名字不能设置为mid。

标记为地址域 ?

标记为响应标识字段 ?

* 字段名称	<input type="text" value="mid"/>
描述	<input type="text" value="输入字段描述"/> 0/1,024
数据类型 (大端模式)	<input type="text" value="int16u"/>
偏移值	<input type="text" value="1-3"/> ?
* 长度	<input type="text" value="2"/> ?
默认值	<input type="text"/> ?

确定 取消

3. 添加SwitchStatus字段，数据类型：string（字符串类型），长度：3字符。

图 2-65 插件开发-添加命令字段 SwitchStatus

×

添加字段

标记为地址域 ?

标记为响应标识字段 ?

* 字段名称

描述 0/1,024

数据类型 (大端模式)

偏移值 ?

* 长度 ?

默认值

4. 单击“添加响应字段”，配置命令下发响应消息。
 - 添加**messageId**，表示消息种类。命令执行结果为上行消息，需要通过**messageId**和数据上报消息进行区分。

图 2-66 插件开发-添加响应字段 messaged

添加字段 ×

i 只有标记为地址域时，名字固定为messaged；其他字段名字不能设置为messaged。

标记为地址域 ?

标记为响应标识字段 ?

标记为命令执行状态字段 ?

* 字段名称

描述 0/1,024

数据类型 (大端模式)

偏移值 ?

* 长度 ?

默认值 ?

确定 取消

- 添加mid字段，用于将下发的命令和命令执行结果进行关联。

图 2-67 插件开发-添加响应字段 mid

×

添加字段

i 只有标记为响应标识字段时，名字固定为mid；其他字段名字不能设置为mid。

标记为地址域 ?

标记为响应标识字段 ?

标记为命令执行状态字段 ?

***** 字段名称

描述 0/1,024

数据类型 (大端模式)

偏移值 ?

***** 长度 ?

默认值 ?

确定 取消

- 添加errcode字段，用于表示命令执行状态：00表示成功，01表示失败，如果未携带该字段，则默认命令执行成功。

图 2-68 插件开发-添加响应字段 errcode

×

添加字段

i 只有标记为命令执行状态字段时，名字固定为errcode；其他字段名字不能设置为errcode。

标记为地址域 ?

标记为响应标识字段 ?

标记为命令执行状态字段 ?

* 字段名称

描述
0/1,024

数据类型 (大端模式)

偏移值 ?

* 长度 ?

默认值 ?

确定取消

- 添加result字段，用于表示命令执行结果。

图 2-69 插件开发-添加响应字段 result

×

添加字段

标记为地址域 ?

标记为响应标识字段 ?

标记为命令执行状态字段 ?

***** 字段名称

描述 0/1,024

数据类型 (大端模式)

偏移值 ?

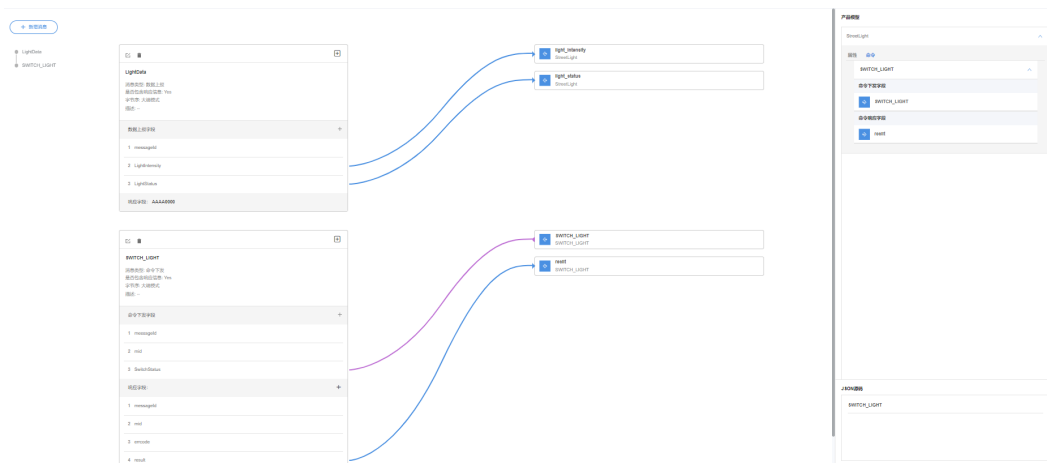
***** 长度 ?

默认值 ?

确定 取消

步骤4 拖动右侧“设备模型”区域的属性字段和命令字段（产品模型文件定义的字段），与编解码插件定义的数据上报消息和命令下发消息的相应字段建立映射关系。

图 2-70 插件开发-在线开发插件



步骤5 编解码插件与产品模型文件建立映射完成后，单击右上角“保存”，并单击“部署”，完成编解码插件的部署。

---结束

使用真实设备在线调试

模拟器在线调测具备设备模拟和应用模拟功能，可以对定义的产品模型文件和编解码插件进行调试，用户可以直观的感受设备上报数据到物联网平台，以及使用物联网平台下发远程控制命令。

步骤1 在智慧路灯产品的开发空间，选择“在线调试”，单击“新增测试设备”。

步骤2 在新增测试设备页面，完成设备参数的填写后，单击“确定”。

- 设备类型：真实设备。
- 设备名称：自定义。
- 设备标识码：设备的IMEI号，用于设备在接入物联网平台时携带该标识信息完成接入鉴权，可在NB模组上查看。
- 设备注册方式：不加密。如果使用DTLS传输层安全协议接入时，设备注册方式选择“加密”，且请妥善保存密钥。

图 2-71 在线调试-新增测试设备



说明

新添加的设备处于未激活状态，此时不能进行在线调试，可参考[连接鉴权](#)，待设备接入平台后，进行调试。

步骤3 单击“调试”，进入调试界面。

图 2-72 在线调试-进入调试



步骤4 模拟远程下发控制命令场景，在应用模拟器中，选择服务：StreetLight，命令：SWITCH_LIGHT，命令取值为：ON，单击“立即发送”，我们可以看到路灯被点亮。

----结束

使用虚拟设备在线调试

步骤1 在智慧路灯产品的开发空间，选择“在线调试”，单击“新增测试设备”。

步骤2 在新增测试设备页面，选择“模拟设备”，然后单击“确定”，创建一个虚拟设备。虚拟设备名称包含“Simulator”字样，每款产品下只能创建一个虚拟设备。

图 2-73 在线调试-创建虚拟设备



步骤3 单击“调试”，进入调试界面。

图 2-74 在线调试-进入调试



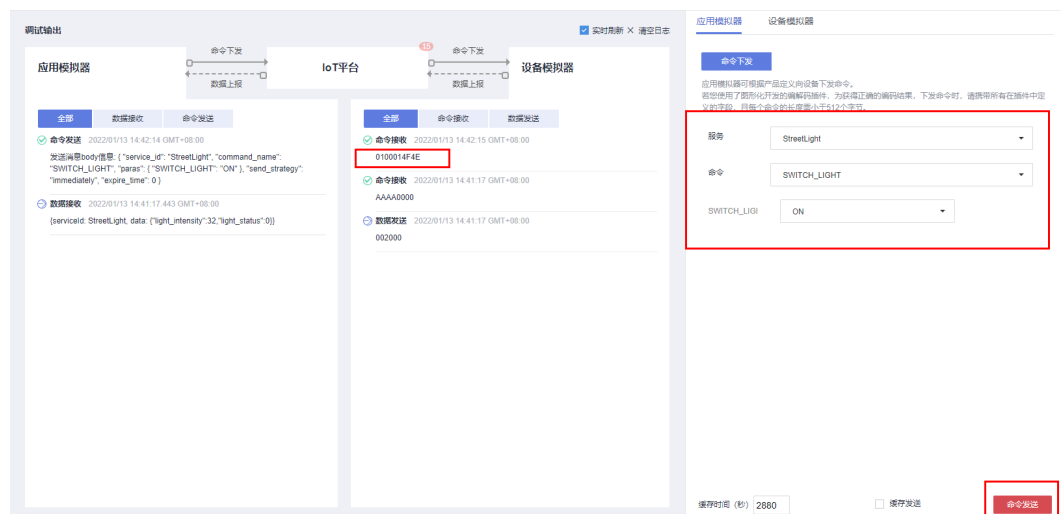
步骤4 模拟设备数据上报场景，在设备模拟器中，输入十六进制码流：002000（00为地址域measgId，长度为1个字节；20表示光照强度，长度为1个字节；00表示路灯开关状态为关闭状态，长度为1个字节），然后单击“发送”，我们可以在应用模拟器中看到转换为JSON格式的数据为：“Light_Intensity”:32, “Light_Status”:0。

图 2-75 在线调试-模拟数据上报



步骤5 模拟远程下发控制命令场景，在应用模拟器中，选择服务：StreetLight，命令：SWITCH_LIGHT，命令取值为：ON，单击“立即发送”，我们可以在设备模拟器中看到转换为十六进制的码流：0100014F4E（经ASCLL码转为为十六进制）。

图 2-76 在线调试-模拟命令下发



----结束

使用离线模拟器调试

NB-IoT设备模拟器用于模拟以LwM2M/CoAP协议接入物联网平台的NB设备，实现数据上报和命令下发功能。

步骤1 获取2中注册设备时的设备标识码，密钥。

步骤2 下载并解压NB-IoT设备模拟器，然后双击“NB-IoTDeviceSimulator_zh.jar”，运行模拟器。

说明

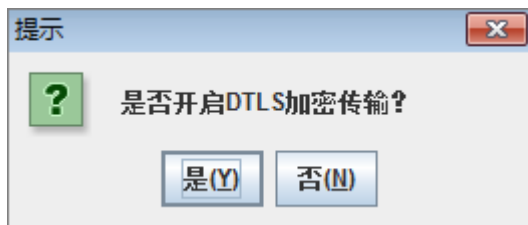
- 推荐在Windows系统下运行，暂不支持MAC OS。
- 请确保已经安装jdk，否则，无法运行jar文件。

Name	Size	Packed	Type	Modified	CRC32
images			文件夹	2020/10/26 9...	
NB-IoTDeviceSimulator_zh.jar	4,464,721	4,080,631	Executable Jar File	2017/8/7 14:07	B47CBF61
Californium.properties	1,203	613	PROPERTIES 文件	2017/7/29 12...	01A69483
setting.properties	111	108	PROPERTIES 文件	2017/8/7 15:21	3CEB9C...

NB-IoT设备模拟器的文件说明如下：

- **NB-IoTDeviceSimulator_zh.jar**：中文版模拟器。
- **Californium.properties**：模拟器的配置文件。
- **setting.properties**：设备模拟器接入物联网平台的配置文件。

步骤3 模拟器启动后，会提示“是否开启DTLS加密传输？”。若**注册设备**时，没有填写密钥，单击“否”，若**注册设备**时，已填写密钥，单击“是”。

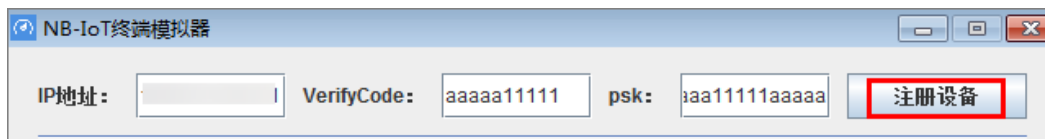


步骤4 填写“IP地址”、“VerifyCode”和“PSK”，单击“注册设备”，将模拟器与物联网平台进行绑定。

注：如果未开启DTLS加密传输，则无需填写密钥。

如下信息根据实际情况填写：

- IP地址：**物联网平台**的接入域名（也可以使用IP，通过ping命令“ping 域名”可以获得IP地址）。
- VerifyCode：设备标识码，如：aaaaa11111。
- PSK：注册设备时预置的密钥，如：aaaaa11111aaaaa。



设备绑定成功，可以回到管理控制台，选择“设备 > 所有设备”，看见设备“aaaaa11111”显示“在线”，表示模拟器绑定成功。

- 步骤5** 模拟设备数据上报场景，则在NB-IoT设备模拟器中，输入十六进制码流：001400（00为地址域measgeID，长度为1个字节；14表示光照强度，长度为1个字节；00表示路灯开关状态为关闭状态，长度为1个字节），然后单击“发送数据”。

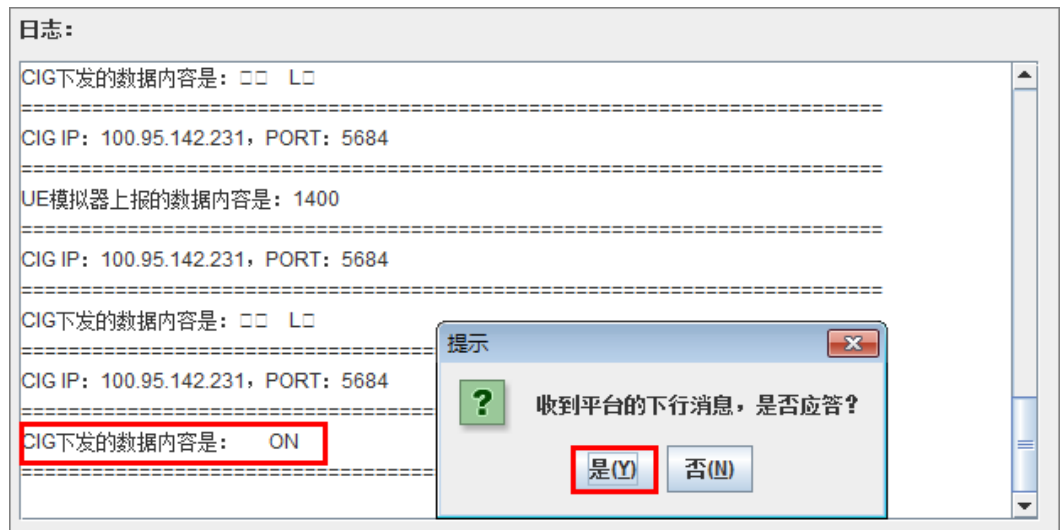


数据上报成功，可以回到管理控制台，在“aaaaa11111”设备详情页面中查看到最新上报数据为：“Light_Intensity”: 20, “Light_Status”: 0。

- 步骤6** 模拟远程下发控制命令场景，在管理控制台，找到对应的产品，单击产品进入产品界面，选择“在线调试”页签，单击设备“aaaaa11111”右边的“调试”，进入调试界面。

在应用模拟器中，选择服务：StreetLight，命令：SWITCH_LIGHT，命令取值为：ON，单击“立即发送”。

我们可以在“日志”栏看到CIG下发的数据内容：ON，且模拟器会提示“收到平台的下行消息，是否应答？”，单击“是”，可以在管理控制台的应用模拟器看到命令状态为“已送达”。



📖 说明

由于插件中的string类型是按照ascii码解析的，因此需要下发可显示字符，非可显示字符在模拟器中不会显示。

----结束

2.6 通过协议转换网关实现泛协议设备接入

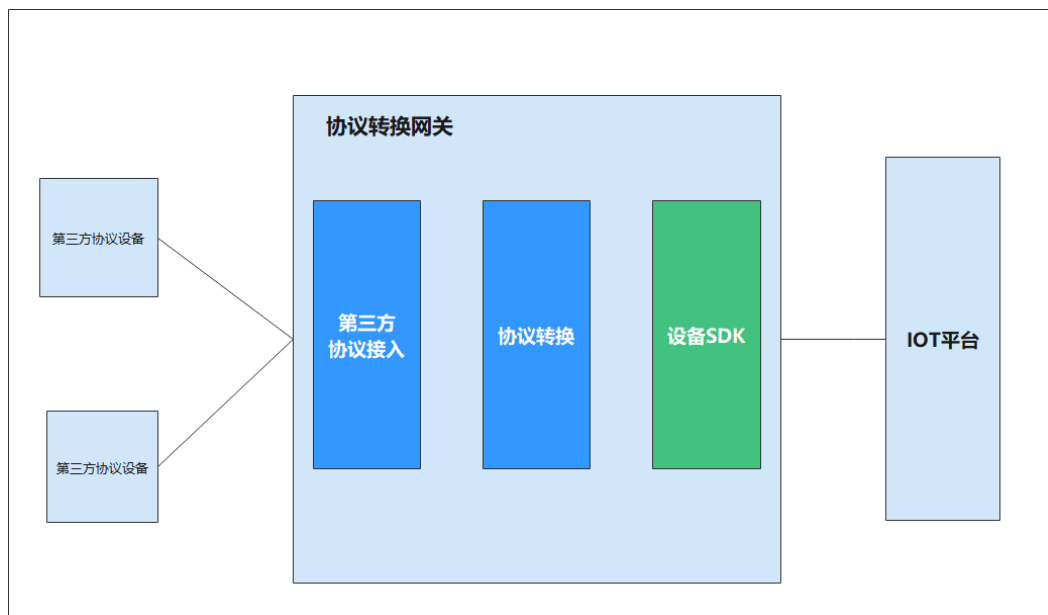
场景说明

目前平台只支持MQTT/HTTP/LWM2M等标准协议接入，如果设备是其他协议（统称为第三方协议），怎么接入平台？

由于第三方协议设备不能直接接入平台，需要在平台外部完成协议转换。我们推荐使用网关来完成协议转换，将第三方协议转成MQTT协议。我们把用来进行协议转换的网关叫做协议转换网关。

实现原理

方案总体架构框图如下：

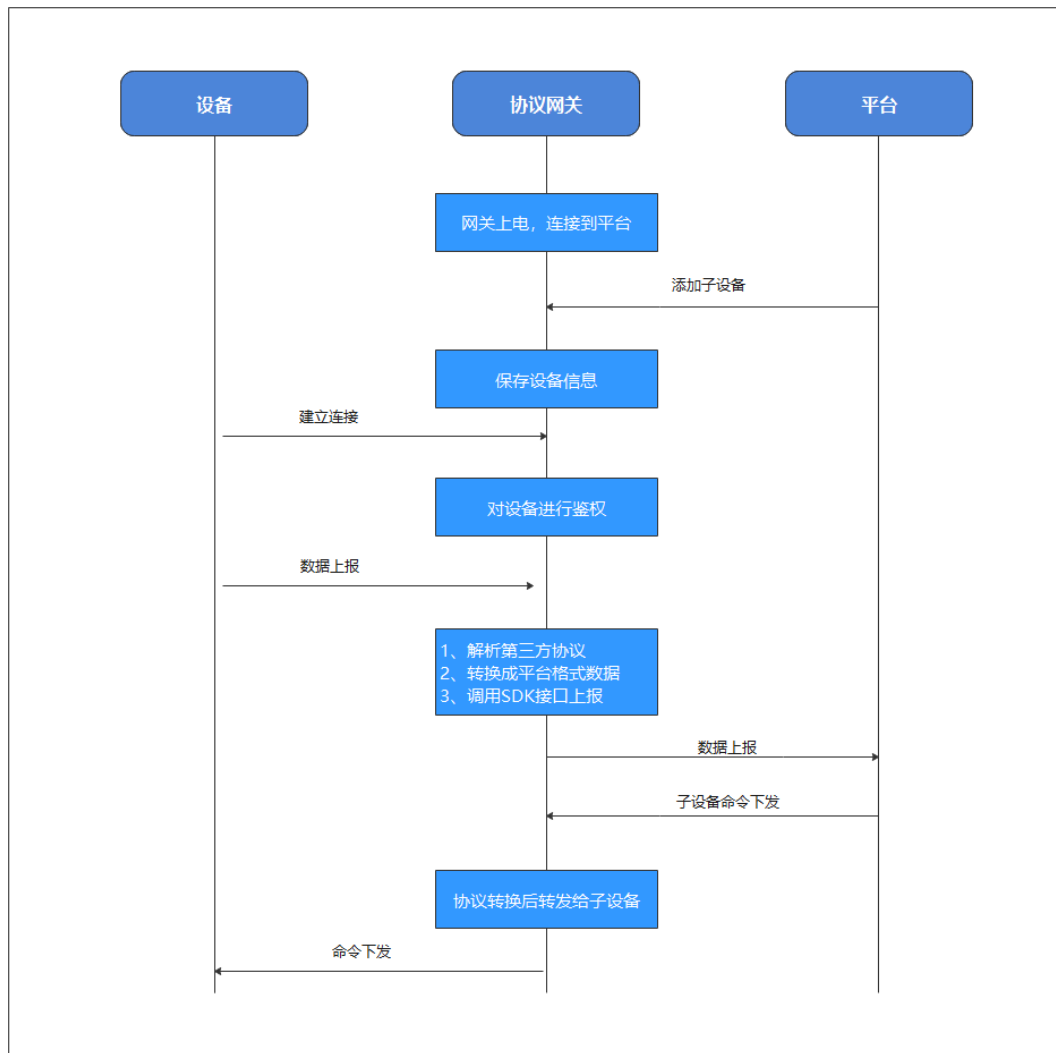


协议转换网关是一个网关，可以部署在云上或者本地。第三方协议设备作为协议转换网关的子设备接入平台。

协议转换网关一般由三部分组成：

1. 第三方协议接入。完成第三方协议的解析，鉴权。
2. 协议转换。负责完成第三方协议数据和平台格式数据的互相转换。
 - 上行：把第三方协议数据转成平台格式数据，并调用设备SDK接口进行上报。
 - 下行：收到平台下行数据时，转换为第三方协议数据转发给第三方协议设备。
3. 设备SDK。即平台提供的设备接入SDK，提供了网关的通用功能实现，用户可以在此基础上实现自己的网关。

业务流程



1. 在物联网平台上注册网关，详细方法请参考[设备注册](#)。
2. 网关上电，连接到平台，连接所需的鉴权参数在[注册网关](#)时获取。
3. 用户在平台上注册子设备时，平台下发添加子设备事件到网关。网关收到后，保存子设备信息到本地并持久化（SDK提供了默认的持久化实现，用户可以自定义扩展）。
4. 第三方协议设备连接到网关。网关根据子设备信息对设备进行鉴权。
5. 设备上报数据到网关。网关转换为平台格式数据后，调用SDK的上报子设备属性/消息的接口上报给平台。
6. 平台向设备下发命令。网关收到后，转换为第三方协议，转发给子设备。设备收到后对命令进行处理。

协议转换网关的具体实现

网关具体实现和使用方法参见 [Java版本](#)、[C版本](#)。

2.7 基于 MQTT.fx 的 X.509 证书接入

本文档以MQTT.fx为例，介绍以MQTT原生协议接入物联网平台。MQTT.fx是目前主流的MQTT客户端，可以快速验证是否可以与物联网平台服务交互发布或订阅消息。

X.509是一种用于通信实体鉴别的数字证书，物联网平台支持设备使用自己的X.509证书进行认证鉴权。使用X.509认证技术时，设备无法被仿冒，避免了密钥被泄露的风险。

前提条件

- 已注册华为云官方账号。未注册可单击[注册页面](#)完成注册。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，单击[管理控制台](#)后开通该服务。

限制说明

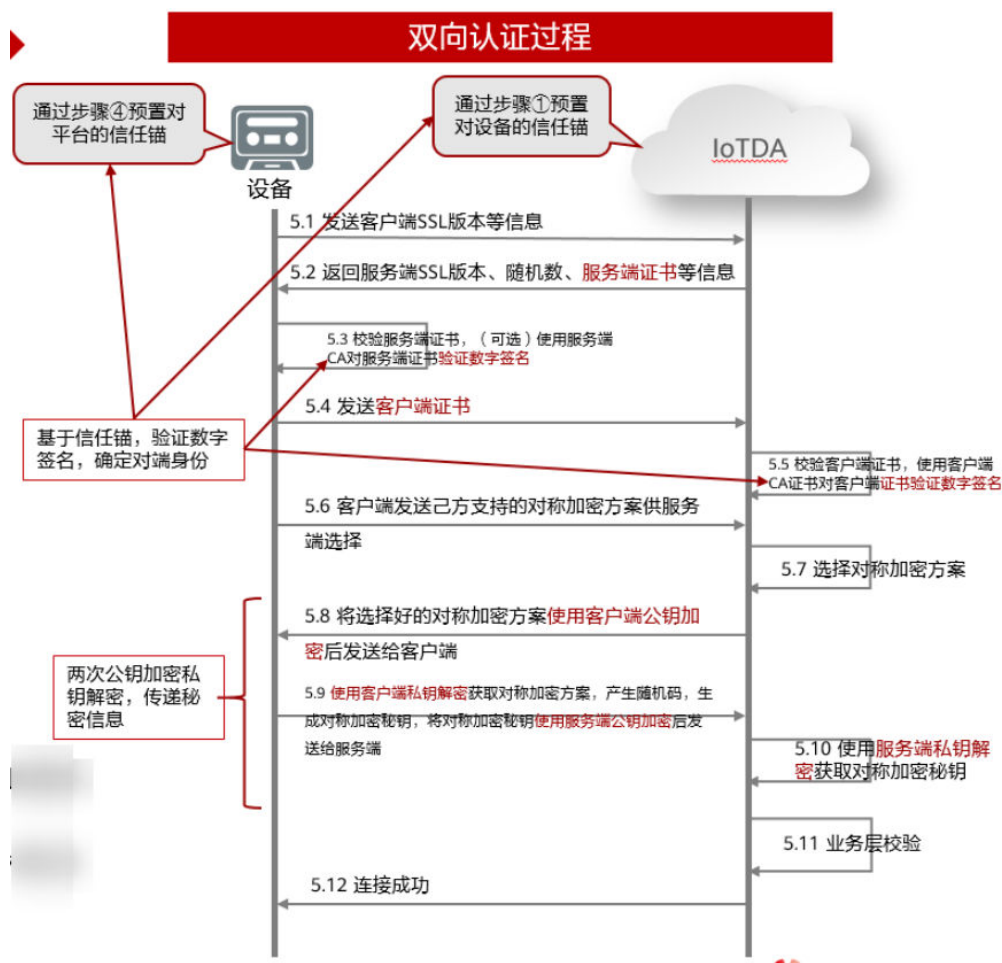
- 当前只有通过MQTT接入的设备支持使用X.509证书进行设备身份认证。
- 每个用户最多上传100个设备CA证书。

X509 认证步骤

- 证书双向认证完整业务流程为：



- 证书双向认证过程为：



获取设备接入信息

在设备接入服务控制台获取设备接入平台所需的信息。

步骤1 访问**设备接入服务**，单击**管理控制台**进入控制台。

步骤2 单击**管理控制台**进入控制台，单击左侧导航栏的“总览”，单击“实例基本信息 - 接入信息”查看设备接入信息，记录域名和端口。

图 2-77 进入平台接入地址



说明

针对不支持用域名接入的设备，通过在cmd命令框中执行“ping 域名”获取IP地址，用IP地址接入平台。由于IP地址不固定，您需要将IP地址做成可配置项。

----结束

创建产品

步骤1 登录**管理控制台**，单击左侧导航栏“产品”，单击页面左侧的“创建产品”。

图 2-78 产品-创建产品



步骤2 根据页面提示填写参数，然后单击“确定”。

基本信息	
所属资源空间	平台自动将新创建的产品归属在默认资源空间下。如需归属在其他资源空间下，下拉选择所属的资源空间。如无对应的资源空间，请先创建 资源空间 。
产品名称	自定义。支持字母、数字、下划线（_）、连字符（-）的字符组合。
协议类型	选择MQTT。
数据格式	选择JSON。
所属行业	请根据实际情况填写。若使用平台预置的产品模型，请根据产品模型所属的行业填写。没有填无。
设备类型	请根据实际情况填写。若使用平台预置的产品模型，会自动关联设备类型，不需要再输入设备类型。
高级配置	
产品ID	定制ProductID，用于唯一标识一个产品。如果携带此参数，平台将产品ID设置为该参数值；如果不携带此参数，产品ID在物联网平台创建产品后由平台分配获得。

图 2-79 创建产品-MQTT

创建产品

* 所属资源空间 ?

如需创建新的资源空间，您可前往当前实例详情创建

* 产品名称

协议类型 ?

* 数据格式 ?

设备类型选择

* 设备类型 ?

高级配置 ▲ 定制ProductID | 备注信息

产品ID ?

产品描述 0/128

---结束

开发产品模型

- 步骤1** 找到新增的产品，单击“产品”进入产品界面。
- 步骤2** 在产品详情“模型定义”页面，单击“自定义模型”，配置产品的服务。
- 步骤3** 新增服务类型“Connectivity”。
- 在“添加服务”页面，填写“服务ID”、“服务类型”和“服务描述”，单击“确定”。
 - “服务ID”：Connectivity
 - “服务类型”：建议和服务ID保持一致
 - “服务描述”：Connectivity

图 2-80 添加服务

2. 在“Connectivity”的下拉菜单下单击“添加属性”，填写相关信息后，单击“确定”。
 - “属性名称”：dailyActivityTime
 - “数据类型”：int（整型）
 - “访问权限”：可读
 - “取值范围”：0~65535
 - “步长”：1
 - “单位”：s

图 2-81 添加属性

修改属性 ×

* 属性名称

属性描述 0/128

* 数据类型

* 访问权限 可读 可写

* 取值范围 -

步长

单位

步骤4 新增服务类型“Battery”。

1. 在“模型定义”下单击“添加服务”，填写“服务ID”、“服务类型”和“服务描述”，单击“确定”。
 - “服务ID”：Battery
 - “服务类型”：建议和服务ID保持一致
 - “服务描述”：Battery

图 2-82 添加服务

2. 在““Battery””的下拉菜单下单击“添加属性”，填写相关信息后，单击“确定”。
 - “属性名称”：batteryLevel
 - “数据类型”：int（整型）
 - “访问权限”：可读
 - “取值范围”：0-100
 - “步长”：1
 - “单位”：%

图 2-83 添加属性

修改属性

* 属性名称

属性描述
0/128

* 数据类型

* 访问权限 可读 可写

* 取值范围 -

步长

单位

----结束

上传设备 CA 证书

步骤1 在左侧导航栏选择“设备 > 设备CA证书”，单击右上角的“上传证书”。

步骤2 在弹出的对话框中，单击“添加文件”，然后单击“确定”。

图 2-84 上传证书

上传证书

* CA证书 ?

📖 说明

- 设备CA证书由设备厂商提供，调测时可[自行制作调测证书](#)，商用时建议更换为商用证书，否则会带来安全风险。
- CA证书具有一个过期日期，在该日期后，这些证书将无法用于验证服务器的证书；请在CA证书的过期日期前替换这些证书，以确保设备可以正常的连接到IoT平台。

----结束

制作设备 CA 调测证书

本文以Windows环境为例，介绍通过Openssl工具制作调测证书的方法，生成的证书为PEM编码格式的证书。

1. 在浏览器中访问[这里](#)，下载并进行安装OpenSSL工具。
2. 以管理员身份运行cmd命令行窗口。
3. 执行cd c:\openssl\bin（请替换为openssl实际安装路径），进入openssl命令视图。
4. 执行以下命令生成密钥对。
openssl genrsa -out rootCA.key 2048
5. 执行以下命令，使用密钥对中的私有密钥生成 CA 证书。
openssl req -x509 -new -nodes -key rootCA.key -sha256 -days 1024 -out rootCA.pem

系统提示您输入如下信息，所有参数可以自定义。

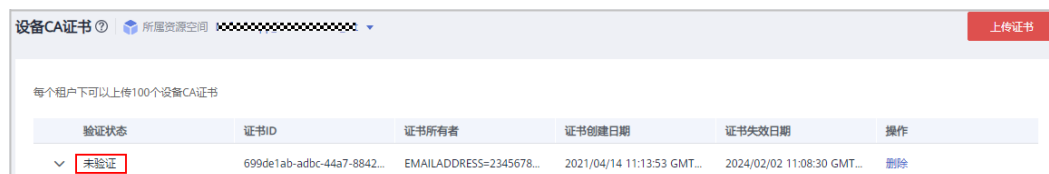
- Country Name (2 letter code) [AU]: 国家，如CN。
- State or Province Name (full name) []: 省份，如GD。
- Locality Name (for example, city) []:城市，如SZ。
- Organization Name (for example, company) []: 组织，如Huawei。
- Organizational Unit Name (for example, section) []: 组织单位，如IoT。
- Common Name (e.g. server FQDN or YOUR name) []: 名称，如zhangsan。
- Email Address []:邮箱地址，如1234567@163.com。

在openssl安装目录的bin文件夹下，获取生成的CA证书（rootCA.pem）。

上传验证证书

如果上传的是调测证书，上传后证书状态显示为“未验证”，您需要上传验证证书，来证明您拥有该CA证书。

图 2-85 设备 CA 证书



验证证书是由设备CA证书对应的私钥创建的，请参考如下操作制作验证证书。

步骤1 获取验证证书的验证码

图 2-86 选择上传验证证书

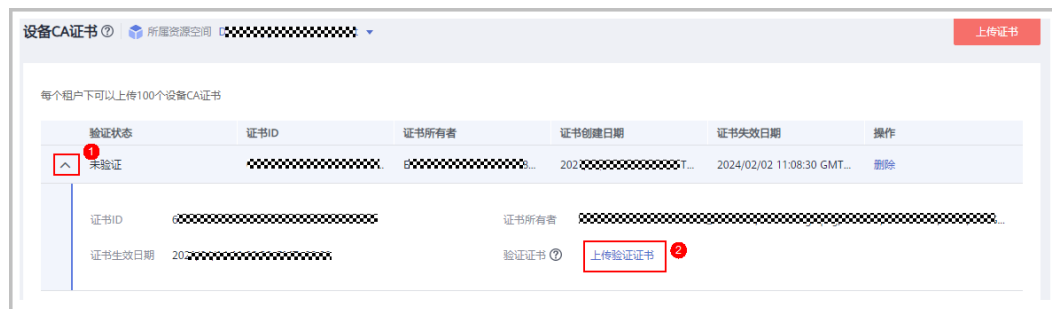


图 2-87 获取验证码



步骤2 执行如下命令为私有密钥验证证书生成密钥对。

```
openssl genrsa -out verificationCert.key 2048
```

步骤3 执行如下命令为私有密钥验证证书创建CSR（Certificate Signing Request）。

```
openssl req -new -key verificationCert.key -out verificationCert.csr
```

系统提示您输入如下信息，**Common Name**填写为验证证书的验证码，其他参数自定义。

- Country Name (2 letter code) [AU]: 国家，如CN。
- State or Province Name (full name) []: 省份，如GD。
- Locality Name (for example, city) []: 城市，如SZ。
- Organization Name (for example, company) []: 组织，如Huawei。
- Organizational Unit Name (for example, section) []: 组织单位，如IoT。
- Common Name (e.g. server FQDN or YOUR name) []: 验证证书的验证码，请参考**步骤1**获取。
- Email Address []: 邮箱地址，如1234567@163.com。
- Password []: 密码，如1234321。
- Optional Company Name []: 公司名称，如Huawei。

步骤4 执行以下命令使用CSR创建私有密钥验证证书。

```
openssl x509 -req -in verificationCert.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out verificationCert.pem -days 500 -sha256
```

在openssl安装目录的bin文件夹下，获取生成的验证证书（verificationCert.pem）。

步骤5 选择对应证书，单击  然后单击“上传验证证书”。

图 2-88 上传验证证书



步骤6 在弹出的对话框中，单击“添加文件”，然后单击“确定”。

图 2-89 完成验证



上传验证证书后，证书状态变为“已验证”，表明您拥有该CA证书。

---结束

预置 X.509 证书

在注册X.509设备之前，您需要在设备侧预置CA机构签发的X.509证书。

说明

X.509证书由CA机构签发，若没有CA机构签发的商用证书，您可以自己[制作设备CA调测证书](#)。

制作X.509调测证书

1. 以管理员身份运行cmd命令行窗口，执行`cd c:\openssl\bin`（请替换为openssl实际安装路径），进入openssl命令视图。
2. 执行如下命令生成密钥对。

```
openssl genrsa -out deviceCert.key 2048
```
3. 执行如下命令为设备证书创建CSR（Certificate Signing Request）。

```
openssl req -new -key deviceCert.key -out deviceCert.csr
```

系统提示您输入如下信息，所有参数可以自定义。

 - Country Name (2 letter code) [AU]: 国家，如CN。

- State or Province Name (full name) []: 省份，如GD。
 - Locality Name (for example, city) []: 城市，如SZ。
 - Organization Name (for example, company) []: 组织，如Huawei。
 - Organizational Unit Name (for example, section) []: 组织单位，如IoT。
 - Common Name (e.g. server FQDN or YOUR name) []: 名称，如zhangsan。
 - Email Address []: 邮箱地址，如1234567@163.com。
 - Password[]: 密码，如1234321。
 - Optional Company Name[]: 公司名称，如Huawei。
4. 执行以下命令使用CSR创建设备证书。
- ```
openssl x509 -req -in deviceCert.csr -CA rootCA.pem -CAkey rootCA.key -CAcreateserial -out deviceCert.pem -days 500 -sha256
```
- 在openssl安装目录的bin文件夹下，获取生成的设备证书（deviceCert.pem）。

## 注册 X.509 证书认证的设备

**步骤1** 访问[设备接入服务](#)，单击[管理控制台](#)进入设备接入控制台。

**步骤2** 在左侧导航栏选择“设备 > 所有设备”，单击右上角“注册设备”，按照如下表格填写参数后，单击“确定”。

图 2-90 单设备注册

| 参数名称   | 说明           |
|--------|--------------|
| 所属资源空间 | 选择设备所属的资源空间。 |

| 参数名称   | 说明                                                                                                                                                                                                                                                                                                                                                                                            |
|--------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 所属产品   | 选择设备所属的产品。<br>如没有，请先 <a href="#">创建产品</a> 。                                                                                                                                                                                                                                                                                                                                                   |
| 设备标识码  | 即node_id，填写为设备的IMEI、MAC地址或Serial No；若没有真实设备，填写自定义字符串，由英文字母和数字组成                                                                                                                                                                                                                                                                                                                               |
| 设备名称   | 即device_name，可自定义。                                                                                                                                                                                                                                                                                                                                                                            |
| 设备认证类型 | X.509证书：设备使用X.509证书验证身份。                                                                                                                                                                                                                                                                                                                                                                      |
| 指纹     | 当“设备认证类型”选择“X.509证书”时填写，导入 <a href="#">设备侧预置的设备证书</a> 对应的指纹，在OpenSSL执行 <code>openssl x509 -fingerprint -sha256 -in deviceCert.pem</code> 命令可查询。 <b>注：填写时需要删除冒号。</b><br><pre>[root@k8s-iot-wl2-2-jump cert1223]# openssl x509 -fingerprint -sha256 -in deviceCert.pem SHA256 Fingerprint=f7:91:90:45:BB:88:37:E6:A7:E7:70:4A:90:75:F3:87:DA:27:5B:7C:49:3E:FF:59:7A:6F:4F:08:4D:F8:54:E8</pre> |

----结束

## 连接鉴权

参考[连接鉴权](#)接口文档，使用MQTT.fx工具激活在物联网平台上注册的设备。

**步骤1** 下载[MQTT.fx](#)（默认是64位操作系统，如果是32位操作系统，单击此处下载[MQTT.fx](#)），安装MQTT.fx工具。

### 📖 说明

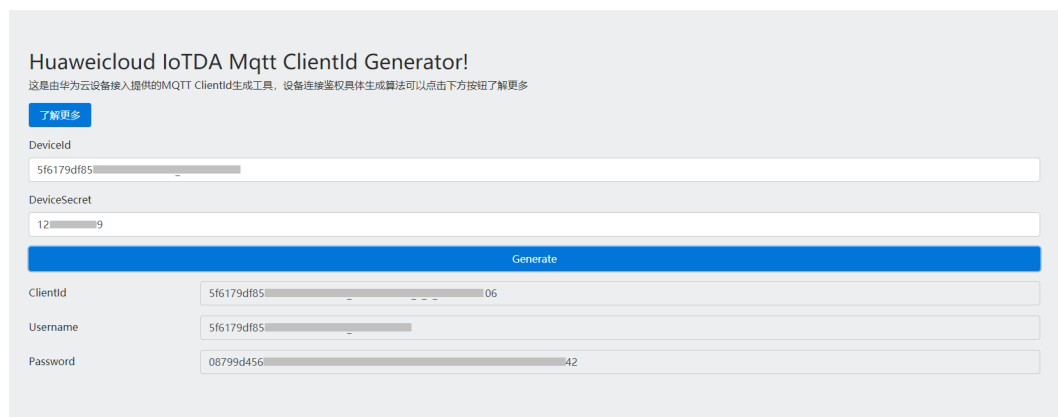
- 安装最新版MQTT.fx工具，可单击此处[下载](#)。
- MQTT.fx 1.7.0及旧版本对带有\$的主题（Topic）处理存在问题，请使用最新版本进行测试。

**步骤2** 访问[这里](#)，填写[注册设备](#)后生成的设备ID（DeviceId）和密钥（DeviceSecret），生成连接信息（ClientId、Username、Password）。

### 📖 说明

DeviceSecret在此场景下可随意填写，如12345678。

图 2-91 获取 ClientId





| 参数       | 必选/<br>可选 | 类型          | 参数描述                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|----------|-----------|-------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ClientId | 必选        | String(256) | <p>一机一密的设备clientId由4部分组成：设备ID、设备身份标识类型、密码签名类型、时间戳，通过下划线“_”分隔。</p> <ul style="list-style-type: none"> <li>● <b>设备ID</b>：指设备在平台成功注册后生成的唯一设备标识，通常由设备的产品ID和设备的NodeId通过分隔符“_”拼装而来。</li> <li>● <b>设备身份标识类型</b>：固定值为0，表示设备ID。</li> <li>● <b>密码签名类型</b>：长度1字节，当前支持2种类型： <ul style="list-style-type: none"> <li>- “0”代表HMACSHA256不校验时间戳。</li> <li>- “1”代表HMACSHA256校验时间戳。</li> </ul> </li> <li>● <b>时间戳</b>：为设备连接平台时的UTC时间，格式为YYYYMMDDHH，如UTC时间2018/7/24 17:56:20 则应表示为2018072417。</li> </ul> |
| Username | 必选        | String(256) | 设备ID。                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

设备通过MQTT协议的connect消息进行鉴权，对于构造clientId的各个部分信息都必须包括进去，平台收到connect消息时，会判断设备的鉴权类型和密码摘要算法。

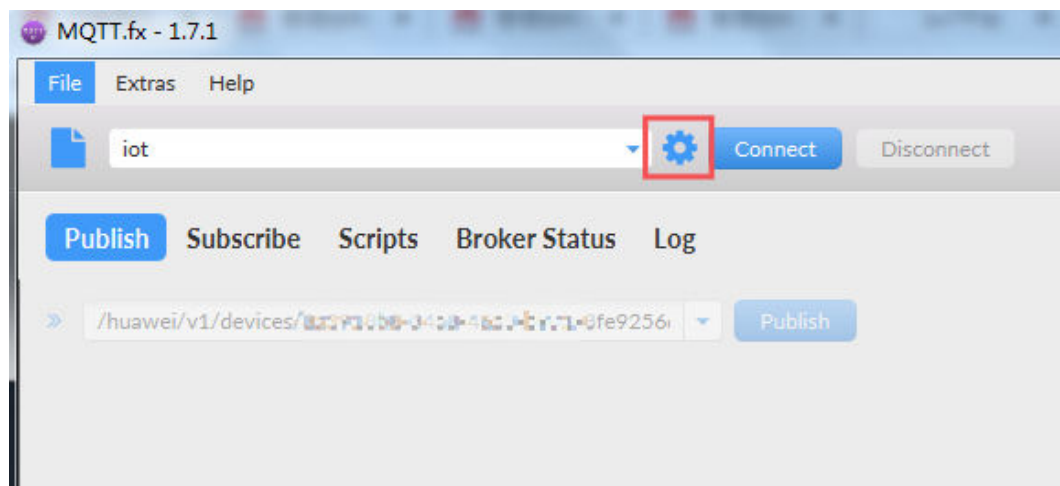
使用生成工具生成的clientId格式，默认不校验时间戳：设备ID\_0\_0\_时间戳。

- 当采用“HMACSHA256”校验时间戳方式时，会先校验消息时间戳与平台时间是否一致，再判断密码是否正确。
- 当采用“HMACSHA256”不校验时间戳方式时，鉴权消息也必须带时间戳，但不检验时间是否准确，仅判断密码是否正确。

connect消息鉴权失败时，平台会返回错误，并自动断开MQTT链路。

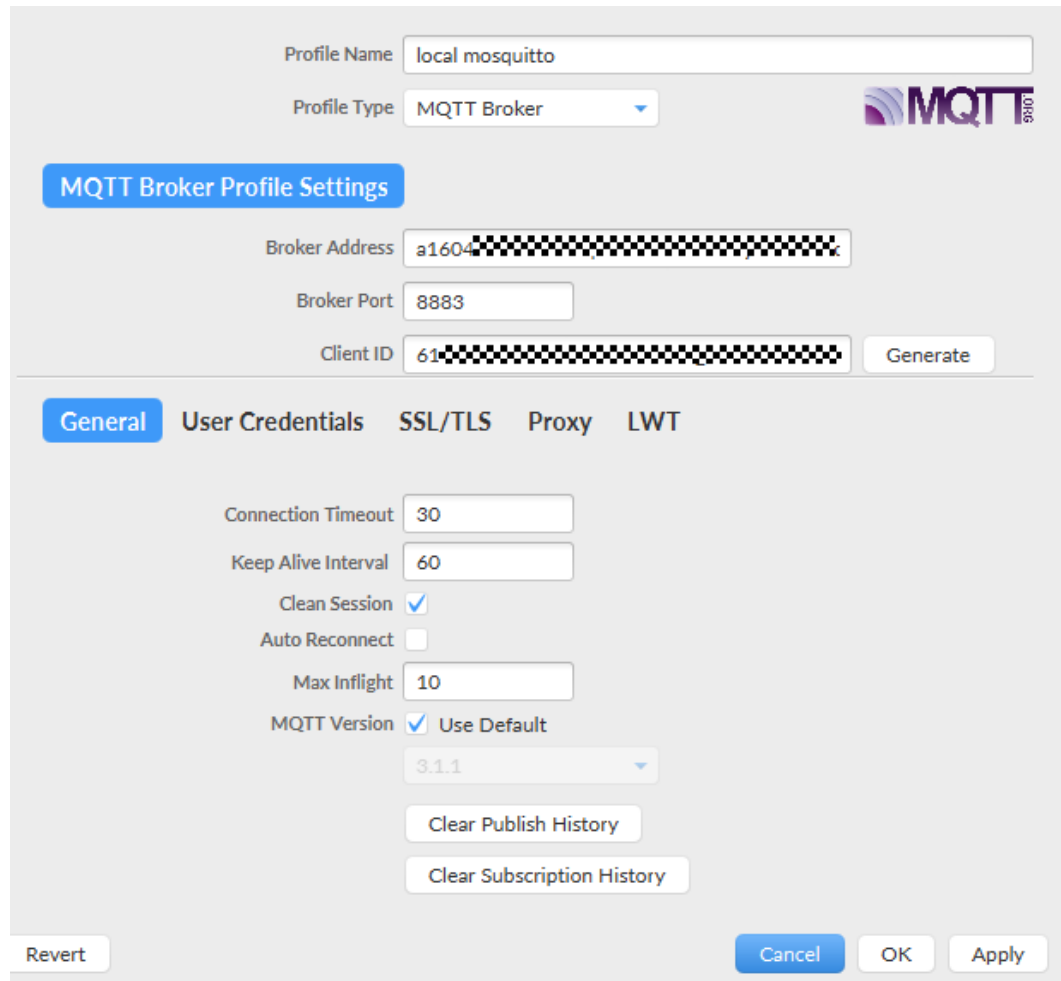
**步骤3** 打开MQTT.fx软件，单击“设置”图标。

图 2-92 设置



**步骤4** 填写“Connection Profile”相关信息。

**图 2-93** “General” 可以使用工具默认信息



| 参数名称           | 说明                                                                     |
|----------------|------------------------------------------------------------------------|
| Broker Address | 填写从设备接入服务控制台获取的 <b>设备对接地址</b> ，此地址为域名信息。不能通过域名接入的设备，填写IP地址，详细请参考2中获取的。 |
| Broker Port    | 为8883。                                                                 |
| Client ID      | 设备cliendID，请参考2中获取。                                                    |

**步骤5** 单击“User Credentials”填写“User Name”。

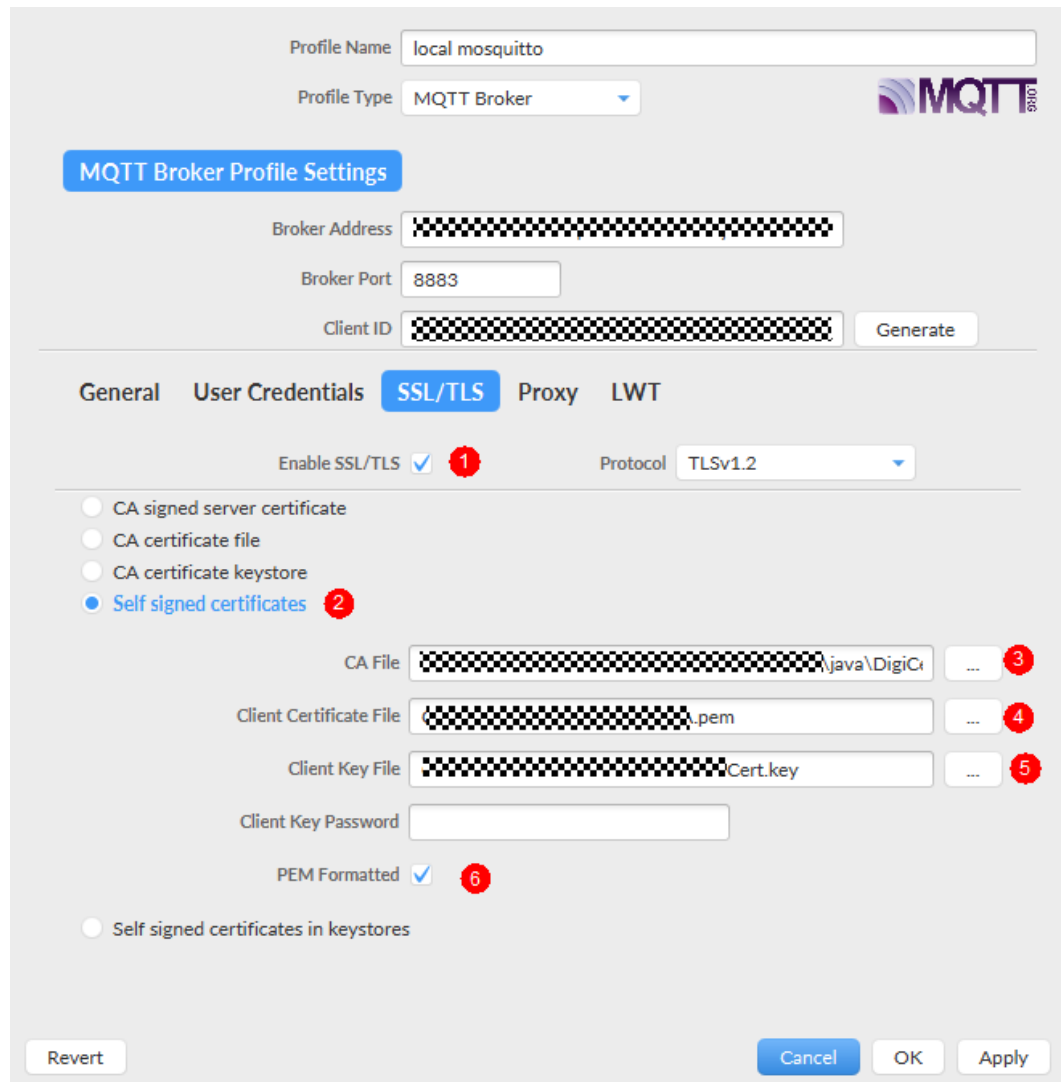
图 2-94 填写设备 ID

The screenshot shows the 'MQTT Broker Profile Settings' dialog box. At the top, 'Profile Name' is 'local mosquitto' and 'Profile Type' is 'MQTT Broker'. Below this is a blue button labeled 'MQTT Broker Profile Settings'. The 'Broker Address' field is obscured by a checkered pattern. 'Broker Port' is '8883'. 'Client ID' is also obscured by a checkered pattern, with a 'Generate' button to its right. Below these fields are tabs for 'General', 'User Credentials', 'SSL/TLS', 'Proxy', and 'LWT'. The 'User Credentials' tab is active, showing 'User Name' (obscured by a checkered pattern) and 'Password' (empty). At the bottom are buttons for 'Revert', 'Cancel', 'OK', and 'Apply'.

| 参数名称      | 说明                 |
|-----------|--------------------|
| User Name | 即设备ID，请参考2中获取。     |
| Password  | 使用X.509证书认证时不需要填写。 |

**步骤6** 单击“SSL/TLS”配置鉴权参数，然后单击““Apply””。选择开启“SSL/TLS”，勾选“Self signed certificates”，配置相关证书内容。

图 2-95 填写“SSL/TLS”相关参数



### 说明

CA File为对应的CA证书。下载并获取证书（加载pem格式的证书），获取证书请根据需要在[资源获取](#)里下载。

Client Certificate File为设备的设备证书（deviceCert.pem）。

Client Key File为设备的私钥（deviceCert.key）。

**步骤7** 单击“Connect”，设备鉴权成功后，在物联网平台可以看到设备处于在线状态。

图 2-96 设备状态



----结束

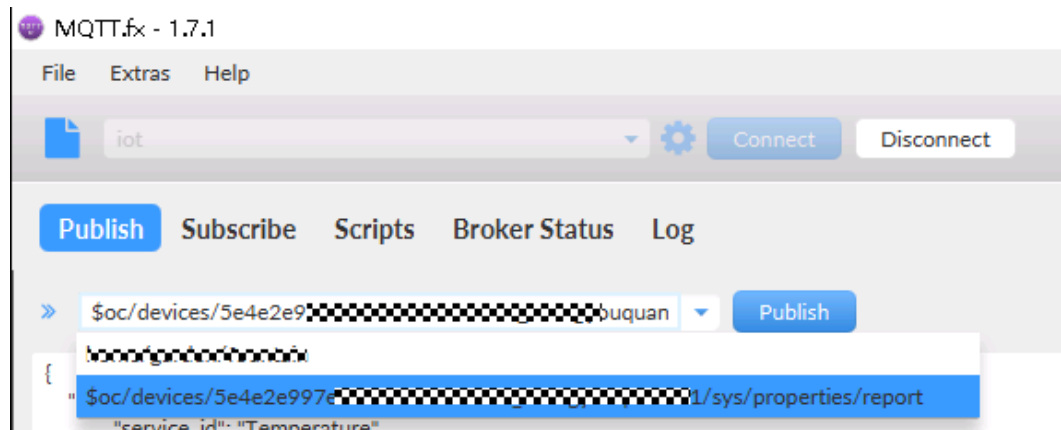
## 上报数据

参考[设备属性上报](#)接口文档，使用MQTT.fx工具向物联网平台上报数据。

设备若通过MQTT通道上报数据，需要发给指定的topic，上报消息的topic格式为：“\$oc/devices/{device\_id}/sys/properties/report”，其中“device\_id”的值，为设备注册成功后返回的“device\_id”值。

**步骤1** 填写接口地址，此处以“\$oc/devices/{device\_id}/sys/properties/report”为例，如“\$oc/devices/5e4e2e92ac-164aefa8fouquan1/sys/properties/report”。

图 2-97 填写接口地址



**步骤2** 填写上报的数据。

### 请求参数

| 字段名      | 必选/可选 | 类型                    | 参数描述                                 |
|----------|-------|-----------------------|--------------------------------------|
| services | 必选    | List<ServiceProperty> | 设备服务数据列表（具体结构参考下表ServiceProperty定义表） |

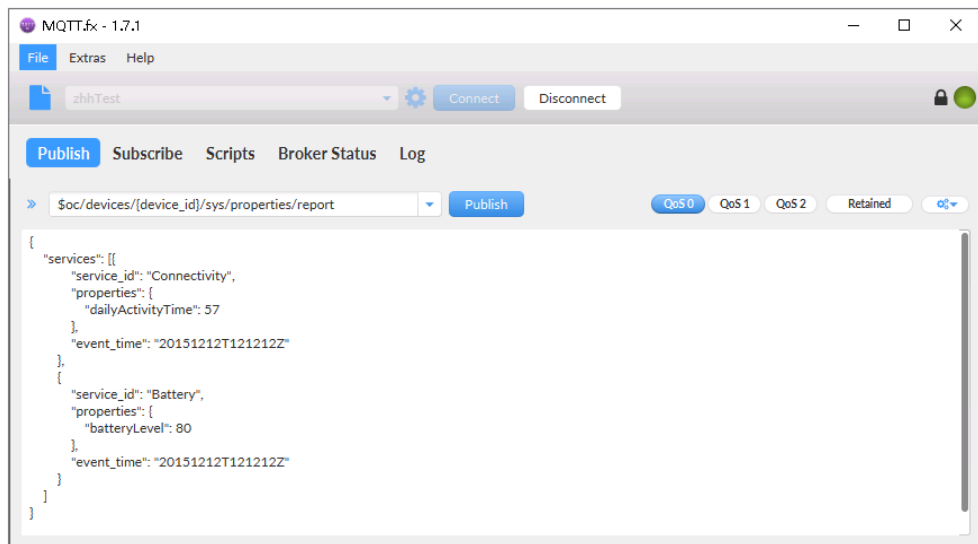
ServiceProperty定义表：

| 字段名        | 必选/可选 | 类型     | 参数描述                                                                                       |
|------------|-------|--------|--------------------------------------------------------------------------------------------|
| service_id | 必选    | String | 设备服务的ID。                                                                                   |
| properties | 必选    | Object | 设备服务的属性列表，具体字段在设备关联的产品模型中定义。                                                               |
| event_time | 可选    | String | 设备采集数据UTC时间（格式：yyyyMMddTHHmssZ），如：20161219T114920Z。<br>设备上报数据不带该参数或参数格式错误时，则数据上报时间以平台时间为准。 |

### 请求示例

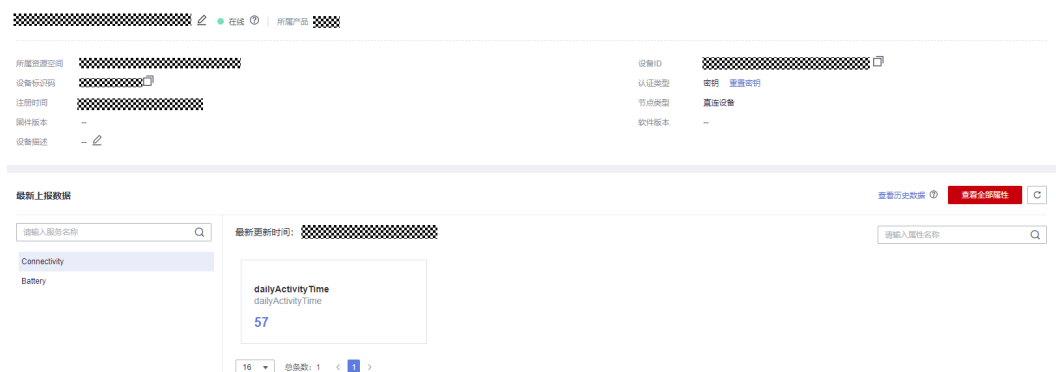
```
{
 "services": [
 {
 "service_id": "Connectivity",
 "properties": {
 "dailyActivityTime": 57
 },
 "event_time": "20151212T121212Z"
 },
 {
 "service_id": "Battery",
 "properties": {
 "batteryLevel": 80
 },
 "event_time": "20151212T121212Z"
 }
]
}
```

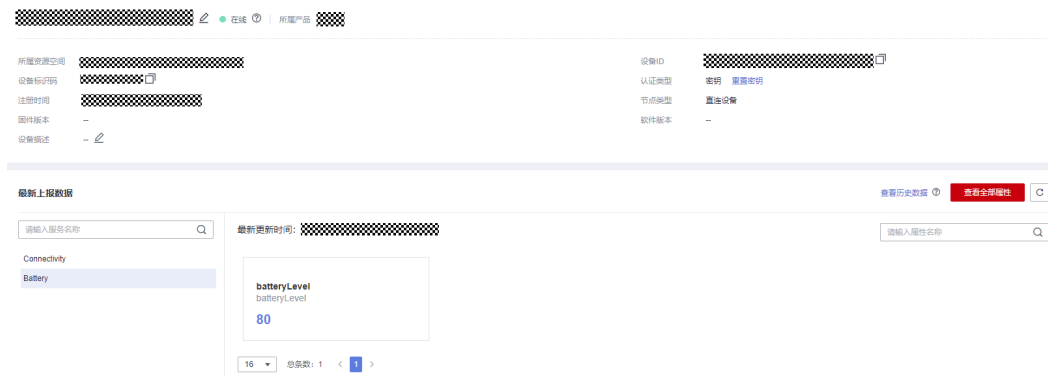
图 2-98 请求示例



步骤3 单击“Publish”，可以在物联网平台上查看设备是否成功上报数据。

图 2-99 上报数据





----结束

## 2.8 基于 BearPi-HM\_Nano 开发板和 OpenHarmony3.0 对接 IoTDA

本文主要介绍基于OpenHarmony 3.0来接入IoTDA，以BearPi-HM\_Nano开发板为例，使用huaweicloud\_iot\_link SDK对接华为云物联网平台的简单流程。

### 前提条件

- 已注册华为云官方账号。未注册可单击[注册页面](#)完成注册。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，单击“管理控制台”后开通该服务。

### 硬件环境

BearPi-HM\_Nano开发板、E53\_IA1扩展模块、Type-C数据线、PC。

### 软件环境

搭建环境可以选择基于IDE方式或者基于命令行方式，其中BearPi-HM\_Nano主控芯片为Hi3861，因此搭建环境时需要安装Hi3861对应环境。

#### 📖 说明

在“安装Hi3861开发板特有环境”的安装gcc\_riscv32步骤时，强烈建议直接下载gcc\_riscv32镜像，否则可能出现某些插件下载或安装失败。

#### 须知

开源代码量较大，下载时间可能较长，需预留好充足时间。

### 创建产品

- 步骤1** 登录[管理控制台](#)，单击左侧导航栏“产品”，单击页面左侧的“创建产品”。
- 步骤2** 根据页面提示填写参数，然后单击“确认”，完成产品的创建。

| 基本信息   |                                                                                      |
|--------|--------------------------------------------------------------------------------------|
| 所属资源空间 | 平台自动将新创建的产品归属在默认资源空间下。如需归属在其他资源空间下，下拉选择所属的资源空间。如无对应的资源空间，请先创建 <a href="#">资源空间</a> 。 |
| 产品名称   | 为产品命名。产品名称在相同资源空间有唯一性。长度不超过64，只允许中文、字母、数字、以及_?'#(),&%@!-等字符的组合。                      |
| 协议类型   | 选择MQTT。                                                                              |
| 数据格式   | 选择JSON。                                                                              |
| 所属行业   | 请根据实际情况选择。                                                                           |
| 设备类型   | 请根据实际情况选择。                                                                           |
| 高级配置   |                                                                                      |
| 产品ID   | 定制ProductID，用于唯一标识一个产品。如果携带此参数，平台将产品ID设置为该参数值；如果不携带此参数，产品ID在物联网平台创建产品后由平台分配获得。       |



图 2-100 创建产品-MQTT

**创建产品** ×

\* 所属资源空间 ?  ▼  
 如需创建新的资源空间, 您可前往当前实例详情创建

\* 产品名称

协议类型 ?  ▼

\* 数据格式 ?  ▼

设备类型选择 标准类型 自定义类型

\* 设备类型 ?

高级配置 ▲ 定制ProductID | 备注信息

产品ID ?

产品描述   
0/128

确定 取消

----结束

## 开发产品模型

**步骤1** 找到新增的产品, 单击“查看”进入产品界面。

**步骤2** 在产品详情“模型定义”页面, 直接上传模型文件[BearPi\\_Agriculture.zip](#)。

### 📖 说明

1. 在产品列表中, 单击对应的产品, 进入产品详情页。您可以查看产品ID、产品名称、设备类型、数据格式、厂商名称、所属资源空间、协议类型等产品基本信息。
2. 产品创建成功后, 您可以单击“删除”删除不再使用的产品。删除产品后, 该产品下的产品模型、编解码插件等资源将被清空, 请谨慎操作。

----结束

## 注册设备

**步骤1** 访问[设备接入服务](#), 单击[管理控制台](#)进入设备接入控制台。

**步骤2** 在左侧导航栏选择“设备 > 所有设备”, 单击右上角“注册设备”, 按照如下表格填写参数后, 单击“确定”。

图 2-101 单设备注册-MQTT

### 单设备注册

×

\* 所属资源空间 ?

\* 所属产品

\* 设备标识码 ?

设备ID ?

设备名称

设备描述  0/2,048

设备认证类型 ? 密钥 X.509证书

密钥

确认密钥

确定
取消

| 参数名称   | 说明                                                                                         |
|--------|--------------------------------------------------------------------------------------------|
| 所属资源空间 | 选择设备所属的资源空间。                                                                               |
| 所属产品   | 选择设备所属的产品。如没有，请先 <a href="#">创建产品</a> 。                                                    |
| 设备标识码  | 即node_id，填写为设备的IMEI、MAC地址或Serial No；若没有真实设备，填写自定义字符串，长度为4至64个字符，只允许字母、数字、下划线（_）、连接符（-）的组合。 |
| 设备名称   | 即device_name，可自定义。                                                                         |
| 设备认证类型 | 选择“密钥”。                                                                                    |
| 密钥     | 设备密钥，可自定义，不填写物联网平台会自动生成。                                                                   |

图 2-102 设备-注册设备成功



### 📖 说明

1. 设备注册成功后，请妥善保管好设备ID和密钥，用于设备接入平台认证。
2. 若密钥丢失，只能通过[重置密钥](#)生成新的密钥，无法找回注册设备时生成的密钥。
3. 用户可在[设备列表](#)删除不再使用的设备。删除设备不支持撤回，请谨慎操作。

----结束

## Huaweicloud\_iotlink SDK 使用

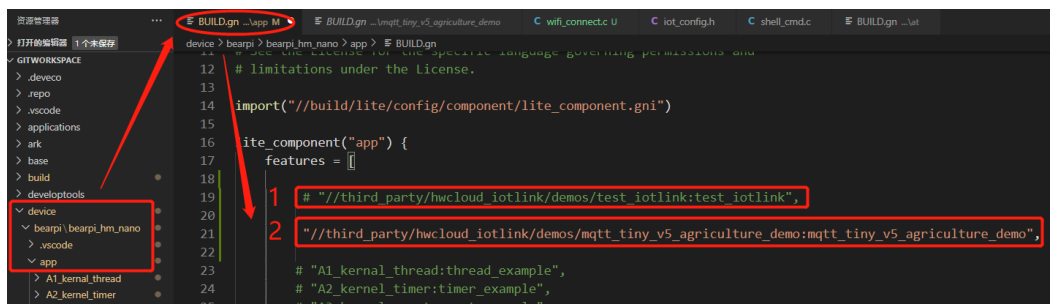
**步骤1** 下载源码：[oh3.0\\_hwcloud\\_iotlink-master.zip](#)。

**步骤2** 将上述源码直接拷贝到OpenHarmony源码根目录下的src->third\_party目录中，此处需要注意的是OpenHarmony和Huaweicloud\_iotlink SDK共同具备的三方库直接使用OpenHarmony的库文件，如JSON和mbedtls。

**步骤3** 在OpenHarmony 3.0源码device\bearpi\bearpi\_hm\_nano\app\BUILD.gn中添加如下代码，根据#是否注释来决定具体编译demo的选择。

```
"//third_party/hwcloud_iotlink/demos/mqtt_tiny_v5_agriculture_demo:mqtt_tiny_v5_agriculture_demo",
```

图 2-103 demo 的选择



说明

上图中2为使用MQTT对接华为云的demo，确认该demo的BUILD.gn文件，具体如图7所示，其中A区域包含开发板硬件和wifi相关的库文件，B区域包括MQTT对接华为云必须的一些库文件，如cjson，mqtt相关、osal相关以及配置库文件，C区域表示编译该文件时需要编译hwcloud\_iotlink库，编译是会按照指定路径找到该文件所必要的库以及C文件进行编译。

图 2-104 代码编译文件

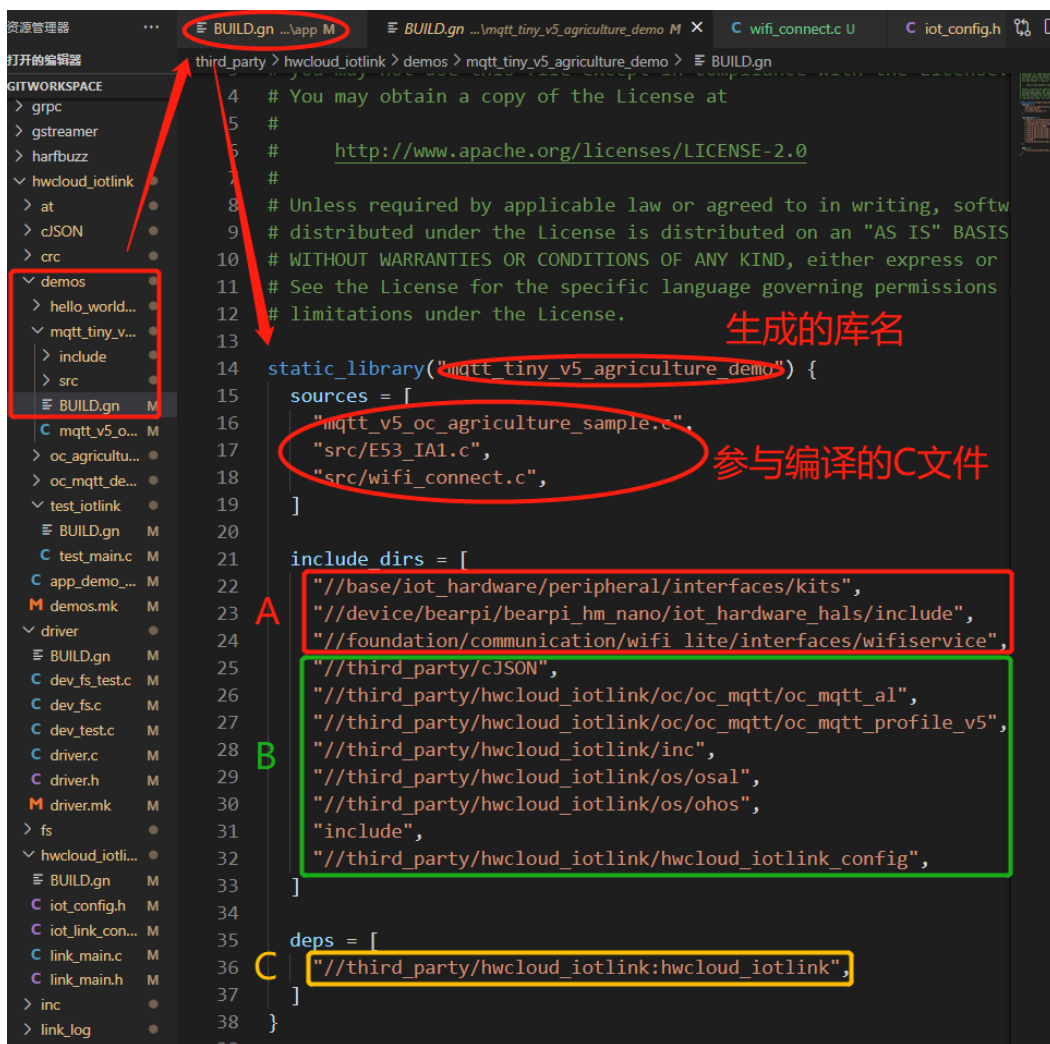


图 2-105 demo 的主函数文件

```

22
23 #include <dtls_al.h>
24 #include <mqtt_al.h>
25 #include <oc_mqtt_al.h>
26 #include <oc_mqtt_profile.h>
27 #include "E53_IA1.h"
28 #include "wifi_connect.h"
355
356 static void IotMainTaskEntry(void)
357 {
358 link_main_task_entry();
359 (void)osal_task_create("CloudMainTaskEntry",CloudMainTaskEntr
360 (void)osal_task_create("SensorTaskEntry",SensorTaskEntry,NULL
361 }
362
363 APP FEATURE_INIT IotMainTaskEntry;

```

**说明**

图8中A区域表示MQTT对接华为云所必须的一些库文件以及dtls的库文件，入口函数IotMainTaskEntry()中务必先调用link\_main\_task\_entry()函数，来确保osal的安装以及其他配置的初始化。

**步骤4 设置参数。**

图 2-106 参数修改

```

30 // 修改为自己的wifi热点
31 #define CONFIG_WIFI_SSID "your wifi hotspot name" // 修改为自己的wifi热点账号
32 #define CONFIG_WIFI_PWD "your wifi hotspot password" // 修改为自己的wifi热点密码
33
34 #define CONFIG_APP_SERVERIP "Access Address" // 替换为自己的对接地址
35
36 #define CONFIG_APP_SERVERPORT "1883"
37
38 #define CONFIG_APP_DEVICEID "your device id" // 替换为注册设备后生成的deviceid
39 #define CONFIG_APP_DEVICEPWD "your device password" // 替换为注册设备后生成的密钥
40
41 #define CONFIG_APP_LIFETIME 60 // seconds
42
43 #define CONFIG_QUEUE_TIMEOUT (5 * 1000)
44
45 #define MSGQUEUE_COUNT 16
46 #define MSGQUEUE_SIZE 10

```

**注意**

- (1) 如果需要对接上云需要根据自己的设备修改配网信息，wifi热点账号以及密码以及自己在云端注册的设备ID和设备密钥，注意此设备仅支持2.4G频段的wifi。
- (2) 对接地址修改为**管理控制台**界面“总览”的“接入信息”设备接入MQTT协议地址。
- (3) 编译和烧录，分为IDE方式和命令行方式，这里不一一介绍。
- (4) 烧录后，按下开发板RESET复位键，设备将启动。
- (5) 上述代码基于OpenHarmony3.0版本，若为其他版本按需修正相应BUILD.gn文件中引入的OpenHarmony源码路径。

----结束

## 设备接入平台

代码烧录到设备后重启设备（首次使用需重启两次，第一次烧录时可能需要配置内部信息），第二次设备重启之后可以正常对接华为云。

登录平台可以看见数据上报的具体信息，也可以进行命令下发等操作控制设备，具体如下图10-12。

图 2-107 查看上报数据-MQTT



图 2-108 在线调试-模拟命令下发

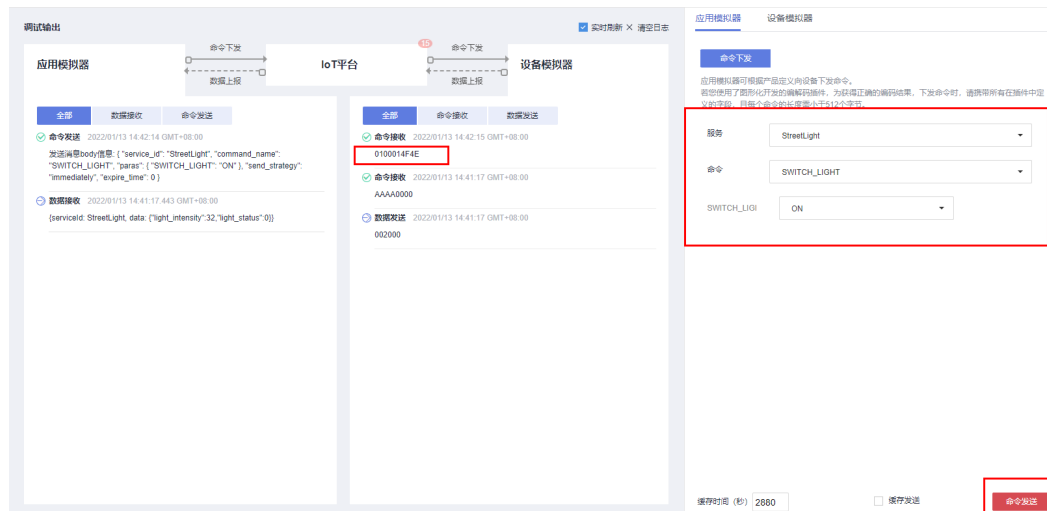


图 2-109 日志信息

```
clients <1> :
 mac_idx mac addr state lease tries rto
 0 201131081a50 10 0 1 4
[DEBUG] [3690] [hub_step] hub_step:enter
[DEBUG] [3690] [dmp_connect] oc_mqtt_connect:server: port:
[DEBUG] [3700] [dmp_connect] oc_mqtt_connect:client_id:
[DEBUG] [3710] [dmp_connect] oc_mqtt_connect:user:
passwd:
[DEBUG] [3720] [dtls_ssl_new] setting up the SSL structure
[DEBUG] [3730] [dtls_ssl_new] set SSL structure succeed
[DEBUG] [3730] [dtls_shakehand] connecting to server
[DEBUG] [3930] [dtls_shakehand] performing the SSL/TLS handshake
[DEBUG] [4370] [dtls_shakehand] handshake succeed
[DEBUG] [4470] [dmp_connect] oc_mqtt_connect:rcode:0 :SUCCESS
[DEBUG] [4470] [dmp_subscribe] oc_mqtt_subscribe:start
[DEBUG] [4480] [dmp_subscribe] oc_mqtt_default_subscribe:topic:$oc/devices/6399432cf5ceb911701ef63_anomaly_detection00lbzm/sys/commands/#
[DEBUG] [5590] [dmp_subscribe] oc_mqtt_default_subscribe:rcode:0:success
[DEBUG] [5590] [dmp_subscribe] oc_mqtt_subscribe:topic:$oc/devices/6399432cf5ceb911701ef63_anomaly_detection00lbzm/sys/messages/down
[DEBUG] [6680] [dmp_subscribe] oc_mqtt_subscribe:rcode:0:success
[DEBUG] [6680] [dmp_subscribe] oc_mqtt_subscribe:topic:$oc/devices/6399432cf5ceb911701ef63_anomaly_detection00lbzm/sys/properties/set/#
[DEBUG] [7810] [dmp_subscribe] oc_mqtt_subscribe:rcode:0:success
[DEBUG] [7810] [dmp_subscribe] oc_mqtt_subscribe:topic:$oc/devices/6399432cf5ceb911701ef63_anomaly_detection00lbzm/sys/properties/get/#
[DEBUG] [8940] [dmp_subscribe] oc_mqtt_subscribe:rcode:0:success
[DEBUG] [8940] [dmp_subscribe] oc_mqtt_subscribe:topic:$oc/devices/6399432cf5ceb911701ef63_anomaly_detection00lbzm/sys/shadow/get/response/#
[DEBUG] [10070] [dmp_subscribe] oc_mqtt_subscribe:rcode:0:success
[DEBUG] [10070] [dmp_subscribe] oc_mqtt_subscribe:topic:$oc/devices/6399432cf5ceb911701ef63_anomaly_detection00lbzm/sys/events/down
[DEBUG] [11090] [dmp_subscribe] oc_mqtt_subscribe:rcode:0:success
[DEBUG] [11090] [hub_step] hub_step:ok exit
ret = 0
oc_mqtt_profile_connect succeed!
```

## 2.9 基于 Jmeter 工具进行 MQTT 性能测试

### 场景说明

随着物联网技术的发展，全球物联网设备连接规模与日俱增。大量设备接入和设备管理对网络带宽、通信协议以及平台服务架构都带来了很大挑战。IoT架构选型时，对平台进行性能测试显得尤为重要。本文介绍如何使用JMeter工具对平台的MQTT协议接入能力进行性能压测。

本文描述的测试方案如下：

测试场景：

- 模拟1万个设备同时在线，验证平台长连接的稳定性。
- 模拟设备每秒发起100次消息上报，验证平台消息处理能力。

测试环境：

- 压测目标平台：华为云设备接入服务SU2（1万设备在线、100TPS消息上下行）。
- 压测执行机：Jmeter执行机1台，规格如下：

表 2-11 压测执行机

| 实例类型    | 规格名称        | vCPUs  | 内存   |
|---------|-------------|--------|------|
| 通用计算型s6 | s6.xlarge.2 | 4vCPUs | 8GiB |

### 说明

单台Jmeter执行机最多能模拟5万设备在线，如果需要模拟更高规格的在线设备数可以使用[华为云的CPTS服务](#)，部署多台Jmeter执行机，以满足规格测试的要求。

## 前提条件

- 已注册华为账号并开通华为云。未注册可单击[注册页面](#)完成注册。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，选择"购买实例"，购买“SU2（1万设备在线、100TPS消息上下行）”。

## 前期准备

- Jmeter执行机安装Java运行环境。访问[Java官网](#)，下载并参照说明安装Java运行环境。
- [下载](#)并安装JMeter (JMeter 应使用5.1.1及以上版本)。
- [下载](#)mqtt-jmeter 插件包，并将插件放置于JMeter安装目录的“lib/ext”目录下。

## 业务流程

使用Jmeter工具对平台进行MQTT协议进行性能压测的具体流程如下：

- 步骤1 创建产品。**创建一个MQTT协议的产品。
- 步骤2 批量注册设备。**采用批量导入的方式注册1万个用于性能测试的设备。
- 步骤3 导入测试计划。**导入已经创建的用于IoT性能测试的测试计划。
- 步骤4 发起压测。**根据业务规格发起对平台的性能压测。
- 步骤5 查看压测结果。**在IoT平台上根据监控指标检查压测结果是否符合预期。

----结束

## 创建产品

- 步骤1** 登录[管理控制台](#)，单击左侧导航栏“产品”，单击页面左侧的“创建产品”。
- 步骤2** 根据页面提示填写参数，然后单击“确定”。

表 2-12 参数说明

| 基本信息   |                                                                                      |
|--------|--------------------------------------------------------------------------------------|
| 所属资源空间 | 平台自动将新创建的产品归属在默认资源空间下。如需归属在其他资源空间下，下拉选择所属的资源空间。如无对应的资源空间，请先创建 <a href="#">资源空间</a> 。 |
| 产品名称   | 自定义。支持字母、数字、下划线（_）、连字符（-）的字符组合。                                                      |
| 协议类型   | 选择MQTT。                                                                              |
| 数据格式   | 选择JSON。                                                                              |
| 所属行业   | 请根据产品模型的所属行业填写。                                                                      |
| 设备类型   | 使用平台预置的产品模型，会自动关联设备类型，不需要再输入设备类型。                                                    |
| 高级配置   |                                                                                      |



|      |                                                                                |
|------|--------------------------------------------------------------------------------|
| 产品ID | 定制ProductID，用于唯一标识一个产品。如果携带此参数，平台将产品ID设置为该参数值；如果不携带此参数，产品ID在物联网平台创建产品后由平台分配获得。 |
|------|--------------------------------------------------------------------------------|

----结束

## 批量注册设备

- 步骤1** 访问[设备接入服务](#)，单击“管理控制台”进入设备接入控制台。
- 步骤2** 在左侧导航栏选择“设备 > 所有设备”，进入“批量注册”页签，单击“批量注册”。
- 步骤3** 下载并填写“批量注册设备文件模板”，可以参照如下表进行参数填写。[样例参考](#)。

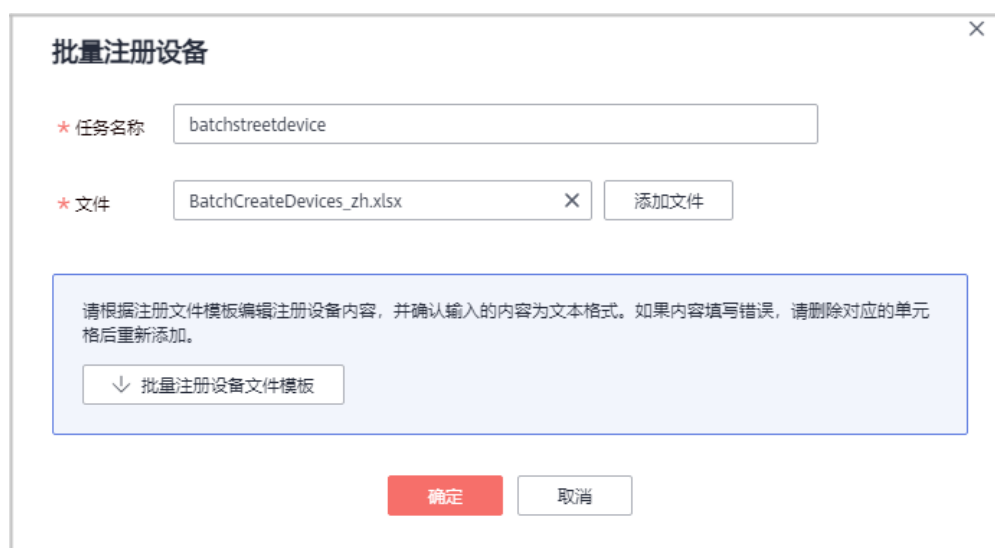
表 2-13 参数说明

| 参数名         | 参数说明                                      |
|-------------|-------------------------------------------|
| node_id     | 设备标识，性能测试时可以按递增的方式进行填写，如10001，10002，10003 |
| product_id  | 产品ID， <a href="#">创建产品</a> 时生成的产品ID。      |
| app_id      | 资源空间，获取方式参考获取 <a href="#">资源空间</a>        |
| device_name | 设备名称，可以同node_id一致。                        |
| secret      | 设备密钥，性能测试时可以固定写死一个密钥。                     |

- 步骤4** 在“批量注册设备窗口”单击“添加文件”上传写好的批量注册文件模板，单击“确定”创建任务。

**步骤5**

图 2-110 设备-批量注册设备



**步骤6** 批量注册执行成功后，保存好设备ID和密钥。

----结束

## 导入测试计划

**步骤1** 下载用于JMeter测试的测试计划。

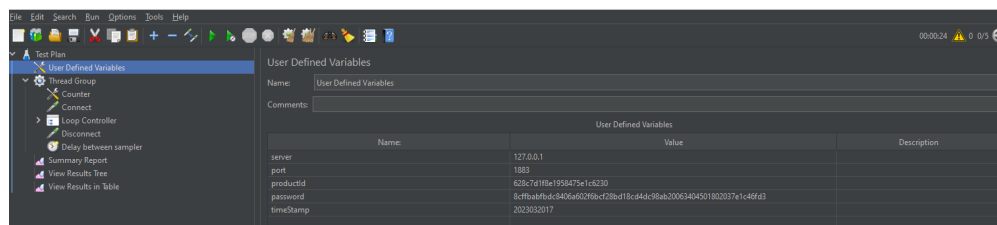
**步骤2** 打开JMeter，选择“Open”导入下载的测试计划。

**步骤3** 在 JMeter左侧文件目录中单击“User Defined Variables”，在“User Defined Variables”对话框中配置以下信息。

表 2-14 参数说明

| 参数名       | 参数说明                                                                                                                   |
|-----------|------------------------------------------------------------------------------------------------------------------------|
| server    | MQTT协议服务端地址。获取方式参考 <a href="#">获取接入地址</a> 。                                                                            |
| port      | MQTT协议端口，填写为8883。                                                                                                      |
| productId | 产品ID， <a href="#">创建产品</a> 时生成的产品ID。                                                                                   |
| password  | MQTT连接密码。为使用“HMACSHA256”算法以时间戳为密钥对secret进行加密后的值。secret为 <a href="#">批量注册设备</a> 时填写的密钥。可以通过 <a href="#">工具</a> 获取加密后的值。 |
| timeStamp | 为对password进行加密的时间戳。时间格式为YYYYMMDDHH。                                                                                    |

图 2-111 参考样例

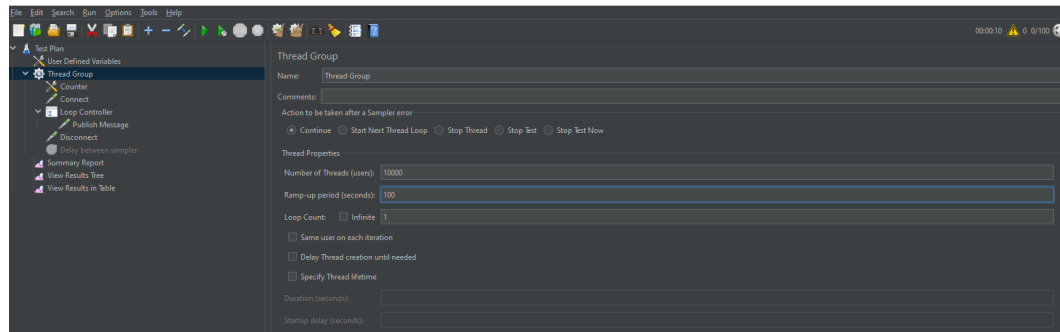


----结束

## 发起压测

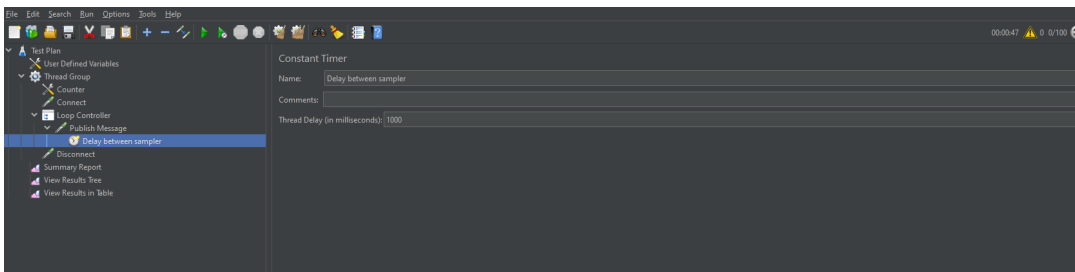
**步骤1** 在 JMeter 左侧文件目录中单击“Thread Group”，配置“Number of Threads”为10000（一个线程对应一个设备在线，配置为10000表示设备与平台间保持1万个设备在线）。

图 2-112 配置设备



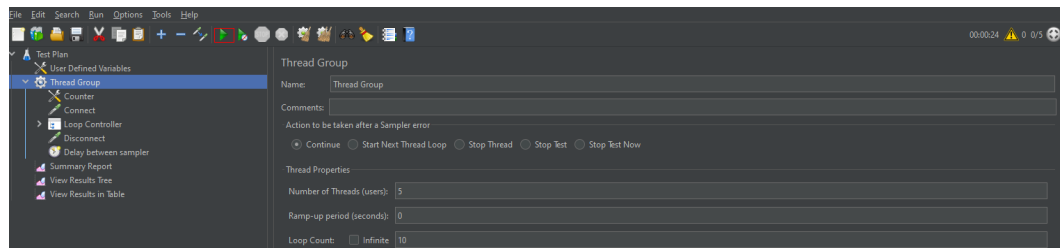
**步骤2** 在 JMeter 左侧文件目录“Thread Group”目录下单击“Loop Controller”->“Publish Message”->“Delay between sampler”，配置“Thread Delay(in milliseconds)”为100000（每设备每100秒发布一次消息）。

图 2-113 配置设备



**步骤3** 在Jmeter工具栏图表中单击“开始”按钮，启动性能测试。

图 2-114 性能测试



**步骤4** 在 JMeter 左侧文件目录中单击“Summary Report”，可以看到“Connect”、“Publish Message”的吞吐量。可以通过调整“Number of Threads”和“Thread Delay(in milliseconds)”来调整Jmeter发起的吞吐量。

图 2-115 性能测试

| Label           | # Samples | Average | Min | Max | Std. Dev. | Error % | Throughput | Received KB/sec | Sent KB/sec | Avg. Bytes |
|-----------------|-----------|---------|-----|-----|-----------|---------|------------|-----------------|-------------|------------|
| Connect         | 50        | 11      | 1   | 30  | 9.17      | 0.00%   | 67.3/sec   | 0.72            | 0.00        | 11.0       |
| Publish Message | 500       | 0       | 0   | 2   | 0.14      | 0.00%   | 645.2/sec  | 12.60           | 35.91       | 20.0       |
| Disconnect      | 50        | 1       | 0   | 5   | 1.17      | 0.00%   | 68.9/sec   | 0.74            | 0.00        | 11.0       |
| TOTAL           | 600       | 1       | 0   | 30  | 4.09      | 0.00%   | 737.1/sec  | 13.32           | 34.19       | 18.5       |

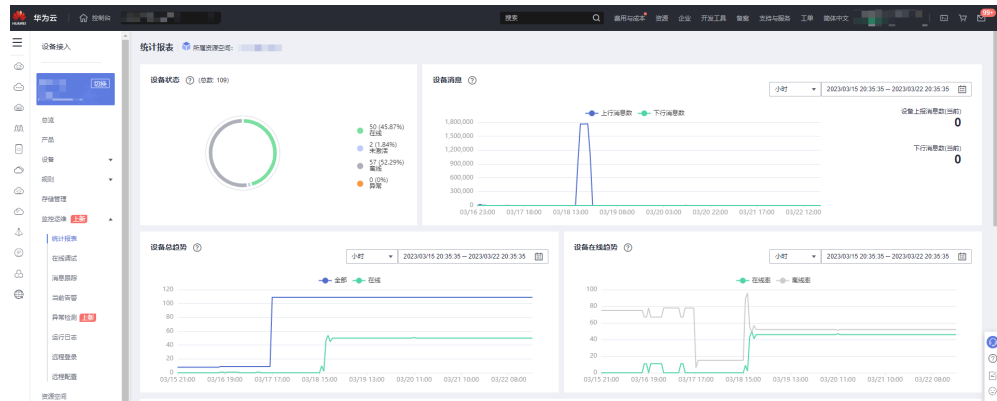
步骤5 JMeter测试计划调试正常后，可以将测试计划导入到[华为云的CodeArts PerfTest服务](#)，进行分布式部署以满足更高规格的性能测试。

----结束

## 查看压测结果

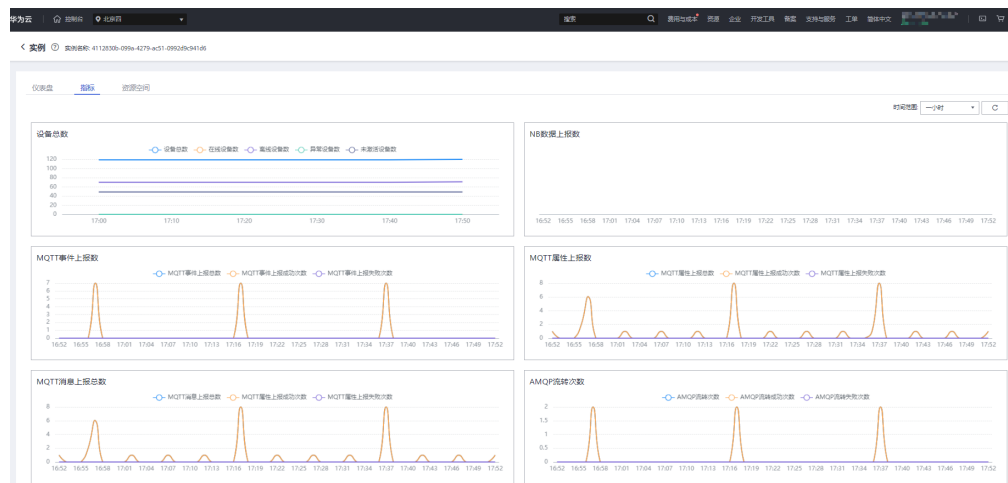
1. 登录[管理控制台](#)，在左侧导航栏“监控运维”->“统计报表”，查看平台的统计指标。

图 2-116 统计报表-查看统计报表



2. 更多报表，可登录[应用运维管理控制台](#)，选择监控 >云服务监控 >物联网>IoTDA查看。

图 2-117 查看指标



# 3 设备管理

## 3.1 预置设备影子自动调节空调温度

### 场景说明

通过恒温控制系统，不论空调是否开机，都可以调整空调默认温度，待空调上电开机后，自动按默认温度调节。空调接入到物联网平台后，用户可以在应用侧或者设备接入控制台设置设备影子，将预置的温度通过设备影子下达属性修改给空调。空调收到修改属性的要求后，自动调节温度。

### 恒温空调产品开发

**步骤1** 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。

**步骤2** 选择左侧导航栏的“产品”，单击右上角下拉框，选择新建产品所属的资源空间。

**步骤3** 单击左侧的“创建产品”，创建恒温空调产品，填写参数后，单击“确定”。

| 基本信息 |                   |
|------|-------------------|
| 产品名称 | 自定义，如aircondition |
| 协议类型 | 选择“MQTT”          |
| 数据格式 | 选择“JSON”          |
| 所属行业 | 自定义               |
| 设备类型 |                   |

**步骤4** 产品创建成功后，单击对应的产品进入产品详情页面。

**步骤5** 在模型定义页面，单击“自定义模型”，参考下表，完成产品模型配置。

| 服务数据 |
|------|
|------|

|    |                                                        |
|----|--------------------------------------------------------|
| 服务 | 服务ID: temperature<br>服务类型: 建议和服务ID保持一致                 |
| 属性 | 属性名称: temperature<br>数据类型: int<br>访问权限: 可读、可写<br>长度: 1 |

**步骤6** 进入“设备 > 设备注册”页面，单击“注册设备”，参考下表填写参数。

**图 3-1** 单设备注册-MQTT

**单设备注册**
×

\* 所属资源空间 ?

\* 所属产品

\* 设备标识码 ?

设备ID ?

设备名称

设备描述  0/2,048

设备认证类型 ? 密钥 X.509证书

密钥  👁

确认密钥  👁

确定
取消

| 参数名称   | 说明                                                               |
|--------|------------------------------------------------------------------|
| 所属产品   | 选择在步骤 <b>步骤3</b> 中创建的产品。                                         |
| 设备标识码  | 即node_id，填写为设备的IMEI、MAC地址或Serial No；若没有真实设备，填写自定义字符串，由英文字母和数字组成。 |
| 设备名称   | 自定义。                                                             |
| 设备认证类型 | 选择“密钥”。                                                          |
| 密钥     | 设备密钥，可自定义，不填写物联网平台会自动生成。                                         |

----结束

## 配置设备影子

用户可以选择在应用侧调用[配置设备影子预期数据接口](#)设置设备影子，也可以在设备接入控制台设置设备影子。此处以在设备接入控制台设置设备影子为例。

- 步骤1** 登录[控制台](#)，选择左侧导航栏的“设备”，单击步骤[步骤6](#)注册的设备进入到恒温空调设备的详情页面。
- 步骤2** 选择“设备影子”页签，单击“属性配置”。
- 步骤3** 在弹出窗口中输入服务属性对应的期望值。此处设置“temperature”的属性值为25。

图 3-2 配置设备影子



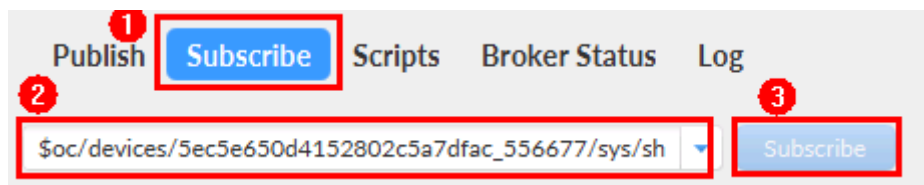
----结束

## 验证操作

方法一：

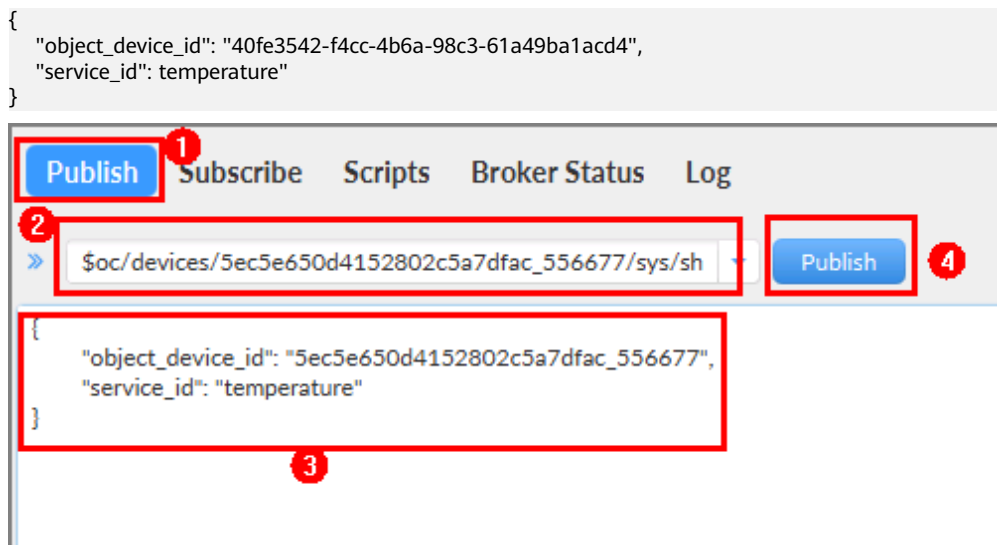
您可以使用MQTT.fx模拟设备验证。

1. 使用MQTT.fx模拟恒温空调，并连接到物联网平台。操作方法请参考[在线开发MQTT协议的模拟智慧路灯](#)。
2. 选择“Subscribe”页签，输入订阅设备影子的topic=\$oc/devices/{device\_id}/sys/shadow/get/response/#，其中{device\_id}与步骤[步骤6](#)的设备ID保持一致，单击“Subscribe”按钮。

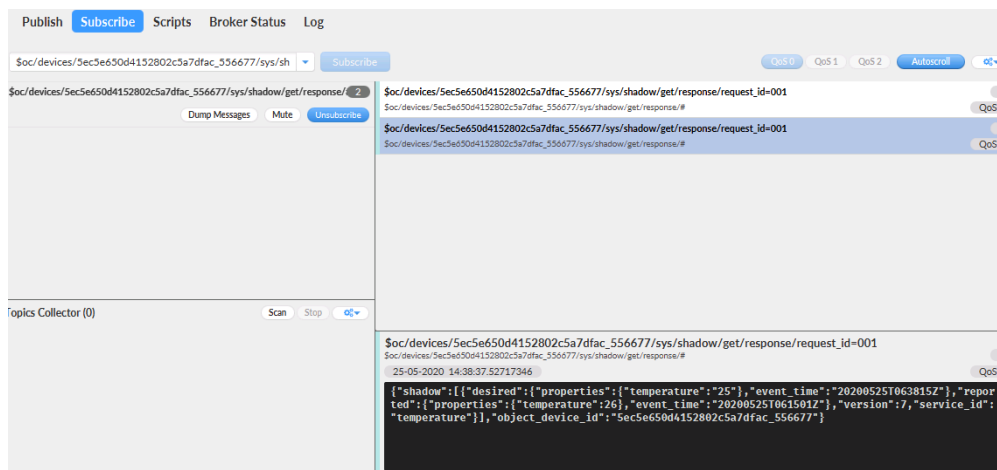


3. 选择“Publish”页签，输入请求设备影子的Topic=\$oc/devices/{device\_id}/sys/shadow/get/request\_id={request\_id}。
4. 输入一条获取设备影子的请求，单击“Publish”按钮。

示例：



5. 选择“Subscribe”页签，看到平台下发的设备影子数据。



方法二：

您可以使用配置设备接入服务时注册的真实设备接入平台，设备会收到平台下发的设备影子，修改空调的预设温度值。

## 3.2 使用自定义 Topic 管理室内空调

### 场景说明

自定义Topic功能用于MQTT协议设备接入物联网平台的场景，支持**设备消息上报**、**平台消息下发**的Topic进行自定义。根据Topic不同，应用侧可实现不同的业务逻辑处理，也适用于应用侧和设备侧之间，设备无法按照产品模型中定义的属性上报或命令下发的场景。

本示例中，应用服务器收到设备上报Topic为室内温度的数据，判断是否开启或关闭室内空调。

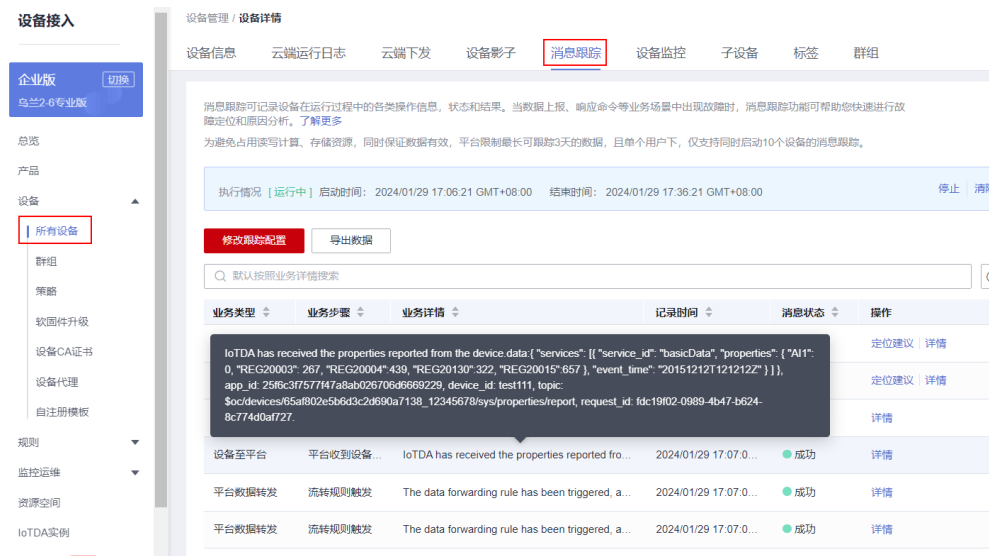
### 前提条件

- 已经注册华为账号并开通华为云，具体可参考[华为账号注册](#)。





图 3-3 查看消息跟踪

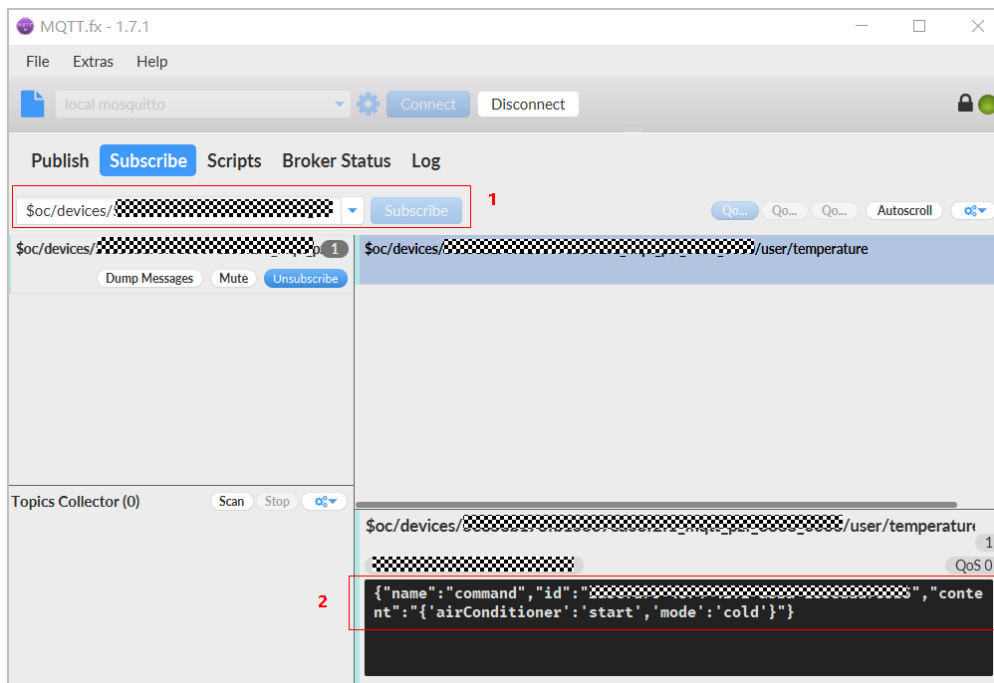


- 应用侧通过数据转发的方式，获取MQTT设备上上报的自定义Topic消息。更多数据转发方式，请参考[数据转发](#)。您也可以参考最佳实践案例[数据转发到OBS长期存储](#)。

## MQTT 设备消息下发

本示例通过使用postman，下发开启室内空调指令。

- 使用MQTT.fx模拟器，订阅自定义Topic。





## 3.3 MQTT 协议设备 OTA 固件升级

### 场景说明

MQTT ( Message Queuing Telemetry Transport ) 是一个基于客户端-服务器的消息发布/订阅传输协议，主要应用于计算能力有限，且工作在低带宽、不可靠的网络的远程传感器和控制设备，适合长连接的场景，如智能路灯等。固件升级功能是网络设备所必须支持的基础服务。在设备固件出现安全漏洞、软件bug维护、自身功能优化、以及设备性能改良等需更新固件时，在线升级功能就尤为重要。本文档通过MQTT.fx模拟固件设备，使用华为云平台对该固件进行设备升级。

#### 说明

软件升级流程和固件升级流程一样，只是在设备上报版本号时对应参数字段不一样。固件升级上报版本号对应参数字段为fw\_version，软件升级上报版本号对应参数字段为sw\_version,请参考[设备上报版本号](#)

### 整体流程

图 3-4 升级实践流程

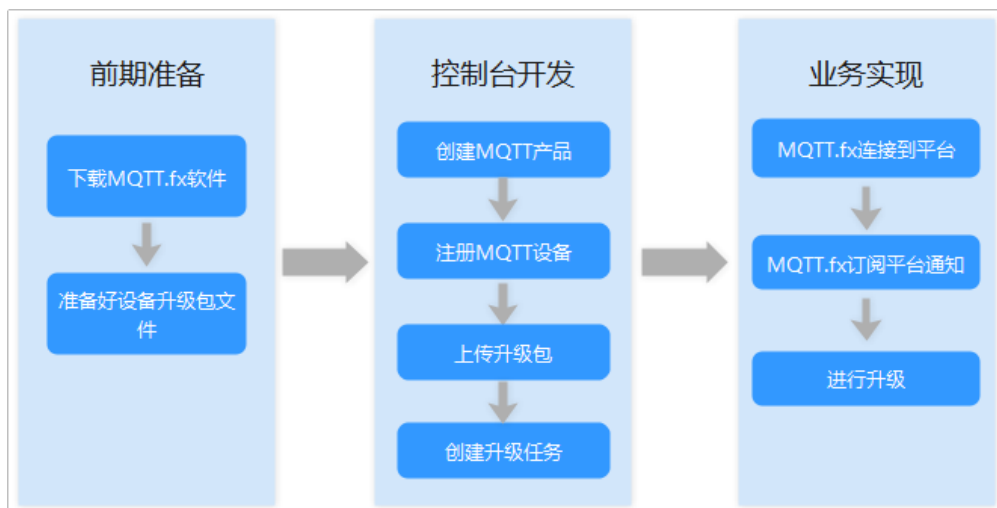
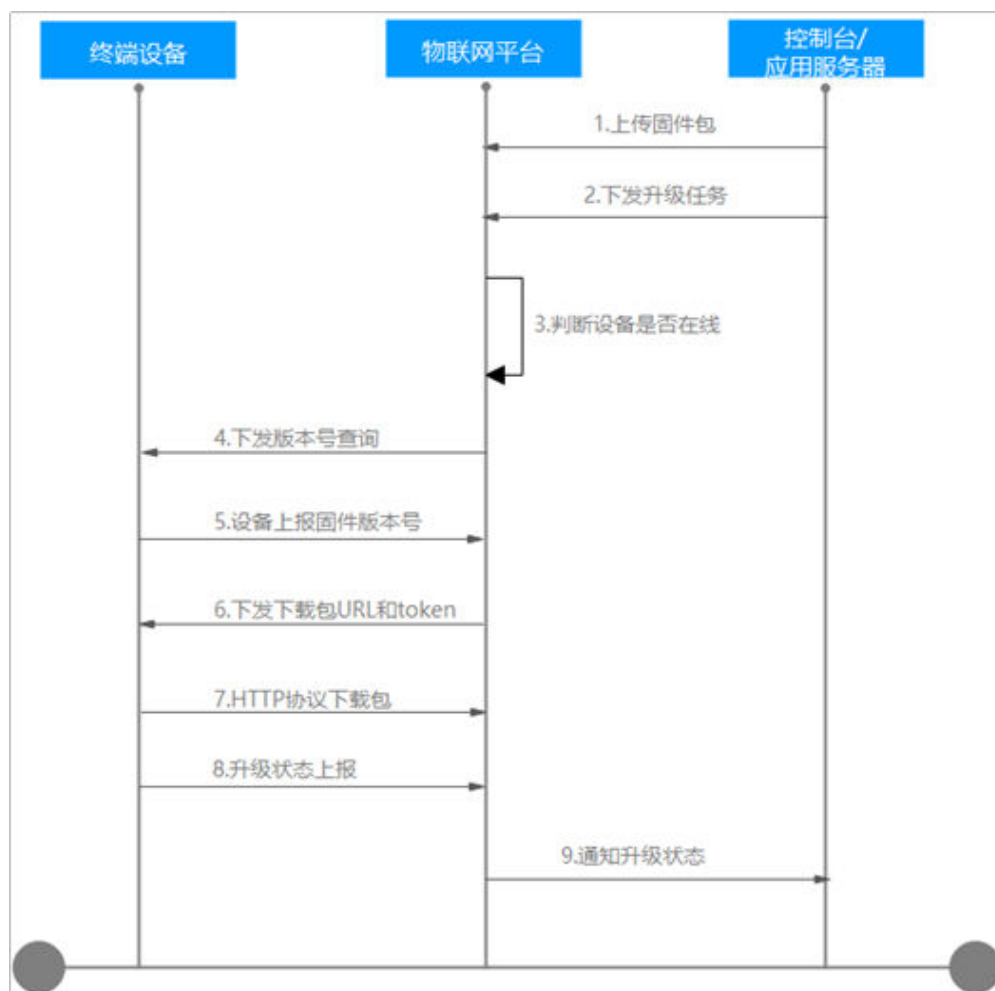


图 3-5 MQTT 设备升级流程



## 开发环境

- 软件：华为云（已开通[设备接入服务](#)）、win7及以上版本的64位操作系统（本文以windows 10 64-bit系统调试）、[MQTT.fx](#)模拟器。

## 前提条件

- 已注册华为云官方账号。未注册可单击[注册页面](#)完成注册。
- 已完成实名制认证。未完成可在华为云上单击完成认证，否则无法使用设备接入功能。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，单击“立即使用”后开通该服务。

## 前期准备

- 下载并安装MQTT.fx(应使用1.7.1及以上版本)。
- 升级包文件准备。固件升级包及其版本号需向模组厂家获取。本次演示使用临时文件代替。
- [创建产品](#)，创建MQTT协议产品（如果有MQTT协议产品可直接使用）。
- [注册单个设备](#)。

- **设备群组**，将注册的设备加入一个设备群组(创建升级任务时会用到)。

## 上传升级包

- 步骤1** 访问**设备接入服务**，单击“立即使用”进入设备接入控制台。
- 步骤2** 在左侧导航栏选择“设备 > 软固件升级”。
- 步骤3** 选择“资源包管理”页签，选择“固件列表”。
- 步骤4** 单击“上传固件”，在弹出的页面中有两种上传方式，OBS文件上传和本地文件上传。

图 3-6 上传升级包-OBS 文件

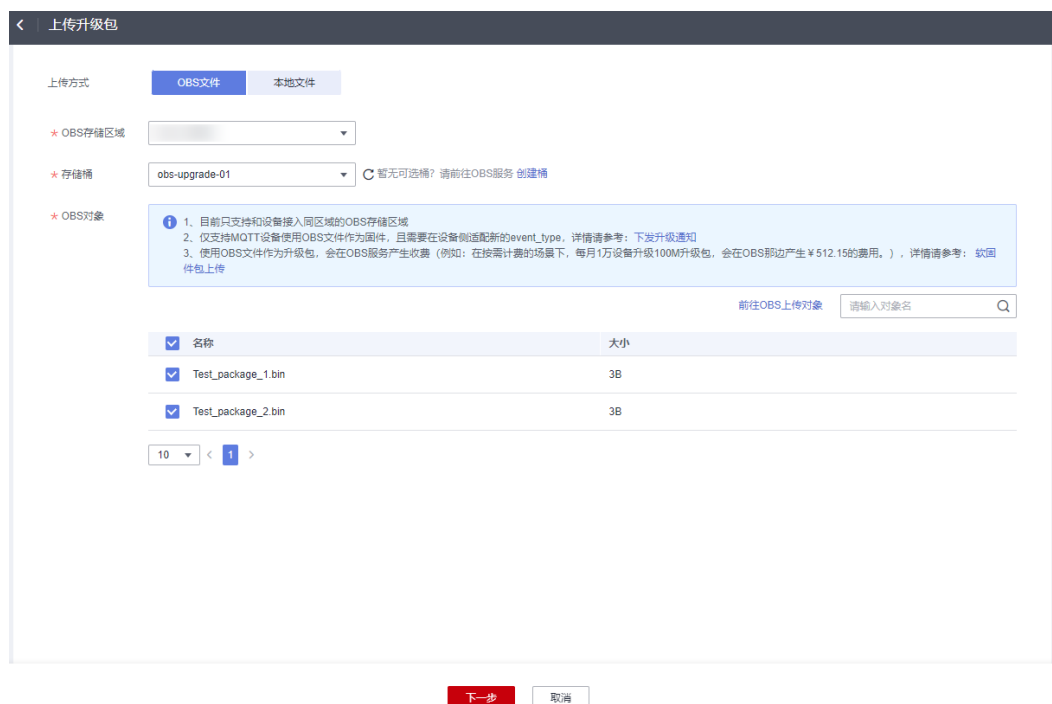


图 3-7 上传升级包-本地文件



- 步骤5** 上传升级包后，填写升级包参数信息，单击“确定”完成升级包上传。

图 3-8 上传升级包-OBS 文件参数

上传方式: OBS文件 | 本地文件

升级包列表

Test\_package\_1\_bin [删除]

\* 固件版本: V1.0 \* 选择产品: Test\_1

支持设备源版本: 版本号可使用英文分号分隔 [添加] 描述: [输入框]

V0.1 X 0/1,024

推送给设备的自定义信息: [输入框] 0/4,096

Test\_package\_2\_bin [删除]

\* 固件版本: V2.0 \* 选择产品: Test\_1

支持设备源版本: 版本号可使用英文分号分隔 [添加] 描述: [输入框]

V0.2 X 0/1,024

推送给设备的自定义信息: [输入框] 0/4,096

上一步 确定 取消

图 3-9 上传升级包-本地文件参数

上传方式: OBS文件 | 本地文件

\* 升级包上传 注意: 多个文件上传时, 文件名不能有重复, 且文件名不能包含特殊字符/或\*

拖拽本地文件至此处, 或 添加文件  
单次最多支持10个文件同时上传, 单文件不超过60MB

升级包列表

Test\_package\_3\_bin [删除]

\* 固件版本: V3.0 \* 选择产品: Test\_1

支持设备源版本: 版本号可使用英文分号分隔 [添加] 描述: [输入框]

V0.3 X 0/1,024

推送给设备的自定义信息: [输入框] 0/4,096

确定 取消

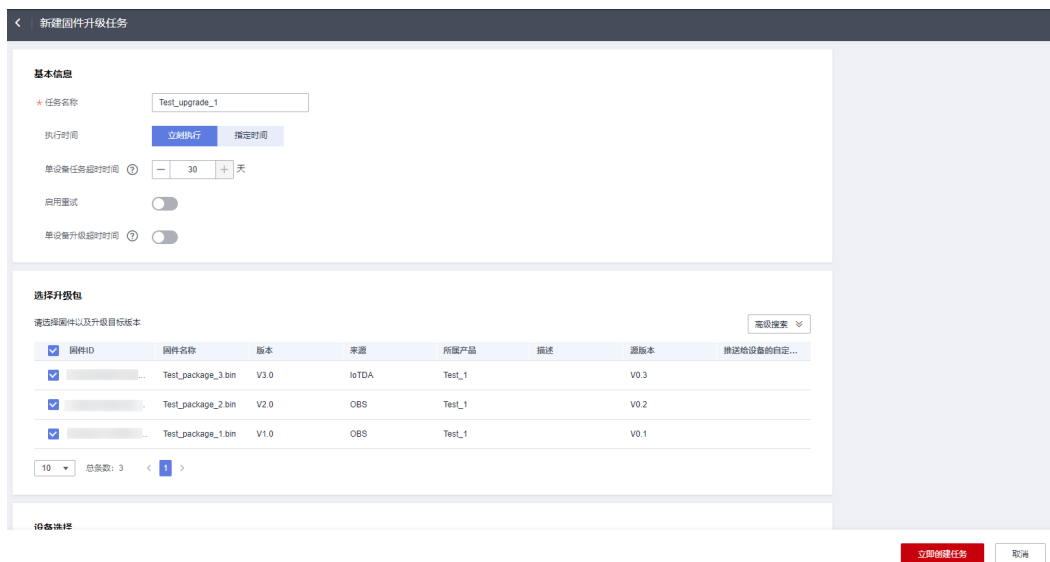
步骤6 升级包上传完成后会在"固件列表"中显示出来。

图 3-10 软固件升级-固件列表



**步骤7** 选择“升级任务”，在“固件升级”页签，单击“新建任务”按钮，填写参数信息，选择升级包，选择需要升级的设备，单击“立刻创建任务”。

图 3-11 新建固件升级任务-Test\_upgrade\_1



**说明**

创建升级任务时，可以选择多个升级包（最多10个），选择的升级包必须满足支持的源版本没有重复。如果某个升级包没有指定支持源版本，则默认该升级包支持所有源版本设备升级。

**步骤8** 任务创建成功后可以在任务列表中查询到该任务，单击详情可以查看当前任务详情。在任务详情页面可以批量停止（一次最多100个设备）或单个停止正在执行中的设备升级，也可以批量重试（一次最多100个设备）或单个重试失败的设备升级。单击“全部重试”可以重试该批量升级任务中所有失败的设备升级。



图 3-12 任务详情-Test\_upgrade\_1



----结束

## 业务实现

**步骤1** 使用MQTT.fx模拟设备接入平台，详情请参考连接鉴权。

**步骤2** 用MQTT.fx订阅平台下行信息Topic，MQTT.fx会收到平台服务端下发的查询版本命令。

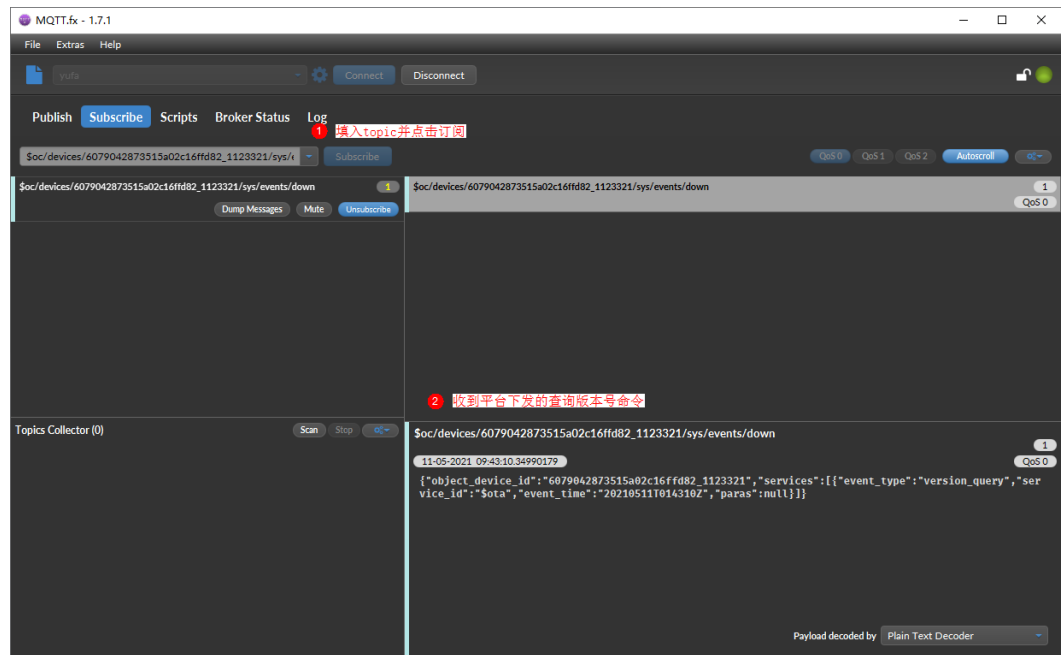
### Topic

下行: \$oc/devices/{device\_id}/sys/events/down

### 参数说明

请参考[平台下发获取版本信息通知](#)

图 3-13 订阅通知



**步骤3** 通过MQTT.fx上报软固件版本信息。

### Topic

上行: \$oc/devices/{device\_id}/sys/events/up

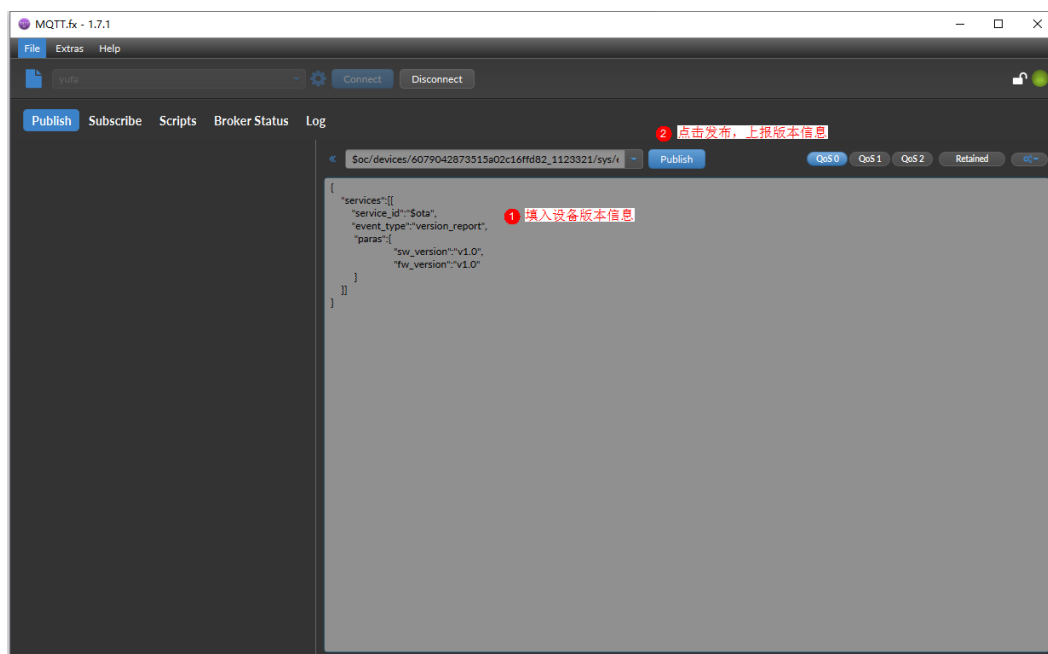
### 参数说明

请参考[设备上报软固件版本](#)

### 示例

```
Topic: $oc/devices/{device_id}/sys/events/up
数据格式: {
 "object_device_id": "{object_device_id}",
 "services": [{
 "service_id": "$ota",
 "event_type": "version_report",
 "event_time": "20151212T121212Z",
 "paras": {
 "sw_version": "v1.0",
 "fw_version": "v1.0"
 }
]
}
```

图 3-14 上报版本号



**步骤4** 上报完版本号后，模拟器会接收到平台下发的升级通知，通知信息如下：

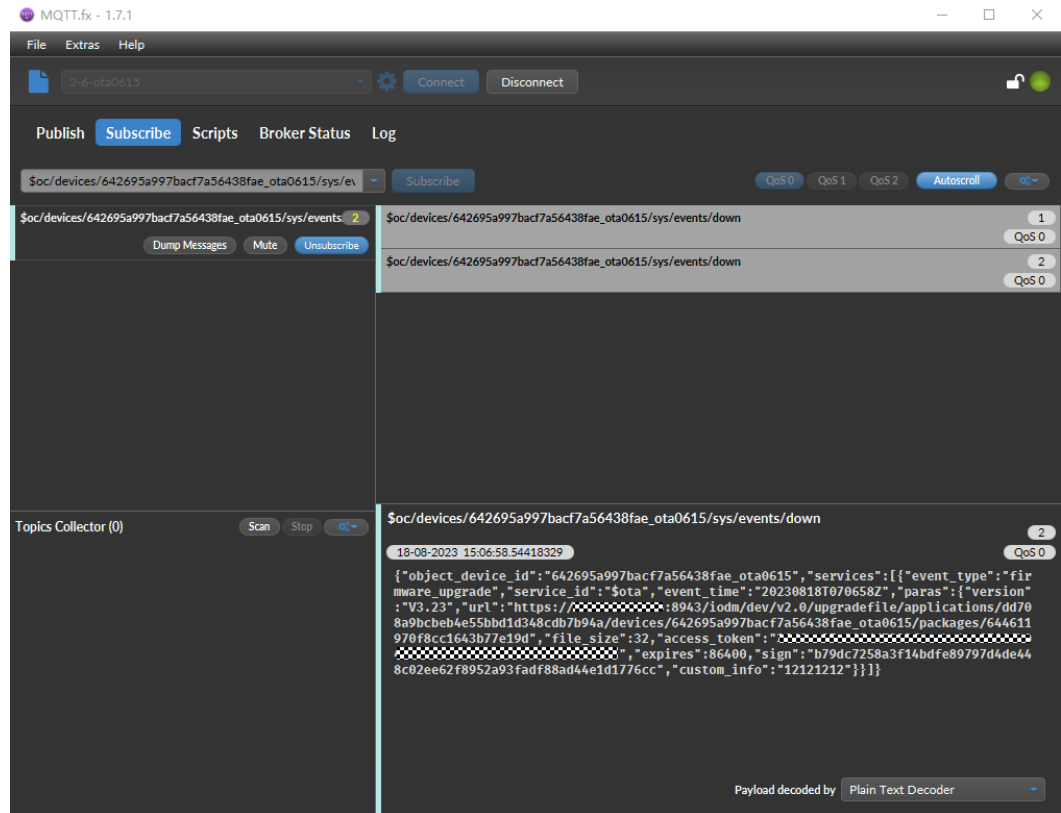
#### Topic

上行: \$oc/devices/{device\_id}/sys/events/down

#### 参数说明

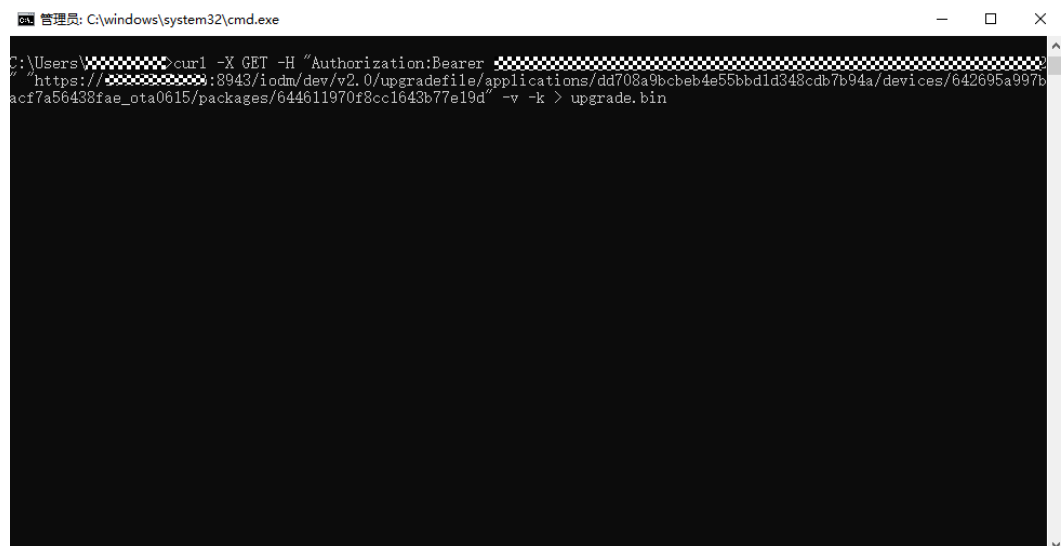
请参考[平台下发升级通知](#)

图 3-15 获取升级通知



**步骤5** 设备收到升级通知后，可通过Http请求下载升级包。  
本次演示使用Curl命令代替。

图 3-16 下载升级包



示例

```
curl -X GET -H "Authorization:Bearer *****" "https://100.93.28.202:8943/iodm/dev/v2.0/upgradefile/applications/ddecccc223574aee9466fe8f7e16a205/devices/6079042873515a02c16ffd82_123456789/packages/4f201f38c281ca5d40794a3f" -v -k
```

**⚠ 注意**

- 应在http附加请求消息头（header）字段附加Authorization信息，其值为“Bearer {access\_token}”，其中{access\_token}为收到的升级通知中的access\_token。Bearer和{access\_token}之间有一个空格。
- 如果event\_type为firmware\_upgrade\_v2、software\_upgrade\_v2，则在请求下载软件包时不需要携带请求头。请求示例如下：

```
GET https://*****.obs.cn-north-4.myhuaweicloud.com:443/test.bin?
AccessKeyId=DX5G7W*****
```

**步骤6 设备上报升级状态**

**Topic**

上行: \$oc/devices/{device\_id}/sys/events/up

**参数说明**

请参考[设备上报升级状态](#)

**示例**

Topic: \$oc/devices/{device\_id}/sys/events/up

数据格式:

```
{
 "object_device_id": "{object_device_id}",
 "services": [{
 "service_id": "$ota",
 "event_type": "upgrade_progress_report",
 "event_time": "20151212T121212Z",
 "paras": {
 "result_code": 0,
 "progress": 50,
 "version": "V1.0",
 "description": "upgrade processing"
 }
]
}
```

如下图所示上报升级进度为50%，可以在平台查询到升级进度为50%。

图 3-17 上报升级进度 50%

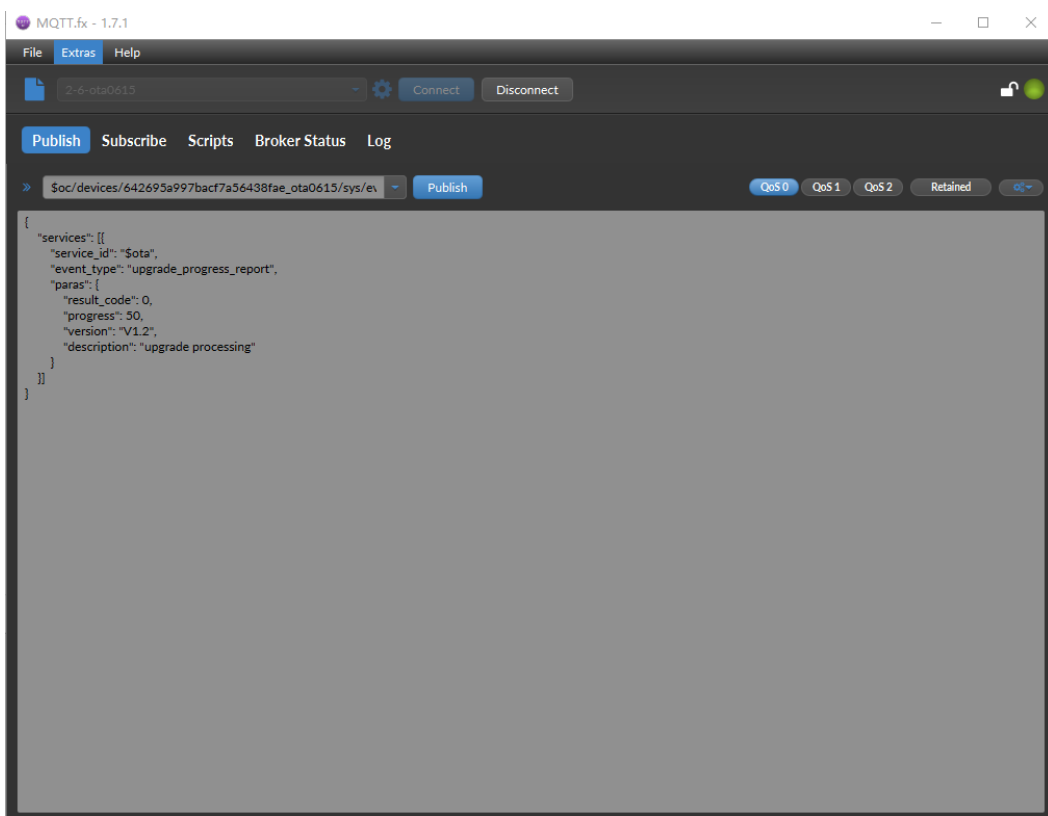


图 3-18 任务详情-Test\_upgrade\_1-升级进度

### 任务详情 ×

---

**基本信息**

|                                                    |                               |                                                    |                    |
|----------------------------------------------------|-------------------------------|----------------------------------------------------|--------------------|
| 任务名称                                               | Test_upgrade_1                | 状态                                                 | 执行中                |
| 执行时间                                               | 立刻执行                          | 任务类型                                               | 固件升级               |
| 开始时间                                               | 2024/01/12 10:35:59 GMT+08:00 | 重试策略                                               | 重试0次   重试间隔 (分钟) 0 |
| 单设备升级超时时间 <span style="font-size: 0.8em;">?</span> | 30分钟                          | 单设备任务超时时间 <span style="font-size: 0.8em;">?</span> | 30天                |

**执行详情** 已选升级包

○  
升级设备总数  
**1**

●  
升级成功数  
**0**

●  
升级失败数  
**0**

●  
其他  
**1**

所有状态 ▼

|                          | 状态                                      | 设备ID               | 升级描述                        | 操作                                      |
|--------------------------|-----------------------------------------|--------------------|-----------------------------|-----------------------------------------|
| <input type="checkbox"/> | <span style="color: blue;">●</span> 执行中 | Test_1_Test_MQTT_1 | downloading progress is 50% | <a href="#">重试</a>   <a href="#">停止</a> |

10
总条数: 1
< 1 >

### 步骤7 完成升级

上报升级进度为100%，且当前版本为升级的目标版本，可以在平台查询到升级完成。

图 3-19 上报升级进度 100%

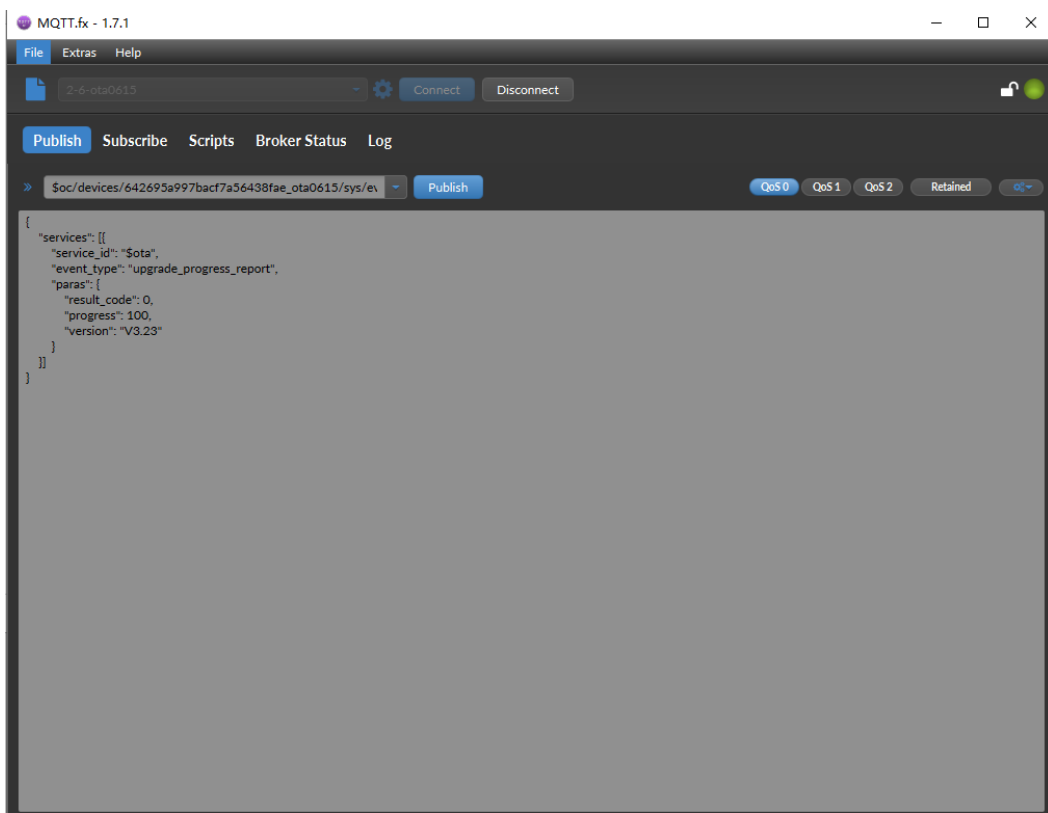


图 3-20 软固件升级-固件升级



----结束

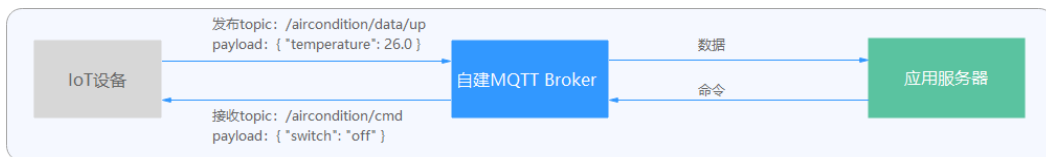
## 3.4 设备自定义 TOPIC 迁移上云

### 场景说明

随着IoT终端设备规模不断增加，企业自建物联网面临服务器扩容，硬件投入和运维成本越来越高等诸多问题。为此，华为云物联网平台提供设备极简迁移上云解决方案，企业设备通过极少的改动，就可以快速将设备迁移到华为云物联网平台。

### 整体方案

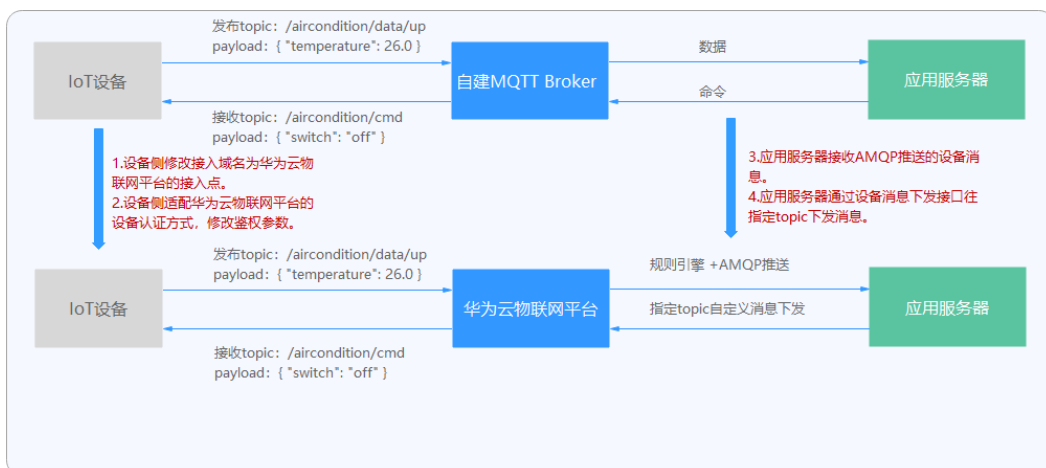
假设企业终端设备接入自建MQTT集群，业务架构如下图所示。



基于MQTT协议的上行数据和下行指令的业务定义如下：

| 业务场景    | 通信Topic               | 报文Payload               |
|---------|-----------------------|-------------------------|
| 设备上报数据  | /aircondition/data/up | { "temperature": 26.0 } |
| 服务端控制指令 | /aircondition/cmd     | { "switch": "off" }     |

为减少企业改造成本，华为云物联网平台提供如下迁移方案，设备侧不用改变原有的topic和payload报文格式，就可以快速迁移设备到华为云物联网平台。



企业设备迁移上云有三个核心变更点：

- 设备侧修改接入域名为华为云物联网平台的接入点。
- 配置规则引擎，把设备数据流转应用到应用服务器、AMQP消费组或华为云第三方云服务产品。
- 应用服务器适配设备消息下发接口往指定topic下发消息。

## 操作步骤

1. 在控制台配置产品、设备和数据流转方案，请参见[云端配置开发](#)。
2. 对设备端进行业务开发，请参见[设备端开发](#)。
3. 对服务端进行业务开发，实现接收设备数据和下发控制指令，请参见[服务端开发](#)。

## 云端配置开发

**步骤1** 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。

**步骤2** 创建产品。



1. 选择左侧导航栏的“产品”。
2. 单击左侧的“创建产品”，创建待迁移设备的产品，填写参数后，单击“确定”。

| 基本信息 |                           |
|------|---------------------------|
| 产品名称 | 自定义，如aircondition         |
| 协议类型 | 选择“MQTT”                  |
| 数据格式 | 选择“JSON”                  |
| 所属行业 | 无                         |
| 设备类型 | 根据实际情况填写，如“aircondition”。 |

### 步骤3 注册设备。

1. 选择左侧导航栏的“设备 > 所有设备 > 设备注册”，单击页面右上角的“注册设备”。

图 3-21 单设备注册-aircondition

### 单设备注册

×

\* 所属资源空间 ?

\* 所属产品  MQTT类型的设备已默认订阅平台预置topic, [查看已订阅topic列表](#)

\* 设备标识码 ?

设备ID ?

设备名称

设备描述  0/2,048

设备认证类型 ? 密钥 X.509证书

密钥  👁

确认密钥  👁

确定
取消

| 参数名称   | 说明                                                               |
|--------|------------------------------------------------------------------|
| 所属产品   | 选择在步骤2中创建的产品。                                                    |
| 设备标识码  | 即node_id，填写为设备的IMEI、MAC地址或Serial No；若没有真实设备，填写自定义字符串，由英文字母和数字组成。 |
| 设备名称   | 自定义。                                                             |
| 设备认证类型 | 根据企业设备现有的认证类型，选择对应的认证方式。本文以“密钥”方式为例。                             |
| 密钥     | 设备密钥。填写待迁移设备的密钥。                                                 |

2. 填写设备参数后，单击“确定”。

#### 步骤4 创建数据转发规则。

1. 选择左侧导航栏的“规则 > 数据转发”，单击页面左侧的“创建规则”。
2. 参考下表填写参数后，单击“创建规则”。

| 参数名  | 参数说明                                                         |
|------|--------------------------------------------------------------|
| 规则名称 | 创建的规则名称。                                                     |
| 规则描述 | 对该规则的描述。                                                     |
| 数据来源 | 选择“设备消息”。设备迁移上云时，设备按原有的topic和payload进行上报，物联网平台默认按设备消息流程进行处理。 |
| 触发事件 | 选择数据来源后，自动匹配触发事件。                                            |
| 资源空间 | 您可以选择单个资源空间或所有资源空间。                                          |

3. 在设置转发目标页面，单击“添加”，在弹出的页面中参考下表配置完参数后，单击“确认”。

| 参数名  | 参数说明                                                                                                                                                                                                                       |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 转发目标 | 选择数据转发目标，本文以“AMQP推送消息队列”为例。                                                                                                                                                                                                |
| 消息队列 | <p>单击“选择”，选择消息队列。</p> <ul style="list-style-type: none"> <li>- 若没有消息队列，请新建消息队列，队列名称自定义且单个租户名下唯一，长度8-128，只能包含大写字母、小写字母、数字和指定特殊字符（如_-:）。</li> <li>- 若需要删除消息队列，单击消息队列右侧的“删除”即可。</li> </ul> <p><b>说明</b><br/>已经订阅的队列不允许删除。</p> |

4. 完成完整的规则定义后，单击“启动规则”，实现数据转发至AMQP消息队列。

---结束

## 设备端开发

完成云端配置后，需要进行设备端业务开发。完整的设备开发流程可参考[设备侧开发](#)。本章节以MQTT.fx为例，介绍在设备迁移场景下，设备侧如何在尽量少改动的情况下，实现设备建立MQTT连接、数据上报、指令接收等功能。

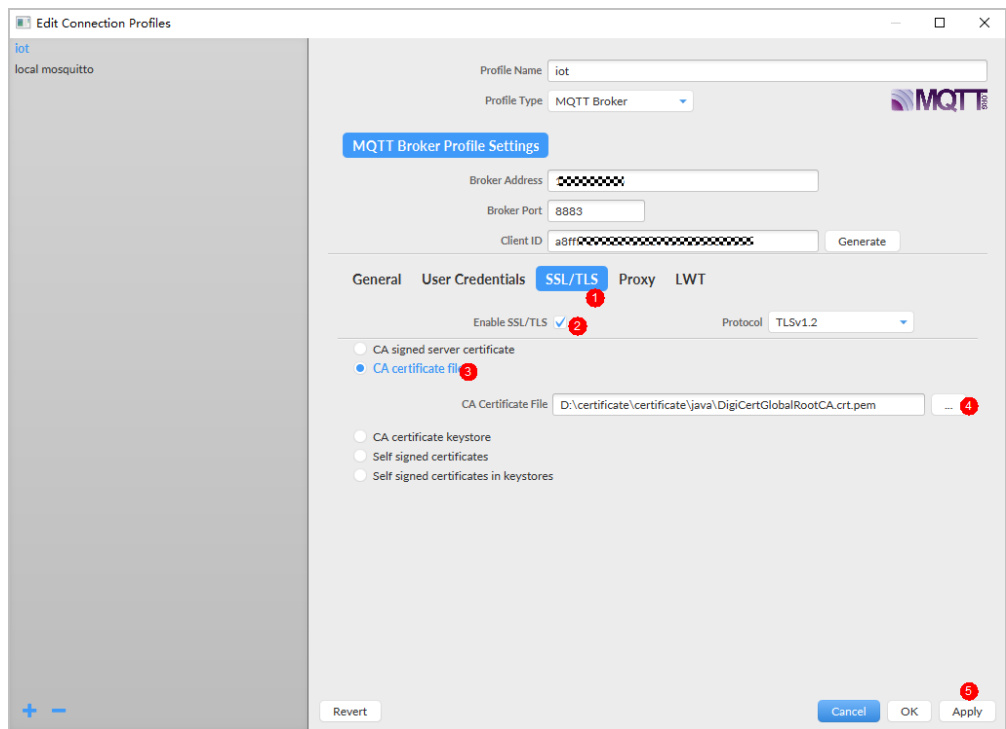
### 步骤1 设备同物联网平台建立MQTT连接。

1. 参考下表配置鉴权参数。

| 参数             | 必选/可选 | 参数描述                                           |
|----------------|-------|------------------------------------------------|
| Broker Address | 必选    | 华为云物联网平台的MQTT协议接入地址，请参考 <a href="#">此处</a> 获取。 |
| Broker Port    | 必选    | 8883。                                          |
| Client ID      | 可选    | 使用设备迁移前的Client ID。                             |

| 参数        | 必选/可选 | 参数描述                                                                                                                               |
|-----------|-------|------------------------------------------------------------------------------------------------------------------------------------|
| User Name | 必选    | 填写步骤3注册设备时生成的设备ID，默认通过控制台生成的设备ID会添加产品ID前缀。在设备迁移场景，设备侧 User Name参数无法修改时可以调用 <a href="#">创建设备</a> 接口，指定设备ID参数值同迁移前的User Name参数值保持一致。 |
| Password  | 必选    | 加密后的设备密钥。Password的值为使用“HMACSHA256”算法以时间戳为密钥，对secret进行加密后的值。<br>secret为注册设备时平台返回的secret。                                            |

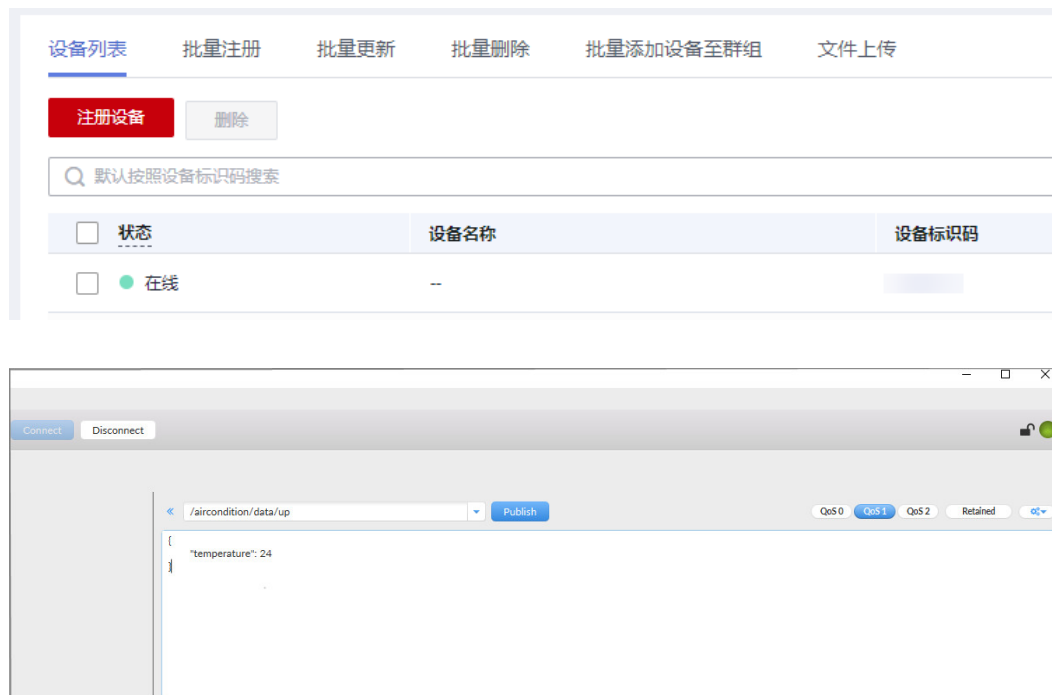
2. 参考下表配置“SSL/TLS”认证参数，然后单击“Apply”。



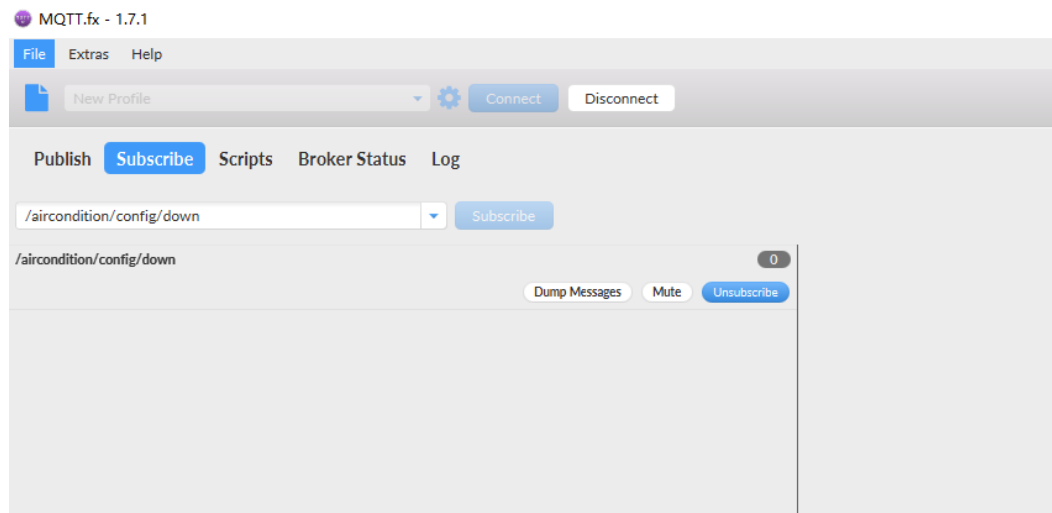
| 参数                  | 必选/可选 | 参数描述                               |
|---------------------|-------|------------------------------------|
| Enable SSL/TLS      | 必选    | 选择“Enable SSL/TLS”                 |
| CA certificate file | 必选    | 上传 <a href="#">证书资源</a> 页面获取的CA证书。 |

**步骤2** 设备同物联网平台建立连接后，设备沿用迁移前的topic和payload格式上报数据。物联网平台针对这类非系统预定义的topic，统一按照“设备消息”的处理流程将设备上报的数据转发给第三方应用或者华为云的其他云服务处理。

图 3-22 设备在线



**步骤3** 根据迁移前的topic进行订阅，接收应用服务器下发的指令。



----结束

## 服务端开发

对服务端进行业务开发，实现接收设备数据和下发控制指令。本文以Java脚本为例，演示接收设备数据和下发控制指令。

- **业务服务器接收设备数据**

服务器通过AMQP客户端接收消息，详细说明请参见[AMQP客户端接入说明](#)，[Java SDK接入示例](#)。设备迁移场景AMQP收到消息接口参考[设备消息上报通知](#)。

AMQP客户端收到的消息样例，body字段携带设备上报的原始topic和payload。

```
{
 "resource": "device.message",
```

```
"event": "report",
"event_time": "20201114T034403Z",
"notify_data": {
 "header": {
 "app_id": "QAksSBSBBQpWYtEK3LrrOboNk0a",
 "device_id": "5fae45e358115902ce609882_20201113",
 "node_id": "20201113",
 "product_id": "5fae45e358115902ce609882",
 "gateway_id": "5fae45e358115902ce609882_20201113"
 },
 "body": {
 "topic": "/aircondition/data/up",
 "content": {
 "temperature": 26.0
 }
 }
}
```

JAVA核心代码样例如下:

```
package com.huawei.iot.amqp.jms;

import org.apache.qpid.jms.JmsConnection;
import org.apache.qpid.jms.JmsConnectionFactory;
import org.apache.qpid.jms.JmsConnectionListener;
import org.apache.qpid.jms.message.JmsInboundMessageDispatch;
import org.apache.qpid.jms.transports.TransportOptions;
import org.apache.qpid.jms.transports.TransportSupport;

import javax.jms.*;
import javax.naming.Context;
import javax.naming.InitialContext;
import java.net.URI;
import java.util.Hashtable;
import java.util.concurrent.ExecutorService;
import java.util.concurrent.LinkedBlockingQueue;
import java.util.concurrent.ThreadPoolExecutor;
import java.util.concurrent.TimeUnit;

public class HwIotAmqpJavaClientDemo {
 //异步线程池，参数可以根据业务特点作调整，也可以用其他异步方式来处理。
 private final static ExecutorService executorService = new
 ThreadPoolExecutor(Runtime.getRuntime().availableProcessors(),
 Runtime.getRuntime().availableProcessors() * 2, 60,
 TimeUnit.SECONDS, new LinkedBlockingQueue<>(5000));

 public static void main(String[] args) throws Exception {
 //连接凭证接入键值。
 String accessKey = "${yourAccessKey}";
 long timeStamp = System.currentTimeMillis();
 //UserName组装方法，请参见文档：AMQP客户端接入说明。
 String userName = "accessKey=" + accessKey + "|timestamp=" + timeStamp;
 //连接凭证接入码。
 String password = "${yourAccessCode}";
 //按照qpid-jms的规范，组装连接URL。
 String connectionUrl = "amqps://${UUCID}.iot-amqps.cn-north-4.myhuaweicloud.com:5671?
 amqp.vhost=default&amqp.idleTimeout=8000&amqp.saslMechanisms=PLAIN";
 Hashtable<String, String> hashtable = new Hashtable<>();
 hashtable.put("connectionfactory.HwConnectionURL", connectionUrl);
 //队列名，可以使用默认队列DefaultQueue
 String queueName = "${yourQueue}";
 hashtable.put("queue.HwQueueName", queueName);
 hashtable.put(Context.INITIAL_CONTEXT_FACTORY,
 "org.apache.qpid.jms.jndi.JmsInitialContextFactory");
 Context context = new InitialContext(hashtable);
 JmsConnectionFactory cf = (JmsConnectionFactory) context.lookup("HwConnectionURL");
 //同一个链接可创建多个queue,与前面queue.HwQueueName作好配对就行
 Destination queue = (Destination) context.lookup("HwQueueName");

 //信任服务端
 }
}
```

```
TransportOptions to = new TransportOptions(); to.setTrustAll(true);
cf.setSslContext(TransportSupport.createJdkSslContext(to));

// 创建连接
Connection connection = cf.createConnection(userName, password);
((JmsConnection) connection).addConnectionListener(myJmsConnectionListener);
// 创建 Session
// Session.CLIENT_ACKNOWLEDGE: 收到消息后, 需要手动调用message.acknowledge()。
// Session.AUTO_ACKNOWLEDGE: SDK自动ACK (推荐)。
Session session = connection.createSession(false, Session.AUTO_ACKNOWLEDGE);
connection.start();
// 创建 Receiver Link
MessageConsumer consumer = session.createConsumer(queue);
//处理消息有两种方式
// 1, 主动拉数据 (推荐), 参照receiveMessage(consumer)
// 2, 添加监听, 参照consumer.setMessageListener(messageListener), 服务端主动推数据给客户
端, 但得考虑接受的数据速率是客户能力能够承受住的
receiveMessage(consumer);
// consumer.setMessageListener(messageListener);
}

private static void receiveMessage(MessageConsumer consumer) throws JMSEException{
 while (true){
 try{
 // 建议异步处理收到的消息, 确保receiveMessage函数里没有耗时逻辑。
 Message message = consumer.receive(); processMessage(message);
 } catch (Exception e) {
 System.out.println("receiveMessage hand an exception: " + e.getMessage());
 e.printStackTrace();
 }
 }
}

private static MessageListener messageListener = new MessageListener(){
 @Override
 public void onMessage(Message message){
 try {
 // 建议异步处理收到的消息, 确保onMessage函数里没有耗时逻辑。
 // 如果业务处理耗时过程过长阻塞住线程, 可能会影响SDK收到消息后的正常回调。
 executorService.submit(() -> processMessage(message));
 } catch (Exception e){
 System.out.println("submit task occurs exception: " + e.getMessage());
 e.printStackTrace();
 }
 }
};

/**
 * 在这里处理您收到消息后的具体业务逻辑。
 */
private static void processMessage(Message message) {
 try {
 String body = message.getBody(String.class); String content = new String(body);
 System.out.println("receive an message, the content is " + content);
 } catch (Exception e){
 System.out.println("processMessage occurs error: " + e.getMessage());
 e.printStackTrace();
 }
}

private static JmsConnectionListener myJmsConnectionListener = new JmsConnectionListener(){
 /**
 * 连接成功建立。
 */
 @Override
 public void onConnectionEstablished(URI remoteURI){
 System.out.println("onConnectionEstablished, remoteUri:" + remoteURI);
 }
}
```

```

/**
 * 尝试过最大重试次数之后，最终连接失败。
 */
@Override
public void onConnectionFailure(Throwable error){
 System.out.println("onConnectionFailure, " + error.getMessage());
}

/**
 * 连接中断。
 */
@Override
public void onConnectionInterrupted(Uri remoteUri){
 System.out.println("onConnectionInterrupted, remoteUri:" + remoteUri);
}

/**
 * 连接中断后又自动重连上。
 */
@Override
public void onConnectionRestored(Uri remoteUri){
 System.out.println("onConnectionRestored, remoteUri:" + remoteUri);
}

@Override
public void onInboundMessage(ImsInboundMessageDispatch envelope){
 System.out.println("onInboundMessage, " + envelope);
}

@Override
public void onSessionClosed(Session session, Throwable cause){
 System.out.println("onSessionClosed, session=" + session + ", cause =" + cause);
}

@Override
public void onConsumerClosed(MessageConsumer consumer, Throwable cause){
 System.out.println("MessageConsumer, consumer=" + consumer + ", cause =" + cause);
}

@Override
public void onProducerClosed(MessageProducer producer, Throwable cause){
 System.out.println("MessageProducer, producer=" + producer + ", cause =" + cause);
}
};
}

```

- **业务服务器下发控制指令**

服务器通过调用消息下发接口下发控制指令，具体请参考[下发设备消息](#)。

下发设备消息关键参数说明：

| 名称           | 必选/<br>可选 | 类型     | 位置     | 说明                                                                                                                                    |
|--------------|-----------|--------|--------|---------------------------------------------------------------------------------------------------------------------------------------|
| X-Auth-Token | 必选        | String | Header | 用户Token。通过调用IAM服务 <a href="#">获取IAM用户Token</a> 接口获取，接口返回的响应消息头中“X-Subject-Token”就是需要获取的用户Token。简要的获取方法样例请参见 <a href="#">Token认证</a> 。 |



| 名称              | 必选/可选 | 类型          | 位置     | 说明                                                            |
|-----------------|-------|-------------|--------|---------------------------------------------------------------|
| Instance-Id     | 可选    | String      | Header | 实例ID。物理多租下各实例的唯一标识，一般华为云租户无需携带该参数，仅在物理多租场景下从管理面访问API时需要携带该参数。 |
| project_id      | 必选    | String      | Path   | 项目ID。获取方法请参见 <a href="#">获取项目ID</a> 。                         |
| device_id       | 必选    | String      | Path   | 填写迁移设备的设备ID。                                                  |
| message         | 必选    | String      | Body   | 设备执行的消息，字符串，具体格式需要应用和设备约定。                                    |
| topic_full_name | 可选    | String(128) | Body   | 迁移设备的topic。                                                   |

接口样例：

```
POST https://{Endpoint}/v5/iot/{project_id}/devices/{device_id}/messages
Content-Type: application/json
X-Auth-Token: *****
Instance-Id: *****
```

```
{
 "message": "{\"switch\": \"off\"}",
 "topic_full_name": "/aircondition/cmd"
}
```

JAVA核心代码样例如下：

```
public class DeviceMessage {
 public static void main(String[] args) throws KeyManagementException,
 NoSuchAlgorithmException, IOException {
 String token = Authentication.getToken();
 Map<String, String> headers = new HashMap<String, String>();
 headers.put("Content-Type", "application/json");
 headers.put("X-Auth-Token", token);

 Message message = new Message();
 message.setMessage("{\"switch\" : \"off\"}");
 message.setTopic_full_name("/aircondition/cmd");

 String projectId = "11111";
 String deviceId = "5fae45e358115902ce609882_20201113";
 String url = "https://iotda.cn-north-4.myhuaweicloud.com/v5/iot/%s/devices/%s/messages";
 url = String.format(url, projectId, deviceId);
 HttpUtils httpUtils = new HttpUtils();
 httpUtils.initClient();
 StreamClosedHttpResponse httpResponse = httpUtils.doPost(url, headers,
 JsonUtils.Obj2String(message));
 System.out.println(httpResponse.getContent());
 }
}
```

# 4 数据转发

## 4.1 指标数据推送至 DMS Kafka 最佳实践

### 场景说明

物联网平台为用户提供了丰富的报表功能，能够将数据直观地呈现出来，当您想将报表中的数据集成到自己的运维系统时，可以使用AOM的订阅功能将IoTDA推送的指标数据推送至DMS Kafka，您可以消费DMS Kafka中的指标数据展示在您自己的运维系统中。本示例为使用AOM订阅功能将指标数据推送至DMS Kafka，并以在线设备数指标为例展示。IoTDA支持的指标数据可参考[IoTDA指标详情](#)。

### 前提条件

- 已注册华为官方账号。未注册可参考[注册页面](#)注册华为账户完成注册。
- 已开通设备接入服务。未开通则访问[设备接入服务](#)，单击“免费试用”或单击“价格计算器”购买并开通该服务。
- 已开通AOM服务。访问[AOM服务](#)，单击“AOM2.0控制台”，单击“免费开通服务并授权”开通该服务。
- 已开通DMS Kafka服务。未开通则访问[DMS Kafka服务](#)，单击“立即购买”后开通该服务。

### 创建 AOM 订阅

**步骤1** 访问[设备接入服务](#)，单击“管理控制台”进入设备接入控制台。

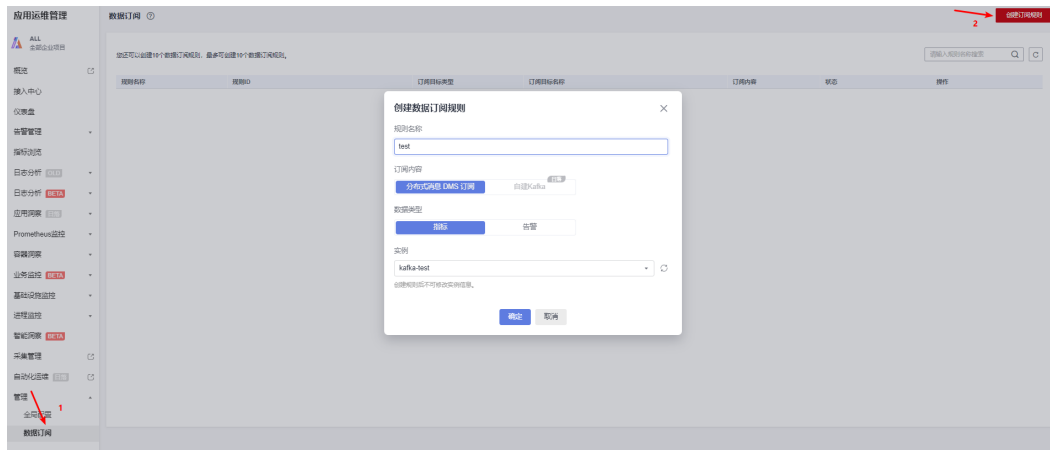
**步骤2** 选择左侧导航栏“监控运维 > 统计报表 > 创建数据订阅 > 前往AOM”进入应用运维管理页面。

图 4-1 创建数据订阅-前往 AOM



**步骤3** 选择左侧导航栏“管理 > 数据订阅 > 创建订阅规则”选择订阅内容，数据类型以及DMS实例信息，单击“确定”。

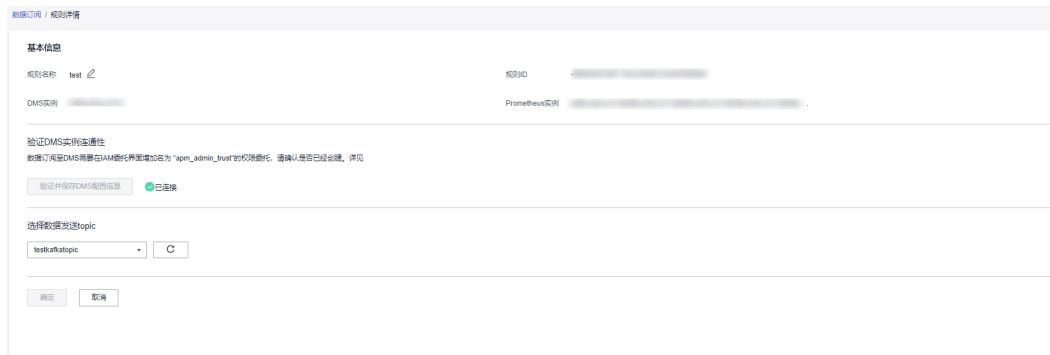
图 4-2 创建数据订阅-AOM



**步骤4** 首次配置DMS订阅时，需配置IAM委托，可参考[如何创建委托apm\\_admin\\_trust](#)。

**步骤5** 单击“验证并保存DMS配置信息”校验AOM与DMS的连通性，显示已连接后，选择数据发送topic，单击“确定”。

图 4-3 规则详情-AOM



----结束

## 模拟设备上线及结果验证

**步骤1** 使用MQTT模拟器连接到平台，模拟设备上线。（模拟器使用请参考：[使用MQTT.fx调测](#)）。

**步骤2** 等待一分钟左右后，前往[Kafka控制台](#)，选择“消息查询”，topic名称填写AOM中配置的topic，查询内容填写设备所在的资源空间ID，查看收到转发的设备总数指标中是否包含在线设备数的指标信息。

图 4-4 查看 kafka 消息-AOM



----结束

# 5 设备联动

## 5.1 设备触发告警并邮件或短信通知

### 场景说明

很多物联网设备24小时运行，设备管理者并不需要知道设备每一刻的状态，只需要设备在上报某些特定状态时能通知到管理者。

设备接入服务提供规则引擎功能可以满足该诉求，您可以通过设置规则简单实现当设备上报的数据满足某个条件时，物联网平台触发指定动作进行通知。

本示例为设备上报的电池电量小于20%时，物联网平台会上报告警并发送邮件或短信通知给指定的手机号码。

### 配置消息通知服务

在消息通知服务中创建主题并添加订阅，供设备接入服务调用以发送邮件或短信。

1. 登录华为云官方网站，访问[消息通知服务](#)。
2. 单击“立即使用”进入消息通知服务控制台。若您未开通消息通知服务，请先开通服务。
3. 进入“主题管理 > 主题”页面，单击“创建主题”。
4. 输入主题名称，如“Test\_1”，单击“确定”。

图 5-1 创建主题-SMN

5. 进入“主题管理 > 订阅”页面，单击“添加订阅”。
6. 填写订阅信息，填写完成后单击“确定”。

图 5-2 添加订阅-SMN

表 5-1 添加订阅参数填写说明

| 参数名称 | 说明                                                                                                                                      |
|------|-----------------------------------------------------------------------------------------------------------------------------------------|
| 主题名称 | 选择步骤4创建的主题。                                                                                                                             |
| 协议   | <ul style="list-style-type: none"> <li>若需要发送邮件通知，选择“邮件”。</li> <li>若需要发送短信通知，选择“短信”。</li> </ul>                                          |
| 订阅终端 | <ul style="list-style-type: none"> <li>若协议为“邮件”，填写接收通知邮件的邮箱地址；</li> <li>若协议为“短信”，输入接收短信通知的手机号码。</li> </ul> 需要添加多个终端时，每个终端地址占1行，最多填写10行。 |

## 配置设备接入服务

在设备接入服务中创建产品模型、注册设备并设置设备联动规则，实现当设备上报特定数据时触发告警并发送邮件或短信。

1. 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。
2. 选择左侧导航栏的“产品”。

**注：**本文中使用的产品模型和设备仅为示例，您可以使用自己的产品模型和设备进行操作。

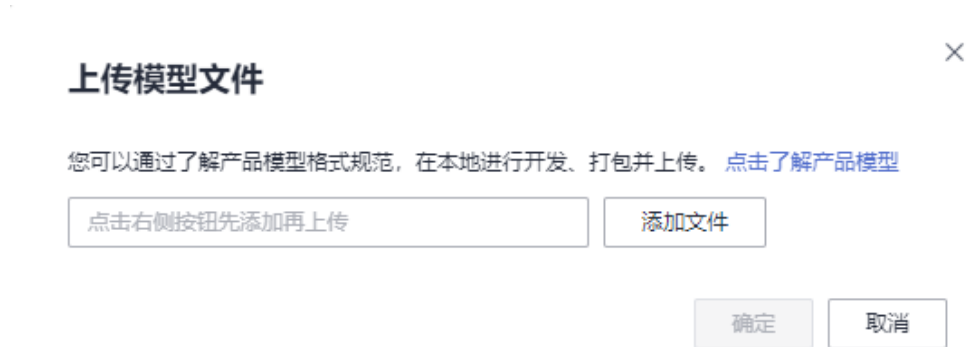
3. 单击左侧的“创建产品”，创建一个基于MQTT协议的产品，填写参数后，单击“确认”。

表 5-2 创建产品参数示例

| 基本信息 |                  |
|------|------------------|
| 产品名称 | 自定义，如MQTT_Device |
| 协议类型 | 选择“MQTT”         |
| 数据格式 | 选择“JSON”         |
| 所属行业 | 根据实际情况进行填写。      |
| 设备类型 |                  |

4. 单击[文件](#)，获取产品模型文件样例。
5. 找到新增的产品，单击“产品”进入产品详情界面。
6. 在模型定义页面，单击“上传模型文件”，在弹出的页面中加载产品模型文件，然后单击“确认”。

图 5-3 产品-上传模型文件



7. 选择左侧导航栏的“设备 > 所有设备 > 设备注册”，单击右上角的“注册设备”，填写设备注册参数。填写完成后单击“确定”，请注意保存注册成功返回的“设备ID”和“设备密钥”。

图 5-4 单设备注册-MQTT



表 5-3 注册设备参数填写说明

| 参数名称 | 说明            |
|------|---------------|
| 所属产品 | 选择在步骤3中创建的产品。 |



| 参数名称   | 说明                                                               |
|--------|------------------------------------------------------------------|
| 设备标识码  | 即node_id，填写为设备的IMEI、MAC地址或Serial No；若没有真实设备，填写自定义字符串，由英文字母和数字组成。 |
| 设备名称   | 自定义。                                                             |
| 设备认证类型 | 选择“密钥”。                                                          |
| 密钥     | 设备密钥，可自定义，不填写物联网平台会自动生成。                                         |

- 选择左侧导航栏的“规则>设备联动”，单击右上角的“创建规则”按钮。（创建规则前需选择所属资源空间。）
- 参考下表参数说明，填写规则内容。以下参数取值仅为示例，您可参考[用户指南](#)创建自己的规则，填写完成后单击“创建规则”。

图 5-5 新建联动规则-BatteryProperty

表 5-4 创建联动规则参数填写说明

| 参数名称 | 说明                                          |
|------|---------------------------------------------|
| 规则名称 | 自定义，例如“Battery_Low_Mail”或“Battery_Low_SMS”。 |
| 激活规则 | 勾选“激活规则”                                    |
| 生效时间 | 选择“一直生效”。                                   |
| 描述   | 自定义对该规则的描述，如“设备上报电量低于20%时上报告警并发送通知”。        |

| 参数名称 | 说明                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
|------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 触发条件 | <ol style="list-style-type: none"> <li>1. 单击“添加条件”；</li> <li>2. 选择“设备属性触发”；</li> <li>3. 选择3中添加的产品，选择“指定设备”，再选择7中添加的设备；</li> <li>4. “服务类型”选择“Battery”，“属性”选择“batteryLevel”，“操作”选择“&lt;”，“值”填写“20”，单击“触发机制”，“触发策略”选择“重复抑制”，数据时效”填写“3600”，然后单击“确认”。</li> </ol>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
| 执行动作 | <p>首先添加告警：</p> <ol style="list-style-type: none"> <li>1. 单击“添加动作”；</li> <li>2. 动作类型选择“上报告警”；</li> <li>3. “告警等级”选择“次要”，“告警隔离级别”选择“设备”，“告警名称”输入“设备电量低”，“告警内容”输入“设备电量低于20%，请及时确认并更换电池。”，然后单击“确认”。</li> </ol> <p>再添加通知：</p> <ol style="list-style-type: none"> <li>1. 单击“添加动作”。</li> <li>2. 动作类型”选择“发送通知”。</li> <li>3. 选择您开通的SMN服务区域，如“cn-north-4”。首次创建对接到SMN服务的规则时，平台会根据对接的云服务和区域弹出对应的云服务访问授权窗口，请根据界面提示，配置云服务访问授权。（可登录SMN服务控制台后在左上角查看）。</li> <li>4. 主题名称选择<a href="#">配置消息通知服务</a>时创建的主题； <ul style="list-style-type: none"> <li>● 若主题对应的订阅协议是邮件，消息标题填写为邮件标题，如“[Huawei IoT Platform] Low Battery Warning”；消息内容填写“You have a device with less than 20% charge, please log in to the Huawei IoT Platform for details.”</li> <li>● 若主题对应的订阅协议是短信，消息标题无需填写，消息内容填写“[Huawei IoT Platform] You have a device with less than 20% charge, please log in to the Huawei IoT Platform for details.”</li> </ul> </li> </ol> |

## 验证操作

- 您可以使用配置设备接入服务时注册的真实设备接入平台，上报“batteryLevel”小于20的数据。
- 您也可以使用模拟器模拟设备上报“batteryLevel”小于20的数据，操作方法请参考[在线开发MQTT协议的模拟智慧路灯](#)。
- 您还可以使用虚拟设备进行在线调试，上报“batteryLevel”小于20的数据。

期望结果：

- 在左侧导航栏选择“监控运维 > 当前告警”，单击“前往AOM”，跳转至应用运维管理服务，有一条名称为“设备电量低”的一般告警。
- 若订阅了邮件通知，接受订阅的邮箱会收到一封通知设备电量低的邮件。
- 若订阅了短信通知，接受订阅的手机会收到一条通知设备电量低的短信。

## 5.2 温度过高时自动关闭设备

### 场景说明

物联网平台支持设备数据上报和设备命令下发，但若要将两者联动起来，一般需要由应用服务器实现对应逻辑。

设备接入服务提供规则引擎功能，平台上简单几步操作即可实现数据上报特定数据时平台自动下发指定命令，减少应用服务器开发工作量。

本示例为设备自带的温度传感器上报的温度大于80度时，平台自动下发命令关闭设备。

### 配置设备接入服务

在设备接入服务中创建产品模型、注册设备并设置设备联动规则，实现当设备上报特定数据时平台自动下发命令。

1. 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。
2. 选择左侧导航栏的“产品”。

**注：**本文中使用的产品模型和设备仅为示例，您可以使用自己的产品模型和设备进行操作。

3. 单击右上角的“创建产品”，创建一个基于MQTT协议的产品，填写参数后，单击“确认”。

表 5-5 创建产品参数示例

| 基本信息 |                  |
|------|------------------|
| 产品名称 | 自定义，如MQTT_Device |
| 协议类型 | 选择“MQTT”         |
| 数据格式 | 选择“JSON”         |
| 厂商名称 | 自定义              |
| 所属行业 | 根据实际情况进行填写。      |
| 设备类型 |                  |

4. 单击[Profile\\_tempSensor.zip](#)，获取产品模型文件样例。
5. 在模型定义页面，单击“上传模型文件”，在弹出的页面中加载产品模型文件，然后单击“确认”。

图 5-6 产品-上传模型文件



6. 选择左侧导航栏的“设备 > 所有设备 > 设备注册”，单击右上角的“注册设备”，填写设备注册参数。

图 5-7 单设备注册-MQTT

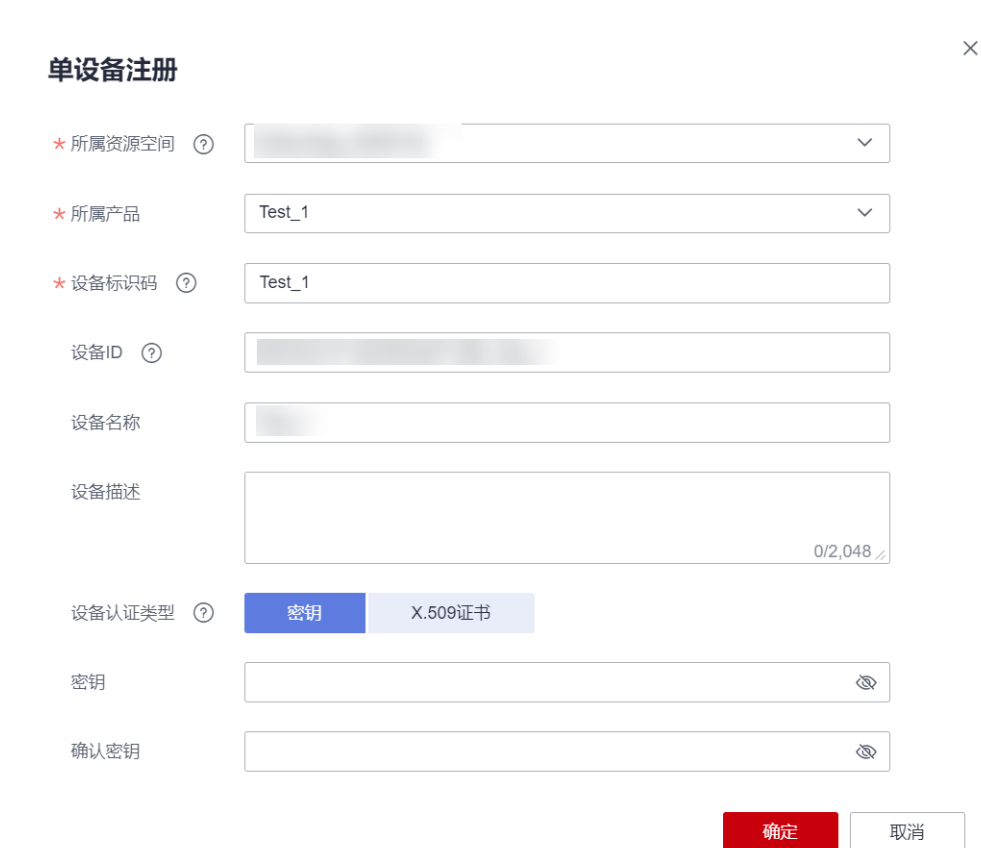


表 5-6 注册设备参数填写说明

| 参数名称  | 说明                                                               |
|-------|------------------------------------------------------------------|
| 所属产品  | 选择在步骤3中创建的产品。                                                    |
| 设备标识码 | 即node_id，填写为设备的IMEI、MAC地址或Serial No；若没有真实设备，填写自定义字符串，由英文字母和数字组成。 |

| 参数名称   | 说明                       |
|--------|--------------------------|
| 设备名称   | 自定义。                     |
| 设备认证类型 | 选择“密钥”。                  |
| 密钥     | 设备密钥，可自定义，不填写物联网平台会自动生成。 |

填写完成后单击“确定”，请注意保存注册成功返回的“设备ID”和“设备密钥”。

7. 选择左侧导航栏的“规则>设备联动”，单击右上角的“创建规则”按钮。（创建规则前需选择所属资源空间。）
8. 参考下表参数说明，填写规则内容。以下参数取值仅为示例，您可参考[用户指南](#)创建自己的规则，填写完成后单击“创建规则”。

图 5-8 新建联动规则-TemperatureProperty

表 5-7 创建联动规则参数填写说明

| 参数名称 | 说明                           |
|------|------------------------------|
| 规则名称 | 创建的规则名称，例如“Overheated”。      |
| 立即触发 | 勾选“激活规则”                     |
| 生效时间 | 选择“一直生效”。                    |
| 描述   | 对该规则的描述，如“设备温度高于80度时自动关闭设备”。 |

| 参数名称 | 说明                                                                                                                                                                                                                                             |
|------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 触发条件 | <ol style="list-style-type: none"><li>1. 单击“添加条件”；</li><li>2. 选择“设备属性触发”；</li><li>3. 选择步骤3中添加的产品，选择指定设备，再选择步骤6中添加的设备；</li><li>4. 服务选择“tempSensor”，属性选择“temperature”，操作选择“&gt;”，值填写“80”，单击“触发机制”，“触发策略”选择“重复抑制”，数据时效填写“300”，然后单击“确定”。</li></ol> |
| 执行动作 | <ol style="list-style-type: none"><li>1. 单击“添加动作”；</li><li>2. “动作类型”选择“下发命令”，下发设备选择步骤6创建的设备；</li><li>3. 服务选择“deviceSwitch”，命令选择“ON_OFF”，单击“参数配置”，“power”的值选择“OFF”，然后单击“确定”。</li></ol>                                                          |

## 验证操作

- 您可以使用配置设备接入服务时注册的真实设备接入平台，上报“temperature”大于80的数据。
- 您也可以使用模拟器模拟设备订阅Topic：“\$oc/devices/{device\_id}/sys/properties/report”（其中{deviceid}需要替换成实际设备ID）后，上报“temperature”大于80的数据，操作方法请参考[在线开发MQTT协议的模拟智慧路灯](#)。
- 您还可以使用虚拟设备进行在线调试，上报“temperature”大于80的数据。

期望结果：

- 若您使用真实设备上报数据，设备会收到一条“power”取值为“OFF”的“ON\_OFF”命令。
- 若您使用模拟器上报数据，可在“Subscribe”页签查看到平台下发的“power”取值为“OFF”的“ON\_OFF”命令。

## 5.3 煤气浓度过高自动打开推窗器

### 场景说明

煤气安全关系到千家万户的生命财产安全。本示例基于华为物联网平台提供的设备接入服务，通过触发设备联动规则，自动打开推窗器。具体场景是煤气监测器将煤气浓度数值上报到物联网平台，当煤气浓度超标时，会触发设置好的设备联动规则，物联网平台会下发打开窗户的命令到无线推窗器，无线推窗器收到命令后，自动把窗户打开。

### 创建煤气监测产品

**步骤1** 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。

**步骤2** 选择左侧导航栏的“产品”。

**步骤3** 单击左侧的“创建产品”，创建煤气浓度监测产品，填写参数后，单击“确定”。

**表 5-8** 创建产品参数示例

| 基本信息 |                 |
|------|-----------------|
| 产品名称 | 自定义，如gasdevice。 |
| 协议类型 | 选择“MQTT”        |
| 数据格式 | 选择“JSON”        |
| 所属行业 | 自定义             |
| 设备类型 |                 |

**步骤4** 在模型定义页面，单击“自定义模型”，参考下表，完成产品模型配置。gaslevel服务用于监控煤气浓度，windowswitch服务用于执行开关窗命令。

**表 5-9** 煤气浓度监测产品模型参数示例

| 服务ID         | 参数类型 | 添加参数说明内容                                                 |
|--------------|------|----------------------------------------------------------|
| gaslevel     | 属性   | 属性名称：gaslevel<br>数据类型：int<br>访问权限：可读<br>取值范围：0~100       |
| windowswitch | 命令   | 命令名称：switch<br>下发参数名称：switch<br>数据类型：enum<br>枚举值：on, off |

----结束

## 注册设备

**步骤1** 选择左侧导航栏的“设备 > 所有设备 > 设备注册”，单击右上角的“注册设备”，填写设备注册参数。注册煤气浓度监测设备，并记录设备ID和密钥。

图 5-9 单设备注册-gasdevice

**单设备注册** ×

\* 所属资源空间 ?

\* 所属产品   
MQTT类型的设备已默认订阅平台预置topic, [查看已订阅topic列表](#)

\* 设备标识码 ?

设备ID ?

设备名称

设备描述   
0/2,048

设备认证类型 ?  密钥  X.509证书

密钥

确认密钥

表 5-10 注册设备参数填写说明

| 参数名称   | 说明                                                                   |
|--------|----------------------------------------------------------------------|
| 所属产品   | 选择在创建煤气监测产品步骤3中创建的产品。                                                |
| 设备标识码  | 即node_id, 填写为设备的IMEI、MAC地址或Serial No; 若没有真实设备, 填写自定义字符串, 由英文字母和数字组成。 |
| 设备名称   | 自定义。                                                                 |
| 设备认证类型 | 选择“密钥”。                                                              |
| 密钥     | 设备密钥, 可自定义, 不填写物联网平台会自动生成。                                           |

**步骤2** 选择左侧导航栏的“设备 > 所有设备 > 设备注册”，单击右上角的“注册设备”，填写设备注册参数。注册推窗器设备，并记录设备ID和密钥。



图 5-10 单设备注册-windowswitch

**单设备注册**
×

\* 所属资源空间 ?

\* 所属产品   
MQTT类型的设备已默认订阅平台预置topic, [查看已订阅topic列表](#)

\* 设备标识码 ?

设备ID ?

设备名称

设备描述   
0/2,048

设备认证类型 ? 密钥 X.509证书

密钥

确认密钥

确定
取消

表 5-11 注册设备参数填写说明

| 参数名称   | 说明                                                                   |
|--------|----------------------------------------------------------------------|
| 所属产品   | 选择在步骤3中创建的产品。                                                        |
| 设备标识码  | 即node_id, 填写为设备的IMEI、MAC地址或Serial No; 若没有真实设备, 填写自定义字符串, 由英文字母和数字组成。 |
| 设备名称   | 自定义。                                                                 |
| 设备认证类型 | 选择“密钥”。                                                              |
| 密钥     | 设备密钥, 可自定义, 不填写物联网平台会自动生成。                                           |

----结束

## 配置设备联动规则

- 步骤1** 选择左侧导航栏的“规则>设备联动”，单击右上角的“创建规则”按钮。（创建规则前需选择所属资源空间。）
- 步骤2** 参考下表参数说明，填写规则内容。以下参数取值仅为示例，您可参考[用户指南](#)创建自己的规则，填写完成后单击“创建规则”。

图 5-11 新建联动规则-GasLevel

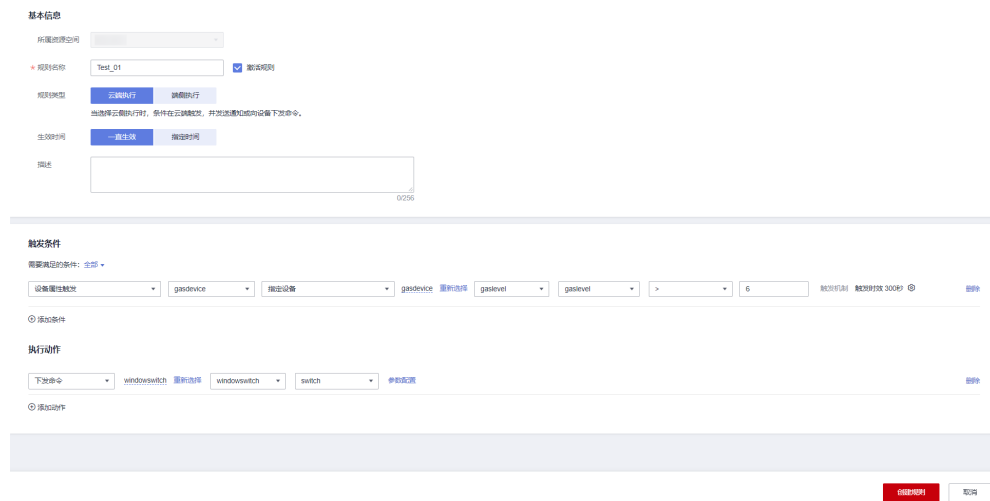


表 5-12 创建联动规则参数填写说明

| 参数名称 | 说明                                                                                                                                                                                                                                                       |
|------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 规则名称 | 创建的规则名称，例如“windowswitch”。                                                                                                                                                                                                                                |
| 立即触发 | 勾选“激活规则”                                                                                                                                                                                                                                                 |
| 生效时间 | 选择“一直生效”。                                                                                                                                                                                                                                                |
| 描述   | 对该规则的描述，如“煤气浓度高于6时自动打开推窗设备”。                                                                                                                                                                                                                             |
| 触发条件 | <ol style="list-style-type: none"> <li>1. 单击“添加条件”；</li> <li>2. 选择“设备属性触发”；</li> <li>3. 选择3中添加的产品，选择“指定设备”后选择1中添加的设备；</li> <li>4. “服务类型”选择“gaslevel”，“属性名字”选择“gaslevel”，“操作”选择“&gt;”，“值”填写“6”，单击“触发机制”，“触发策略”选择“重复抑制”，“数据时效”填写“300”，然后单击“确定”。</li> </ol> |
| 执行动作 | <ol style="list-style-type: none"> <li>1. 单击“添加动作”；</li> <li>2. “动作类型”选择“下发命令”，下发设备选择步骤2创建的设备；</li> <li>3. “服务类型”选择“windowswitch”，“命令名字”选择“on_off”，单击“参数配置”，“switch”的值选择“on”，然后单击“确定”。</li> </ol>                                                        |

----结束

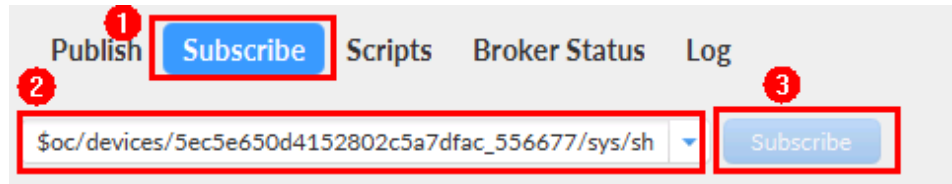
## 验证操作

### 方法一：

您可以使用MQTT.fx模拟设备验证。

1. 使用MQTT.fx分别模拟煤气探测器设备和推窗器，并连接到物联网平台。操作方法请参考[在线开发MQTT协议的模拟智慧路灯](#)。
2. 打开模拟推窗器的MQTT.fx，订阅平台下发的命令。
  - a. 选择“Subscribe”页签。
  - b. 输入订阅平台下发命令的Topic=\$oc/devices/{device\_id}/sys/commands/#（其中与步骤2获取的设备ID保持一致。）。
  - c. 单击“Subscribe”按钮，完成平台命令下发订阅。

图 5-12 创建 MQTT 订阅



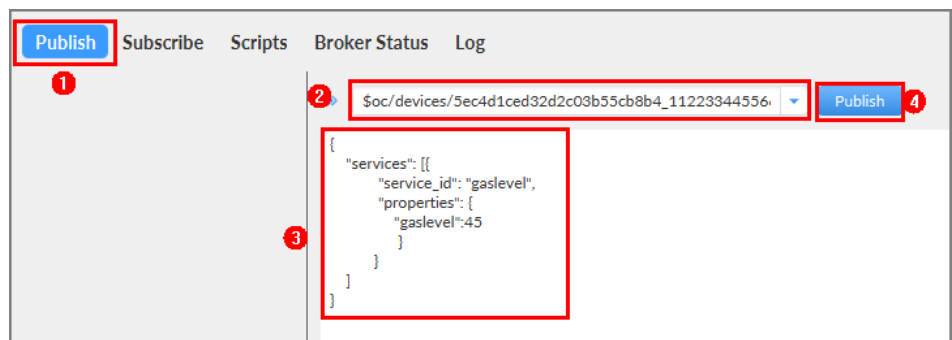
3. 切换到模拟煤气探测的MQTT.fx，上报属性。
  - a. 选择“Publish”页签。
  - b. 输入属性上报的Topic：“\$oc/devices/{device\_id}/sys/properties/report（其中{deviceid}与步骤1获取的设备ID保持一致。）。
  - c. 上报1条“gaslevel”大于6的属性。

样例：

```
{
 "services": [{
 "service_id": "gaslevel",
 "properties": {
 "gaslevel": 45
 }
 }
]
```

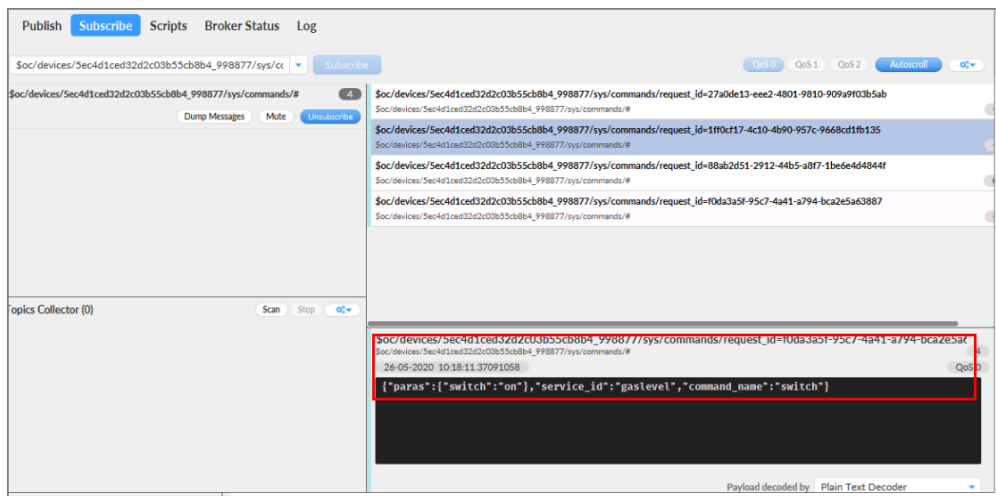
- d. 单击“Publish”按钮上报属性值。

图 5-13 MQTT 属性上报



4. 切换到模拟推窗器的MQTT.fx，选择“Subscribe”页签。可以看到收到平台下发的“switch”取值为“on”的命令。

图 5-14 查看下发命令



**方法二：**

您可以使用配置设备接入服务时注册的真实设备接入平台，上报“gaslevel”大于6的数据。设备会收到一条“switch”取值为“on”的命令，自动推开窗户。

## 5.4 监测设备状态变化情况并发送通知

### 场景说明

在物联网场景下，部分设备具备重要的应用场景，比如物联网网关等，设备管理者需要感知这些设备的上下线情况。

设备接入服务提供规则引擎功能可以满足该诉求，您可以通过简单的操作实现当设备状态满足某个条件时，物联网平台触发指定动作进行通知。

**示例说明：**

某企业的网关产品下有一批网关设备，单个网关设备下挂载了约400个子设备，用户需要实时关注这批网关设备的状态，确保子设备正常上报数据，同时由于网关设备和物联网平台通过4G网络建立连接，存在由于网络抖动导致频繁告警的问题，因此用户认为短暂的离线后上线属于正常场景，不希望感知这种场景。

通过以下示例可以实现监控该网关产品下所有网关设备在离线持续时间达到5分钟后通过物联网平台上报告警，在设备上线持续时间达到1分钟后恢复该告警，同时发送邮件或短信通知给指定的手机号码。

### 对接流程

1. **配置设备接入服务**：创建物联网产品和设备，创建联动规则，实现设备上下线满足特定条件后发送告警到应用运维管理服务。
2. **配置消息通知服务**：创建短信或邮件订阅。
3. **配置应用运维管理服务**：创建告警规则，处理由设备接入服务上报上来的告警，并通过消息通知服务发送短信或邮件通知。

## 配置设备接入服务

在设备接入服务中创建产品模型、注册设备并设置设备联动规则，实现当设备离线达到5分钟后上报告警到应用运维管理平台，设备上线1分钟后恢复该告警。

### 说明

本文中使用的产品模型和设备仅为示例，您可以使用自己的产品模型和设备进行操作。

- 步骤1** 访问[设备接入服务](#)，单击“立即使用”进入设备接入控制台。
- 步骤2** 选择左侧导航栏的“产品”，单击页面左侧的“创建产品”，创建一个基于MQTT协议的产品，填写参数后，单击“确认”，完成产品的创建。

表 5-13 创建产品参数示例

| 基本信息 |                   |
|------|-------------------|
| 产品名称 | 自定义，如MQTT_Device。 |
| 协议类型 | 选择“MQTT”。         |
| 数据格式 | 选择“JSON”。         |
| 所属行业 | 根据实际情况进行填写。       |
| 设备类型 |                   |

- 步骤3** 单击[文件](#)，获取产品模型文件样例。
- 步骤4** 找到新增的产品，单击“产品”进入产品详情界面。
- 步骤5** 在产品详情“模型定义”页面，单击“上传模型文件”，在弹出的页面中加载产品模型文件，然后单击“确定”。

图 5-15 产品-上传模型文件



- 步骤6** 选择左侧导航栏的“设备 > 设备注册”，单击右上角的“注册设备”，填写设备注册参数。填写完成后单击“确定”，请注意保存注册成功返回的“设备ID”和“设备密钥”。

图 5-16 单设备注册-gateway

**单设备注册**
×

\* 所属资源空间 ?

\* 所属产品  ▼  
MQTT类型的设备已默认订阅平台预置topic, [查看已订阅topic列表](#)

\* 设备标识码 ?

设备ID ?

设备名称

设备描述   
0/2,048

设备认证类型 ? 密钥 X.509证书

密钥

确认密钥

确定
取消

表 5-14 注册设备参数填写说明

| 参数名称   | 说明                                                                   |
|--------|----------------------------------------------------------------------|
| 所属产品   | 选择在步骤4中创建的产品。                                                        |
| 设备标识码  | 即node_id, 填写为设备的IMEI、MAC地址或Serial No; 若没有真实设备, 填写自定义字符串, 由英文字母和数字组成。 |
| 设备名称   | 自定义。                                                                 |
| 设备认证类型 | 选择“密钥”。                                                              |
| 密钥     | 设备密钥, 可自定义, 不填写物联网平台会自动生成。                                           |

**步骤7** 选择左侧导航栏的“规则>设备联动”，单击右上角的“创建规则”按钮。（创建规则前需选择所属资源空间）

**步骤8** 参考下表参数说明，填写规则内容，创建网关离线上报告警规则。以下参数取值仅为示例，您可参考[用户指南](#)创建自己的告警规则，填写完成后单击“创建规则”。

表 5-15 创建联动规则参数填写说明

| 参数名称 | 步骤说明                                                                                                                                                                                        |
|------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 规则名称 | 自定义，例如网关离线规则“Gateway_Offline_Monitor”。                                                                                                                                                      |
| 激活规则 | 勾选“激活规则”。                                                                                                                                                                                   |
| 生效时间 | 选择“一直生效”。                                                                                                                                                                                   |
| 描述   | 对该规则的描述，如“网关设备离线持续时长达到5分钟上报告警。”。                                                                                                                                                            |
| 触发条件 | <ol style="list-style-type: none"> <li>1. 单击“添加条件”；</li> <li>2. 选择“设备状态触发”；</li> <li>3. 选择步骤4中添加的产品“MQTT_Device”，选择“全部设备”，触发状态选择“下线”；</li> <li>4. “状态持续时长”选择“5分钟”。</li> </ol>               |
| 执行动作 | <ol style="list-style-type: none"> <li>1. 单击“添加动作”；</li> <li>2. 动作类型选择“上报告警”；</li> <li>3. “告警等级”选择“重要”，告警隔离级别选择“设备”维度隔离，“告警名称”输入“网关状态变更”，“告警内容”输入“网关设备离线，请及时确认并安排人员进行维护”并单击“确定”。</li> </ol> |

图 5-17 新建联动规则-GatewayOffline

**步骤9** 创建网关上线恢复告警规则。

图 5-18 新建联动规则-GatewayOnline

### 须知

- 告警名称、级别和维度组合起来共同标识一条AOM（应用运维管理）告警，恢复的告警和上报的告警需要保持同一告警属性，否则恢复告警失败。
- 设备状态监控有流控限制，监控设备量过多会触发流控导致设备离线告警无法上报的情况，详细请参考[使用限制](#)。

----结束

## 配置消息通知服务

在消息通知服务中创建主题并添加订阅，供应用运维管理服务调用以发送邮件或短信。

- 步骤1** 登录华为云官方网站，访问[消息通知服务](#)。
- 步骤2** 单击“立即使用”进入消息通知服务控制台。若您未开通消息通知服务，请先开通服务。
- 步骤3** 进入“主题管理 > 主题”页面，单击“创建主题”。
- 步骤4** 输入主题名称，如“Test\_1”，单击“确定”。



图 5-19 创建主题-SMN

创建主题

\* 主题名称  ?  
主题创建后, 不允许修改主题名称。

显示名

\* 企业项目  ? 新建企业项目

启动日志记录

标签  
如果您需要使用同一标签标识多种云资源, 即所有服务均可在标签输入框下拉选择同一标签, 建议在TMS中创建预定义标签。 [查看预定义标签](#)  
在下方键/值输入框输入内容后单击添加, 即可将标签加入此处

您还可以添加20个标签。

步骤5 进入“主题管理 > 订阅”页面，单击“添加订阅”。

步骤6 填写订阅信息。填写完成后单击“确定”。

图 5-20 添加订阅-SMN

添加订阅

主题名称 Test\_1

\* 协议

\* 订阅终端 ? 终端 备注  
  
+ 添加订阅终端  
批量添加订阅终端

表 5-16 添加订阅参数填写说明

| 参数名称 | 说明                                                                                                                                      |
|------|-----------------------------------------------------------------------------------------------------------------------------------------|
| 主题名称 | 选择 <b>步骤4</b> 创建的主题。                                                                                                                    |
| 协议   | <ul style="list-style-type: none"> <li>若需要发送邮件通知，选择“邮件”。</li> <li>若需要发送短信通知，选择“短信”。</li> </ul>                                          |
| 订阅终端 | <ul style="list-style-type: none"> <li>若协议为“邮件”，填写接收通知邮件的邮箱地址；</li> <li>若协议为“短信”，输入接收短信通知的手机号码。</li> </ul> 需要添加多个终端时，每个终端地址占1行，最多填写10行。 |

---结束

## 配置应用运维管理服务

在应用运维管理服务中创建告警规则和告警行动规则，当设备接入服务满足条件上报告警后，对上报告警进行处理并发送邮件或短信。

- 步骤1** 登录华为云官方网站，访问[应用运维管理服务](#)。
- 步骤2** 单击“立即使用”进入应用运维管理服务控制台。若您未开通应用运维管理服务，请先开通服务。
- 步骤3** 进入“告警 > 告警行动规则”页面，单击“创建告警行动规则”。
- 步骤4** 输入告警行动规则名称，如“Test\_1”，主题选择[配置消息通知服务](#)中创建的主题“Test\_1”，单击“确定”。

图 5-21 创建行动规则-AOM

创建告警行动规则

\* 行动规则名称 ?

\* 企业项目

描述 ?

\* 行动规则类型  指标或事件  日志

\* 行动方式

\* 主题  C

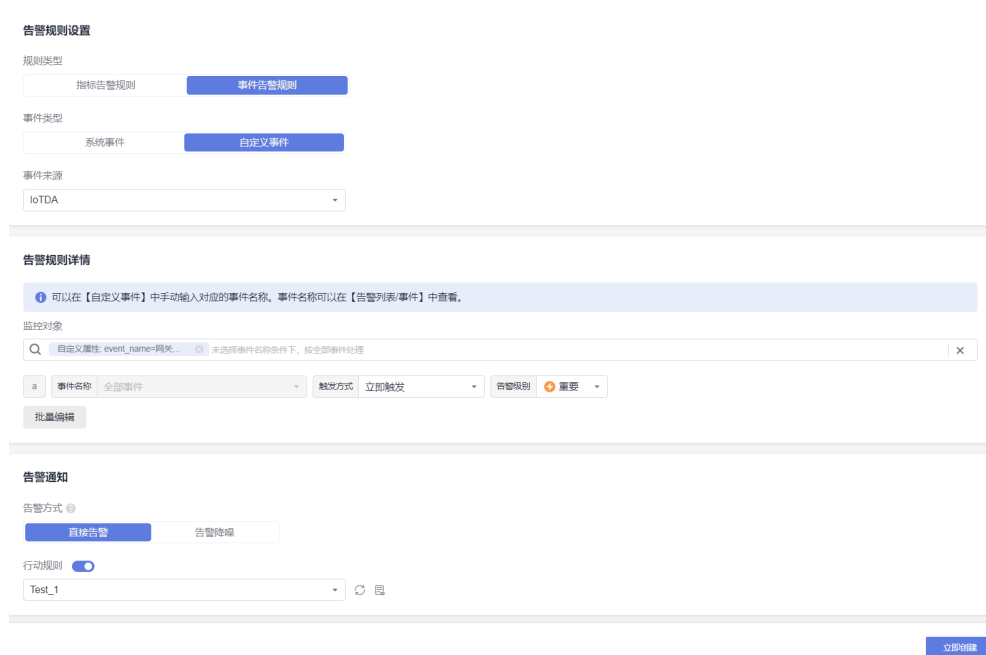
若没有您想要选择的主题，请单击 [创建主题](#)，在SMN界面新建主题

\* 消息模板  C [创建消息模板](#) | [查看消息模板](#)

- 步骤5** 进入“告警 > 告警规则”页面，单击“添加告警”。
- 步骤6** 输入规则名称，如“Gateway\_Status\_Change\_Alarm\_Rule”，选择“事件告警规则 > 自定义事件”，告警来源选择“IoTDA”，监控对象选择自定义属性“event\_name=

网关状态变更”（“=”后面内容为告警名称），触发策略选择“立即触发”，告警方式选择直接告警，行动规则选择步骤4中创建的行动规则，单击右下角“立即创建”。

图 5-22 创建告警规则-AOM



----结束

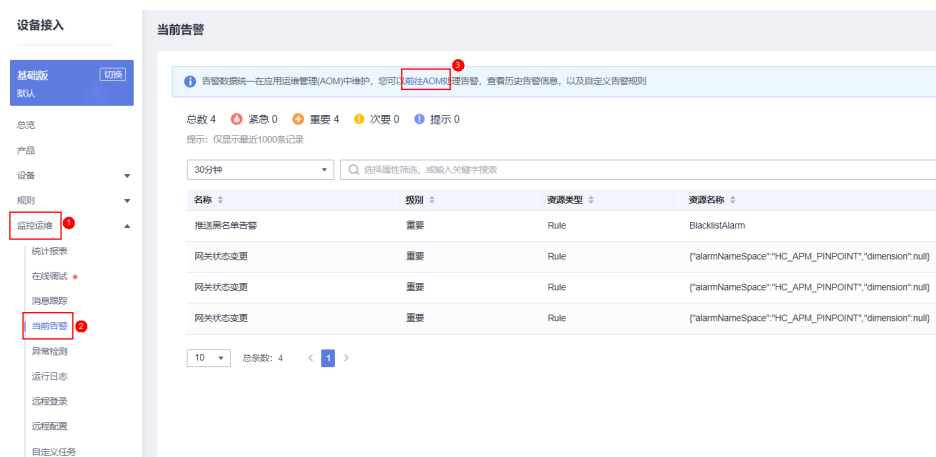
## 验证操作

- 您可以使用配置设备接入服务时注册的真实设备接入平台。
- 您也可以使用模拟器模拟设备上下线，操作方法请参考[在线开发MQTT协议的模拟智慧路灯](#)。

期望结果：

1. 设备离线持续时长达到5分钟后：
  - 在左侧导航栏选择“监控运维 > 当前告警”，单击“前往AOM”，跳转至应用运维管理服务，有一条名称为“网关状态变更”的重要告警。

图 5-23 当前告警-前往 Aom



- 若在应用运维管理服务配置了告警规则和邮件通知告警行动规则，接受订阅的邮箱会收到一封通知网关离线的邮件。
  - 若在应用运维管理服务配置了告警规则和短信通知告警行动规则，接受订阅的手机会收到一条通知网关离线的短信。
2. 设备上线持续时长达到1分钟后：
- 该设备的“网关状态变更”重要告警恢复，可在历史告警中查看。
  - 若在应用运维管理服务配置了告警规则和邮件通知告警行动规则，接受订阅的邮箱会收到一封网关离线恢复的邮件。
  - 若在应用运维管理服务配置了告警规则和短信通知告警行动规则，接受订阅的手机会收到一条网关离线恢复的短信。