

ICS 33.050

CCS M 30

团体标准

T/TAF 215—2024

应用于政企场景的物联网产品互通控制技术 要求

Technical requirements for interworking access control of Internet of
Things products applied to government and enterprise scenarios

2024-02-23 发布

2024-02-23 实施

电信终端产业协会 发布

目 次

前言	II
1 范围	1
2 规范性引用文件	1
3 术语和定义	1
4 缩略语	2
5 总体要求概述	2
6 互通控制方案	2
6.1 端云互通控制方案总体流程及网络架构	2
6.2 局域网内互通控制方案总体流程及网络架构	4
7 端云互通接入技术要求	4
7.1 WLAN 设备发现和配网	4
7.2 蓝牙设备发现和配网	7
7.3 Zigbee 设备发现和配网	10
7.4 有线设备发现和配网	16
7.5 身份认证流程和接口	16
7.6 接入流程和接口	18
8 局域网内接入技术要求	20
8.1 WLAN 设备发现和配网	20
8.2 蓝牙设备发现和配网	20
8.3 Zigbee 设备发现和配网	20
8.4 有线设备发现和配网	20
8.5 设备交互流程及接口	20
9 数据模型	23
10 设备控制技术要求	23
10.1 端云互通控制接口	23
10.2 局域网互通控制	25
10.3 设备上报属性或事件	26
10.4 设备属性获取	27
10.5 设备服务调用及属性设置	28
10.6 设备绑定与解绑	30
10.7 设备属性/事件	32

前 言

本文件按照GB/T 1.1—2020《标准化工作导则 第1部分：标准化文件的结构和起草规则》的规定起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由电信终端产业协会（TAF）提出并归口。

本文件起草单位：中国信息通信研究院、中国移动通信集团有限公司、中国移动通信集团终端有限公司、中国电信集团有限公司、联通数字科技有限公司、深圳高新兴瑞联科技有限公司。

本文件主要起草人：王焕丽、高宏、庞帅、陈昌文、陈江锋、陈思源、张璇、刘聪慧、张宇驰、吕传宇、邓兴、周兆熊、张鑫、卢坤、周祺、程卓、谭武功、邓华、崔芳、国辰、金洁、路鹏、陈逸泽、肖恩涛、高纪、崔凯、付国兴、王海兰、刘兆元。

应用于政企场景的物联网产品互通控制技术要求

1 范围

本文件规定了应用于政企场景中统一的设备发现、配网、接入、认证和控制的流程及技术要求。

本文件适用于应用于政企场景中的应用终端、控制类终端、APP、云平台等相关产品的互联互通软件开发。

2 规范性引用文件

本文件没有规范性引用文件。

3 术语和定义

下列术语和定义适用于本文件。

3.1

组网设备 networking equipment

是指用于建立、管理、运维和优化计算机网络的设备。在本文中，组网设备主要指路由器、智能网关等能够提供网络接入的设备。

3.2

全连接网关 fully connect gateway

一种可以实现网络协议转换的终端设备。该设备向下可以连接Zigbee或蓝牙设备，向上可以以有线或WLAN的方式连接到组网设备（智能网关或路由器等）上，从而实现Zigbee或蓝牙协议向以太网或WLAN协议的转换。

3.3

子设备 subset

通过WLAN或有线方式接到组网设备下面的终端设备，或是通过Zigbee或蓝牙方式接到全连接网关下面的终端设备。

3.4

智能网关 intelligent gateway

是指能够提供网络接入服务的一类设备。该设备具备周期性采集并保存子设备信息、存储并下发控制指令、转发数据报文等功能。

3.5

蜂窝类设备 cellular equipment

指的是使用4G、5G等移动网络进行通信的设备，如移动手机、NB-IoT设备等。在本文中，将蜂窝类设备归为组网设备的范畴。

4 缩略语

下列缩略语适用于本文件。

ASCII：美国信息交换标准代码（American Standard Code for Information Interchange）

CRC：循环冗余校验（Cyclical Redundancy Check）

JSON：JavaScript对象符号（JavaScript Object Notation）

MAC：媒介访问控制（Media Access Control）

MQTT：消息队列遥测传输协议（Message Queuing Telemetry Transport）

SN：序列号（Serial Number）

SSID：服务集标识（Service Set Identifier）

UDP：用户数据包协议（User Datagram Protocol）

UUID：通用唯一标识符（Universally Unique Identifier）

ZCL：Zigbee群集库（Zigbee Cluster Library）

5 总体要求概述

本文件典型的应用场景如图1所示。图中，A厂家的子设备（有线设备、WLAN设备、Zigbee设备、蓝牙设备等）可以通过本文件定义的内容接入A厂家的组网设备（智能网关、路由器等）上，也可以接入C厂家的组网设备（智能网关、路由器等）上；B厂家的子设备（有线设备、WLAN设备、Zigbee设备、蓝牙设备等）可以通过本文件定义的流程接入C厂家的组网设备上，也可以接入A厂家的组网设备上；A厂家的组网设备可以接入D厂家的云平台，也可以接入E厂家的云平台（图1中虚线所示）；C厂家的组网设备可以接入E厂家的云平台，也可以接入D厂家的云平台，从而实现跨厂家设备、跨云平台的互联互通。

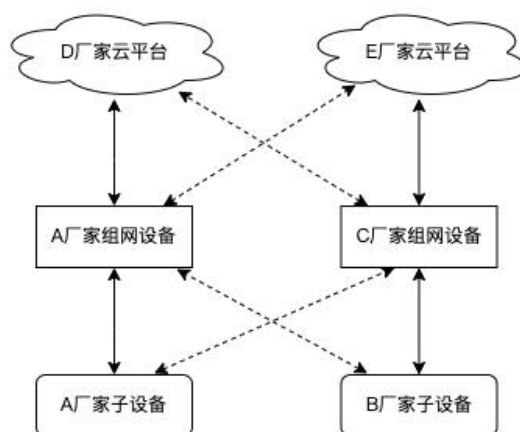


图1 典型应用场景

6 互通控制方案

6.1 端云互通控制方案总体流程及网络架构

端云互通控制方案的网络架构如图2所示。图中主要有2类设备：

- a) 组网设备，包括智能网关、路由器、全连接网关、蜂窝类设备等；
- b) 子设备，包括 WLAN 类子设备、有线类子设备、Zigbee 子设备、蓝牙（经典蓝牙、低功耗蓝牙、蓝牙 mesh 等）子设备等。

WLAN类子设备、有线类设备和全连接网关可以通过本文件定义的协议接入智能网关和路由器（如图2实线和虚线所示），Zigbee子设备和蓝牙子设备可以通过本文件定义的协议接入全连接网关（如图2实线和虚线所示）。智能网关和路由器可以与A厂家的云平台通信（如图2实线所示），也可以与B或N厂家的云平台通信（如图2虚线所示），A厂家的云平台也可以与B或N厂家的云平台进行数据同步和交互，从而实现跨厂商的互通控制。蜂窝类设备可以与A厂家的云平台通信（如图2实线所示），也可以与B或N厂家的云平台通信（如图2虚线所示）。同时，应用程序可以与A厂家的云平台通信（如图2实线所示），也可以与B或N厂家的云平台通信（如图2虚线所示），从而实现各类子设备的互通控制。

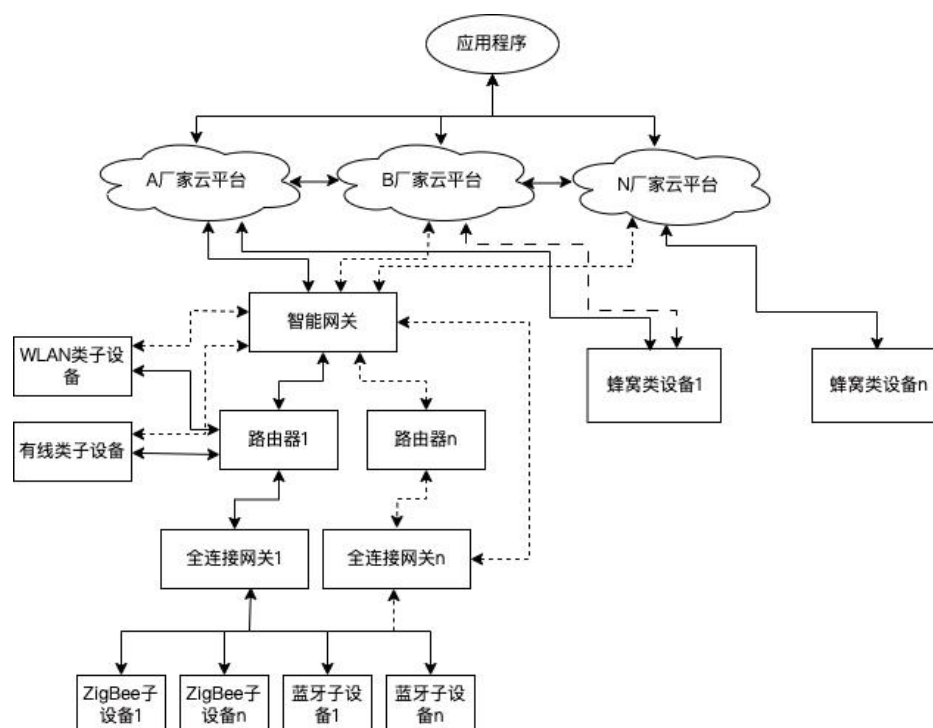


图2 端云互通控制网络架构

端云互通控制主要流程如下：

- a) 设备发现：终端层设备上电直至能够开始配网的过程，主要通过专用（隐藏）WLAN SSID、实体按键、设备定时器触发等方式来实现；
- b) 设备配网：终端层设备或全连接网关完成网络连接，接入智能网关或路由器的过程。主要是各类子设备（WLAN、有线、Zigbee、蓝牙等）的连接及组网，也包括了全连接网关接入智能网关或路由器的过程；
- c) 设备身份认证：各类设备在成功配网之后，与平台建立连接并完成设备认证及维持连接心跳的过程；
- d) 设备接入注册：设备通过身份认证之后，接入并注册到云平台；
- e) 设备互通控制：设备完成接入注册之后，通过应用程序或云平台对设备进行控制及状态查询。

注：在下文中，如无特殊说明，子设备即是指各类终端设备，如WLAN子设备和WLAN设备是指同一类设备。蜂窝类设备无需进行发现和配网，直接与云平台进行交互，交互接口与其它组网设备（智能网关、路由器等）完全一样。

6.2 局域网内互通控制方案总体流程及网络架构

局域网内互通控制方案的网络架构如图3所示。

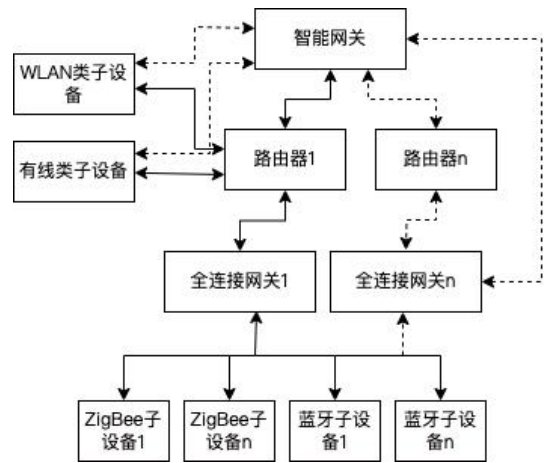


图3 局域网内互通控制方案网络架构

局域网内互通控制主要流程如下：

- a) 设备发现：终端层设备上电直至能够开始配网的过程，主要通过专用（隐藏）WLAN SSID、实体按键、设备定时器触发等方式来实现；
- b) 设备配网：终端层设备或全连接网关接入智能网关或路由器的过程。主要是各类终端设备（WLAN、有线、Zigbee、蓝牙等）的连接及组网，也包括了全连接网关接入智能网关或路由器的过程；
- c) 设备身份认证：各类设备在成功配网之后，在局域网内通过智能网关或路由器进行身份认证的过程；
- d) 设备互通控制：设备完成身份认证之后，通过智能网关下发互通联动规则，实现跨品牌设备之间的互通控制。具体实现方法是：当组网设备（如智能网关）发现云平台不在线之后，随即开启本地联动功能，周期性地采集并保存下挂子设备的信息；当下挂子设备 A 需要与下挂子设备 B 联动时，下挂子设备 A 将联动指令发送给组网设备（如智能网关），组网设备（如智能网关）解析该联动指令，并下发给下挂子设备 B（下挂子设备 A 和下挂子设备 B 可以不是同一个品牌的设备）。

7 端云互通接入技术要求

7.1 WLAN 设备发现和配网

WLAN设备需要经过发现和配网才能实现联网，并与云平台进行通信。WLAN设备发现和配网流程如图4所示。

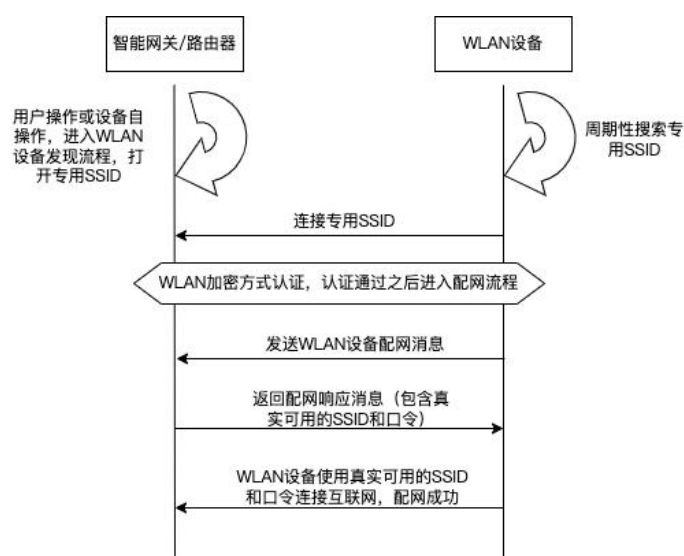


图4 WLAN设备发现和配网流程

详细的WLAN设备发现和配网流程如下：

- 用户操作或设备自操作，进入WLAN设备发现流程，打开专用SSID。用户可以通过操作智能网关或路由器上的实体按键打开专用SSID（推荐为SSID2且名称固定），智能网关或路由器也可以在自身的程序中加入定时器，周期性地开启专用SSID；当某台智能网关或路由器开启专用SSID之后，需要发消息给同网段的其他组网设备（包括智能网关、路由器或全连接网关），通知它们也进入WLAN设备发现流程，打开专用SSID；
- WLAN设备周期性地搜索专用SSID，可以上电之后随即开始搜索专用SSID，也可以在运行稳定之后再周期性地搜索专用SSID；
- WLAN设备搜索到专用SSID之后，便尝试连接该SSID，并进行加密方式的认证；
- WLAN加密方式认证通过之后，设备连上专用SSID，进入配网流程；
- WLAN设备发送配网消息（包含设备MAC、序列号等信息）给智能网关或路由器；
- 智能网关或路由器收到WLAN设备的配网消息之后，对消息内容进行校验，校验通过之后向WLAN设备返回配网响应消息（包含真实可用的SSID和口令等信息）；
- WLAN设备解析智能网关或路由器的响应消息，获取并使用真实的SSID和口令连接互联网，配网流程结束。

发现流程是指WLAN设备搜索到智能网关或路由器的专用SSID，并连接到该SSID的过程。对于智能网关或路由器而言，可以通过操作实体按键的方式打开专用SSID，也可以通过软件定时器的方式周期性地开启专用SSID；对于WLAN设备而言，可以在上电之后便开始搜索专用SSID，也可以在运行稳定之后再周期性地搜索专用SSID。为了便于被发现，建议专用SSID为SSID2且名称固定，例如：专用SSID名为WLAN-Link，口令为test123，加密方式为WPA2。

为了便于政企场景下同一个网段内的多个WLAN设备实现快速配网，当其中一台政企网关或路由器开启专用SSID之后，需要发送消息给该网段内的其他组网设备（包括智能网关、路由器或全连接网关等）同步开启专用SSID，各WLAN设备根据WLAN信号的强弱等自动连接到附件的组网设备上。

以一个实际的场景为例：某中小企业内有2台智能网关（分别叫做智能网关A、智能网关B），3台路由器（分别叫做路由器A、路由器B、路由器C），2台全连接网关（分别叫做全连接网关A、全连接网关B），有6台WLAN设备（分别叫做WLAN设备A、WLAN设备B、WLAN设备C、WLAN设备D、WLAN设备E、WLAN设备F），用户操作路由器B的按键进入WLAN设备发现流程，具体流程如下：

- a) 路由器 B 在用户操作了按键之后开启了专用 SSID，同时发送消息给该网段内的其他组网设备同步进入设备发现流程，即通知智能网关 A、智能网关 B、路由器 A、路由器 C、全连接网关 A、全连接网关 B 进入设备发现流程，开启专用 SSID。
- b) WLAN 设备 A、WLAN 设备 B、WLAN 设备 C 在上电之后就开始搜索专用 SSID，WLAN 设备 D、WLAN 设备 E、WLAN 设备 F 在运行稳定之后周期性地搜索专用 SSID。
- c) WLAN 设备 A 搜索到路由器 A 和路由器 B 的专用 SSID，发现路由器 A 的专用 SSID 信号更强，于是便尝试连接路由器 A 的专用 SSID，同理，WLAN 设备 B 尝试连接路由器 B 的专用 SSID、WLAN 设备 C 尝试连接路由器 C 的专用 SSID、WLAN 设备 D 尝试连接智能网关 A 的专用 SSID、WLAN 设备 E 尝试连接智能网关 B 的专用 SSID、WLAN 设备 F 尝试连接路由器 A 的专用 SSID。
- d) 各 WLAN 设备成功连接专用 SSID 之后，便进行加密方式的认证。
- e) WLAN 加密方式认证通过之后，各 WLAN 设备连上专用 SSID，设备发现流程结束。

前文提到的其中一台组网设备开启专用SSID之后，需要发送消息给该网段内的其他组网设备同步开启专用SSID，该过程可以采用广播方式发送UDP报文来实现，具体UDP端口推荐为8686，消息格式采用JSON，该广播消息无需回复。具体的消息字段参数定义如表1所示。

表1 UDP广播消息字段参数定义

参数名	描述	备注
ID	随机生成的 16 位 nonce 值	String 类型
RPCMethod	请求方法	填写固定 String 类型值，如“networking_notify”
version	消息版本号	整型值，如 1
deviceSn	组网设备 SN	String 类型
deviceMac	组网设备 MAC	String 类型
timestamp	当前时间戳，单位毫秒	长整型

消息样例如下：

```
{
  "ID": "xxxxxxxx",           // 随机生成 16 位 nonce 值
  "RPCMethod": "networking_notify", // 请求方法
  "version": 1,              // 消息版本号
  "deviceSn": "1122334455",   // 组网设备SN
  "deviceMac": "112233445566", // 组网设备MAC
  "timestamp": 1626948825804   // 当前时间戳
}
```

配网流程是指WLAN设备向智能网关或路由器发送配网信息，直至成功连接互联网的过程。同样以上述实际的场景为例，当WLAN设备A成功连接路由器A的专用SSID之后，配网的具体流程如下：

- a) WLAN 设备 A 通过 DHCP 协议获取到同一个网段内的 IP 地址（如 192.168.10.2），然后向路由器 A 的网关地址（如 192.168.10.1）发送 UDP 请求消息，该消息中携带 WLAN 设备 A 的 SN 和 MAC 地址等信息。
- b) 路由器 A 收到 WLAN 设备 A 的请求消息之后，对消息内容进行校验；如果校验成功，则向 WLAN 设备 A 返回包含真实可用的 SSID 和口令等内容的配网响应消息；如果校验失败，则向 WLAN 设备 A 返回配网失败的响应消息。

c) WLAN 设备 A 解析路由器 A 的响应消息,并使用真实可用的 SSID 和口令连接互联网,配网成功。
WLAN设备的请求消息及路由器的响应消息采用UDP协议,端口推荐为6688,消息格式采用JSON。具体的消息字段参数定义如表2所示。

表2 WLAN设备的请求消息字段参数定义

参数名	描述	备注
ID	随机生成的 16 位 nonce 值	String 类型
RPCMethod	请求方法	填写固定 String 类型值,如“boot”
deviceSn	WLAN 设备 SN	String 类型
secret_key	WLAN 设备密钥	String 类型,如 “011CB9249F81239A225D23CDA27EA24C”
deviceMac	WLAN 设备 MAC	String 类型
cmei	WLAN 设备 cmei	String 类型
deviceType	WLAN 设备类型编码	String 类型,如“10101”
firmwareVersion	WLAN 设备固件版本号	String 类型,如“V1.0.01”
softwareVersion	WLAN 设备软件版本号	String 类型,如“V1.0.02”

路由器的响应消息字段参数定义见表3:

表3 路由器的响应消息字段参数定义

参数名	描述	备注
ID	随机生成的 16 位 nonce 值	String 类型
respCode	响应码	int 类型,0-成功,其他-失败
respCont	响应描述	String 类型
SSID	真实可用的 WLAN SSID	String 类型,如“WLAN-1234”
Password	真实可用的 WLAN 口令	String 类型,如“12345678”
SecurityMode	WLAN 加密方式	String 类型,如“MIXED-WPAPSK2”
Encrypt	WLAN 加密算法	int 类型
SSIDAdvertisementEnabled	是否广播	int 类型,1-是,0-不是

相关说明如下:

- a) 表 2 和表 3 中的 ID 字段值一致,表示是对同一个消息的响应;
- b) SecurityMode 为认证加密模式,其取值为:None、WEP、WPA-Personal、WPA2-Personal、MIXED-WPAPSK2、WPA3-SAE、MIXED-WPA2WPA3;
- c) Encrypt 为加密算法,其取值为:0 (None)、1 (AES)、2 (TKIP)、3 (AES+TKIP)。

7.2 蓝牙设备发现和配网

发现流程是指全连接网关搜索到蓝牙子设备,并实现两者互联的过程。

蓝牙子设备上电之后,向全连接网关发送心跳(心跳包发送周期建议为5s)。收到全连接网关的心跳包回应后,双方即认为对方在正常工作,设备发现流程结束。

全连接网关侧也可依据此心跳定期检测蓝牙子设备是否正常工作。若蓝牙子设备无周期性心跳发送，则全连接网关可认为该子设备已离线。

在设备发现过程中，蓝牙子设备发送消息（心跳消息）接口字段和全连接网关响应消息字段定义如8.5节表22和表23所示。

蓝牙设备发现和配网包括发现和配网两个过程，其中设备发现是指蓝牙设备通过广播自己的设备名称和蓝牙地址，被其他蓝牙设备接收并识别，从而实现设备之间的无线连接和通信的过程；设备配网是指蓝牙设备之间建立安全连接，以实现设备之间的数据传输和共享的过程。

蓝牙设备需要完成在全连接网关上的发现和配网，如图5所示。

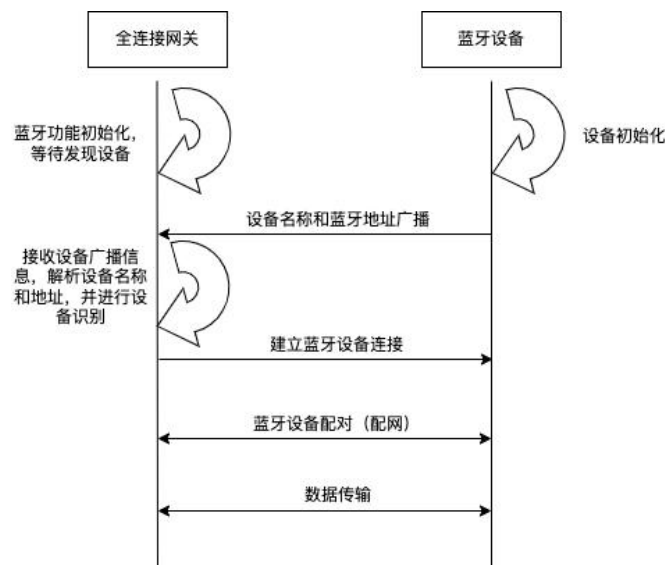


图5 蓝牙设备发现和配网流程

蓝牙设备在全连接网关上的发现和配网的具体流程如下：

- a) 设备初始化：全连接网关的蓝牙功能初始化，等待发现设备；蓝牙设备在开启后，先进行初始化，包括芯片复位、设置基频、建立链路等操作；
- b) 蓝牙设备名称和地址广播：蓝牙设备初始化完成后，将设备名称和蓝牙地址等信息封装到“设备发现广告”数据包中，并通过广播的方式发送出去；
- c) 全连接网关数据接收：全连接网关接收到这些广播后，解析出设备名称和蓝牙地址等信息，并进行设备识别；
- d) 设备连接：全连接网关识别到需要配对的设备之后，就与之建立连接，这个过程涉及到一些连接参数的设置，例如频段、速率、认证等；
- e) 设备配对（配网）：连接建立成功之后，全连接网关和蓝牙设备就可以进行设备的配对。这个过程通常会将设备的密钥、证书等安全信息进行交换和存储，以确保设备之间的安全连接和数据传输；
- f) 数据传输：配网完成后，蓝牙设备和全连接网关之间就可以进行数据传输。

配网流程是指全连接网关获取到蓝牙子设备信息，并实现两者具体的消息通信的过程。

蓝牙子设备配网流程包括获取子设备蓝牙模组信息、发送子设备蓝牙模组工作状态、全连接网关对子设备进行状态查询等。具体的接口字段定义如下所示：

- a) 获取子设备蓝牙模组信息

当全连接网关需要获取蓝牙子设备的蓝牙模组信息（如 pid、版本号等）时，即发送该消息。

全连接网关发送消息字段定义如表 4 所示。

表4 获取子设备蓝牙模组信息中全连接网关发送消息字段定义

序号	长度 (byte)	字段	说明
0	2	帧头	0x55
1	2	帧头	0xAA
2	1	版本号	0x00
3	1	命令字 CMD	0x01
4	2	数据长度	0x00
5	2	数据长度	0x00
6	1	CRC8	从帧头开始按字节求和得出的结果对 256 求余

蓝牙子设备的响应消息字段定义如表 5 所示。

表5 获取子设备蓝牙模组信息中蓝牙子设备的响应消息字段定义

序号	长度 (byte)	字段	说明
0	2	帧头	0x55
1	2	帧头	0xAA
2	1	版本号	0x00
3	1	命令字 CMD	0x01
4	2	数据长度	0x00
5	2	数据长度	0x0d
6~18	0x0d	Data	见表 6
19	1	CRC8	从帧头开始按字节求和得出的结果对 256 求余

Data 格式说明如表 6 所示。

表6 Data格式说明

1~8	9~13
PID	填充的是蓝牙模组软件版本号

响应消息示例：55 AA 00 01 00 0D 66 74 62 38 78 32 78 30 31 2E 30 2E 30 C0
(pid=ftb8x2x0, ver=1.0.0)。

- b) 发送子设备蓝牙模组工作状态
蓝牙子设备发送消息字段定义如表7所示。

表7 蓝牙子设备发送消息字段定义

序号	长度 (byte)	字段	说明
0	2	帧头	0x55
2	1	版本号	0x00

表 7 蓝牙子设备发送消息字段定义（续）

序号	长度（byte）	字段	说明
3	1	命令字 CMD	0x02
4	2	数据长度	0x00
5	2	数据长度	0x0d
6	1	Status	见表 8
7	1	CRC8	从帧头开始按字节求和得出的结果对 256 求余

全连接网关收到消息之后，无需响应，Status 格式说明如表 8 所示。

表8 Status格式说明

Status	工作状态说明	备注
0x00	状态正常	蓝牙模组工作正常
0x01	状态异常	蓝牙模组工作异常

- c) 全连接网关对子设备进行状态查询
- 状态查询为异步处理协议，主要用于全连接网关查询蓝牙子设备蓝牙模组的功能点状态。当子设备收到此帧时，通过CMD 0x02状态上报帧上报功能点状态。全连接网关发送消息字段定义如表9所示。

表9 全连接网关对子设备进行状态查询消息字段定义

序号	长度（byte）	字段	说明
0	2	帧头	0x55
1	2	帧头	0xAA
2	1	版本号	0x00
3	1	命令字 CMD	0x05
4	2	数据长度	0x00
5	2	数据长度	0x00
6	1	CRC8	从帧头开始按字节求和得出的结果对 256 求余

7.3 Zigbee 设备发现和配网

Zigbee设备发现和配网包括发现和配网两个过程，其中设备发现是指Zigbee设备上电初始化之后，通过信道扫描等方式发现网关，并发起入网请求的过程；设备配网是指网关响应Zigbee设备的入网请求，允许其接入网络，并进行数据传输和交互的过程。

Zigbee设备需要完成在全连接网关上的发现和配网，如图6所示。

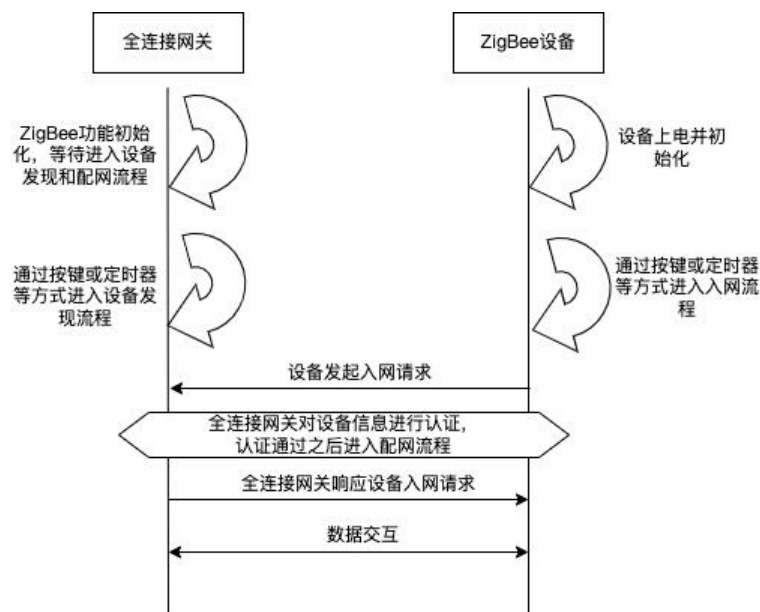


图6 Zigbee设备发现和配网流程

Zigbee设备在全连接网关上的发现和配网的具体流程如下：

- a) 设备初始化：
包括全连接网关和 Zigbee 设备的初始化，其中全连接网关进行功能初始化，等待进入设备发现和配网流程，Zigbee 设备在上电之后进行初始化；
- b) 进入设备发现和入网流程：
对于全连接网关来说，可以通过按键或定时器等方式进入设备发现 流程；对于 Zigbee 设备来说，可以通过按键或定时器等方式进入入网流程；
- c) 设备发起入网请求：
全连接网关和 Zigbee 设备初始化并通过案件或定时器等方式进入入网流程后，Zigbee 设备向全连接网关发起入网请求，Zigbee 设备通过信道扫描等方式发现全连接网关，并向全连接网关发起入网请求；
- d) 设备信息认证：
全连接网关收到设备的入网请求后会对发起入网请求的设备信息进行认证，认证通过之后进入配网流程；
- e) 全连接网关响应入网请求：
全连接网关对设备的入网请求进行响应，允许设备入网；
- f) 数据交互：
设备入网之后，和全连接网关之间进行数据交互，整个设备发现和配网流程结束。

发现流程是指在初始化完成之后，Zigbee设备通过信道扫描等方式发现网关，并发起入网请求的过程。

Zigbee设备的发现流程遵循标准的Zigbee协议，是由各Zigbee模组或芯片底层实现的，本文件不再详细描述。

设备配网流程是指全连接网关响应Zigbee设备的入网请求，允许其接入网络，并进行数据传输和交互的过程。

Zigbee设备的配网接口分为通用标准协议簇接口和扩展协议簇接口两种。其中，通用标准协议簇接口字段定义如表10所示。

表10 通用标准协议簇接口字段定义

Cluster (ID)	Attributes (ID)	Remarks
Basic (0x0000)	ZCL Version (0x0000)	版本号
	ApplicationVersion (0x0001)	固件版本号
	StackVersion (0x0002)	协议栈版本号
	HardwareVersion (0x0003)	硬件版本号
	ManufacturerName (0x0004)	厂商名, ASCII 码, 不超过 31 字节
	ModelIdentifier (0x0005)	设备型号, ASCII 码, 6 字节
	DateCode (0x0006)	固件发布日期, 不超过 16 字节
	PowerSource (0x0007)	遵循 ZCL 定义, 不应为 0x00
	SerialNumber (0x000d)	序列号
	DisableLocalConfig (0x0014)	禁用设备配置功能
	SwBuildId (0x4000)	禁用本地配置属性

对于扩展协议簇接口（即设备扩展定义），针对不同的设备有不同的字段定义。以下以窗帘电机、智能插座、智能开关的扩展协议簇接口进行说明。
窗帘电机的扩展协议簇接口字段定义见表11。

表11 窗帘电机扩展协议簇接口字段定义

Cluster (ID)	Attributes (ID)	DefaultValue	Commands (ID) /Remarks
Identify (0x0003)	IdentifyTime (0x0000)	0x0000	(I) Identify (0x00)
Scenes (0x0005)	SceneCount (0x0000)	0x00	现有场景累积总数 (I) AddScene (0x00) (I) ViewScene (0x01) (I) RemoveScene (0x02) (I) RemoveAllScenes (0x03) (I) StoreScene (0x04) (I) RecallScene (0x05) (I) GetScene Membership (0x06) cmd 列表是对场景操作, 增删改查
	CurrentScene (0x0001)	0x00	当前场景属性保存上次调用的场景的场景 ID。

表11 窗帘电机扩展协议簇接口字段定义（续）

Cluster (ID)	Attributes (ID)	DefaultValue	Commands (ID) /Remarks
Scenes (0x0005)	CurrentGroup (0x0002)	0x0000	保存上次调用的场景的组 ID
	SceneValid (0x0003)	0x00	设备的状态是否与 CurrentScene 和 CurrentGroup 属性关联。TRUE 表示这些属性有效，FALSE 表示它们是无效的。
	NameSupport (0x0004)	0x00	最高有效位指示是否支持场景名称
Window Covering (0x0102)	WindowCovering Type (0x0000)	0x04	窗帘类型
	ConfigStatus (0x0007)	0x03	配置状态
	CurrentPosition Lift Percentage (0x0008)	0x02	当前位置提升百分比
	Mode (0x0017)	0x00	模式属性允许配置
	CurtainState (0x8002)	0x00	Up/Open (0x00) Down/Close (0x01) Stop (0x02) 上报当前状态： 1-open; 2-close; 0-stop
	MotorType (0x8000)	0x02	电机类型
	BackLightMode (0x8001)	0x01	背光灯模式：0-关闭；1-开

智能插座的扩展协议簇接口字段定义见表12。

表12 智能插座扩展协议簇接口字段定义

Cluster (ID)	Attributes (ID)	DefaultValue	Commands (ID) /Remarks
Identify (0x0003)	IdentifyTime (0x0000)	0x0000	(I) Identify (0x00)
Scenes (0x0005)	SceneCount (0x0000)	0x00	现有场景累积总数 (I) AddScene (0x00) (I) ViewScene (0x01) (I) RemoveScene (0x02) (I) RemoveAllScenes (0x03) (I) StoreScene (0x04) (I) RecallScene (0x05) (I) GetScene Membership (0x06) cmd 列表是对场景操作，增删改查
	CurrentScene (0x0001)	0x00	当前场景属性保存上次调用的场景的场景 ID。

表12 智能插座扩展协议簇接口字段定义（续）

Cluster (ID)	Attributes (ID)	DefaultValue	Commands (ID) /Remarks
Scenes (0x0005)	CurrentGroup (0x0002)	0x0000	保存上次调用的场景的组 ID
	SceneValid (0x0003)	0x00	设备的状态是否与 CurrentScene 和 CurrentGroup 属性关联。TRUE 表示这些属性有效，FALSE 表示它们是无效的。
	NameSupport (0x0004)	0x00	最高有效位指示是否支持场景名称
OnOff (0x0006)	On/Off (0x0000)	0x00	(I) Off (0x00) (I) On (0x01) (I) Toggle (0x02)
	globalSceneContrl (0x4000)	0x0000	为了支持用户返回设备的最后设置的用例，引入全局场景，该全局场景在设备关闭时存储设备已打开
	on_time (0x4001)	0x0000	定时开启
	Off Wait Time (0x4002)	0x0000	定时关
	Start Up on/off (0x4003)	0x00	定义设备所需的启动行为
Electrical Measurement (0x0B04)	MeasurementType (0x0000)	0x00000009	测量方式：A 相有功测量
	Voltage (0x0505)	0xffff	电压
	Current (0x0508)	0xffff	电流
	ActivePower (0x050B)	0xffff	有功功率
	PowerFactor (0x0510)	0x00	功率因数
	Voltage Mult (0x0600)	0x01	电压乘数
	VoltageDiv (0x0601)	0x01	电压除数 17
	CurrentMult (0x0602)	0x01	电流乘数
	CurrentDiv (0x0603)	1000	电压除数
	ActivePowerMult (0x0604)	0x01	有功功率乘数
	ActivePowerDiv (0x0605)	0x01	有功功率除数
	ACAlarmsMask (0x0800)	0x0004	AC 警报
	VoltageOverload (0x0801)	275	过载电压阈值
	CurrentOverload (0x0802)	19200	过载电流阈值
	ACActivePower Overload (0x0803)	4224	过载 AC 有功功率阈值
	OverloadProtect Enable (0x8000)	0x01	过载保护使能
Simple Metering (0x0702)	CurrentSummation Delivered (0x0000)	NULL	耗电量
	Status (0x0200)	0x00	状态
	UnitofMeasure (0x0300)	0x00	计量单位

表12 智能插座扩展协议簇接口字段定义（续）

Cluster (ID)	Attributes (ID)	DefaultValue	Commands (ID) /Remarks
Simple Metering (0x0702)	Multiplier(0x0301)	0x000001	乘数
	Divisor(0x0302)	0x0003E8	除数
	Summation Formatting (0x0303)	0xB2	求和格式化
	MeteringDevice Type (0x0306)	0	计量设备类型
	PowerConsumption Clear (0x8000)	0x00	历史电量清除
	Voltage Coefficient (0x8001)	0x3e8	电压系数
	CurrentCoefficient (0x8002)	0x3e8	电流系数
	PowerCoefficient (0x8003)	0x3e8	功率系数
	PowerWasteCoefficient (0x8004)	0x3e8	功耗系数

智能开关的扩展协议簇接口字段定义见表13。

表13 智能开关扩展协议簇接口字段定义

Cluster (ID)	Attributes (ID)	DefaultValue	Commands (ID) /Remarks
Identify (0x0003)	IdentifyTime (0x0000)	0x0000	(I) Identify (0x00)
Scenes (0x0005)	SceneCount (0x0000)	0x00	现有场景累积总数 (I) AddScene (0x00) (I) ViewScene (0x01) (I) RemoveScene (0x02) (I) RemoveAllScenes (0x03) (I) StoreScene (0x04) (I) RecallScene (0x05) (I) GetScene Membership (0x06) cmd 列表是对场景操作, 增删改查
	CurrentScene (0x0001)	0x00	当前场景属性保存上次调用的场景的场景 ID
	CurrentGroup (0x0002)	0x0000	保存上次调用的场景的组 ID
Scenes (0x0005)	SceneValid (0x0003)	0x00	设备的状态是否与 CurrentScene 和 CurrentGroup 属性关联。TRUE 表示这些属性有效, FALSE 表示它们是无效的

表13 智能开关扩展协议簇接口字段定义（续）

Cluster (ID)	Attributes (ID)	DefaultValue	Commands (ID) /Remarks
Scenes (0x0005)	NameSupport (0x0004)	0x00	最高有效位指示是否支持场景名称
OnOff (0x0006)	On/Off (0x0000)	0x00	(I) Off (0x00) (I) On (0x01) (I) Toggle (0x02)
	globalSceneContrl (0x4000)	0x0000	为了支持用户返回设备的最后设置的用例，引入全局场景，该全局场景在设备关闭时存储设备已打开
	on_time (0x4001)	0x0000	定时开启
	Off Wait Time (0x4002)	0x0000	定时关
	Start Up on/off (0x4003)	0x00	定义设备所需的启动行为

对于其它各种类型的Zigbee设备，只要符合Zigbee协议的接口字段定义格式，均可以接入全连接网关。

7.4 有线设备发现和配网

有线设备只需要使用标准格式的网线连接到组网设备（智能网关或路由器等）上，即可完成设备的发现和配网。因此，针对有线设备，无需专门开发设备发现和配网接口，只要正常连接到组网设备上即可成功联网。

7.5 身份认证流程和接口

产品、设备创建时，云平台为每类产品、每个设备分配唯一的 key，设备登录时需要使用通过 key 计算出的鉴权 token 来进行安全认证。在设备与平台交互（包括首次上线、数据报文上报、下发等）的报文中需要设备身份携鉴权参数，具体涉及的参数及算法如下：

- a) 鉴权参数：
设备鉴权参数（鉴权三元组）见表 14。

表14 鉴权三元组

参数	描述
ProductId	产品 ID，平台生成唯一 ID
DeviceName	设备名称
Token	鉴权 token

- b) Token 算法：
Token 由多个参数构成，如表 15 所示。

表15 Token参数

名称	类型	是否必须	参数说明	参数示例
version	string	是	参数组版本号，日期格式，目前仅支持“2018-10-31”	2018-10-31
res	string	是	访问资源 resource 格式为： products/{产品id}/devices/{设备名字}	products/dafdfadfafdaf /devices/chel
et	int	是	访问过期时间 expirationTime，单位秒，unix 时间当一次访问参数中的 et 时间小于当前时间时，平台会认为访问参数过期从而拒绝该访问	1537255523 表示：北京时 间 2018-09-18 15:25:23
method	string	是	签名方法 signatureMethod 支持 hmacmd5、hmacsha1、hmacsha256	sha1（代表使用 hmacsha1 算法）
sign	string	是	签名结果字符串 signature	rBYeJXTp2q4V3C2aj4DBzj aydcw%3D

c) 参数编码
Token 中 key=value 的形式的 value 部分需要经过 URL 编码,需要进行编码的特殊符号见表 16。

表16 参数编码

序号	特殊符号	URL 编码
1	+	%2B
2	空格	%20
3	/	%2F
4	?	%3F
5	%	%25
6	#	%23
7	&	%26
8	=	%3D

编码后，上例中实际传输 Token 为 version=2018-10-31&res=products%2F123123%2Fdevices%2Fchel&et=1537255523&method=sha1&sign=ZjA1NzZlMmMxYzIOTg3MjBzNjYTI2MjA4Yw%3D。

设备的鉴权算法涉及的产品key需要在云平台定义产品信息后由云平台生成，同时需要定义该设备的数据模型（属性、事件、服务），再根据7.5.1中的鉴权算法计算token，实现与云平台的交互。具体流程如图7所示。

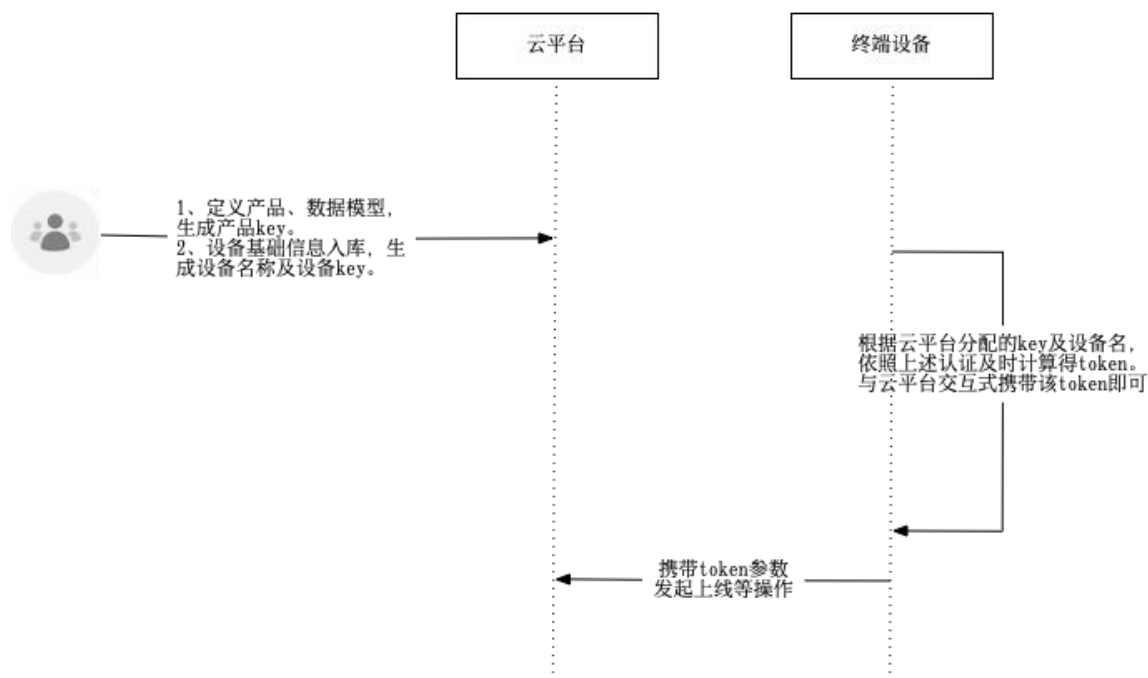


图7 认证流程和接口

支持 MQTT 方式的设备接入接口及参数如下：

- a) 服务地址：broker ip；
- b) 服务端口：broker port；
- c) 客户端 id：设备名称；
- d) 用户名：产品 id；
- e) 口令：token。

7.6 接入流程和接口

设备与云平台建立连接后，会进行登录、心跳检查、登出等操作。具体流程图见图8。

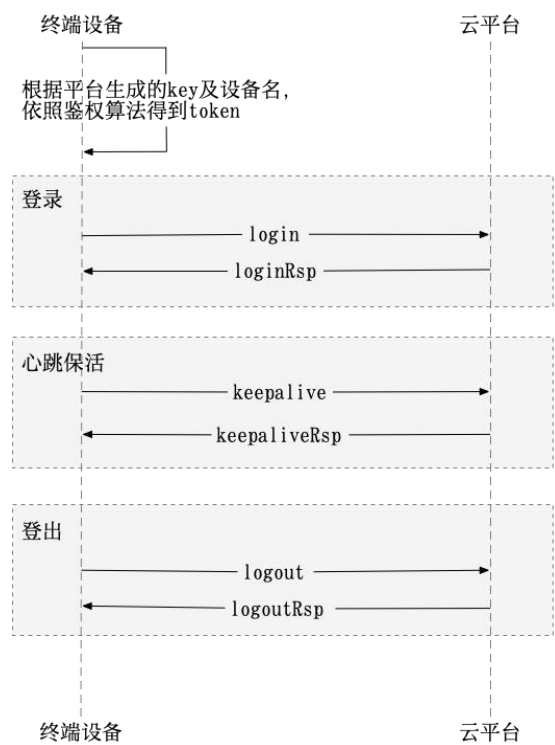


图8 设备接入流程

设备与云平台建立连接后的接入流程如下：

- a) key 由平台通过 uuid 算法生成全局唯一 128 位字符，参照 7.5 节鉴权算法得到 token 参数；
- b) 登录过程为设备上电，由设备发起，携带设备标识、license 和 token 信息连接至 MQTT Broker；当为非直连设备，比如 Zigbee 子设备、蓝牙 mesh 子设备，该流程由其网关代理该类设备发起登录请求；
- c) 心跳保活流程为 MQTT 的 keep Alive 机制，设备在创建和 MQTT Broker 的连接时，只要将连接请求协议包内的 Keep Alive 可变头部字段设置为非 0 值，就可以在通信双方间启用 Keep Alive 机制。Keep Alive 为 0~65535 的一个整数，代表客户端发送两次 MQTT 协议包之间的最大间隔时间；
- d) 登出流程为设备终端主动断电时，由设备端发起，携带相应的设备 id 及 license 信息；当为非直连设备，比如 Zigbee 子设备、蓝牙 mesh 子设备，该流程由其网关代理该类设备发起登出请求。

按设备接入流程中各接口和参数定义如下：

- a) 设备登录接口，直连设备连接 MQTT Broker 即完成登录，子设备通过网关代理登录，接口定义见表 17；

表17 设备登录接口

接口描述	Topic 类	操作权限
请求	<code>\$sys/{pid}/{device-name}/thing/sub/login</code>	发布
响应	<code>\$sys/{pid}/{device-name}/thing/sub/login/reply</code>	订阅

- b) 设备心跳保活依赖 MQTT 的 keepalive 机制;
- c) 设备登出接口中直连设备断连 MQTT Broker 即完成登出, 子设备通过网关代理登出, 接口定义见表 18。

表18 设备登出接口

接口描述	Topic 类	操作权限
请求	<code>\$sys/{pid}/{device-name}/thing/sub/logout</code>	发布
响应	<code>\$sys/{pid}/{device-name}/thing/sub/logout/reply</code>	订阅

8 局域网内接入技术要求

8.1 WLAN 设备发现和配网

参考本文件第7.1节。

8.2 蓝牙设备发现和配网

参考本文件第7.2节。

8.3 Zigbee 设备发现和配网

参考本文件第7.3节。

8.4 有线设备发现和配网

有线设备只需要使用标准格式的网线连接到组网设备（智能网关或路由器等）上，即设备的发现和配网。因此，针对有线设备，无需专门开发设备发现和配网接口，只要正常连接到组网设备上即可成功联网。

8.5 设备交互流程及接口

在局域网内，当子设备接入到组网设备之后，需要定期与组网设备进行心跳消息交互，以了解当前局域网内有哪些设备，便于进行互通控制，局域网内设备间的心跳交互见图9。

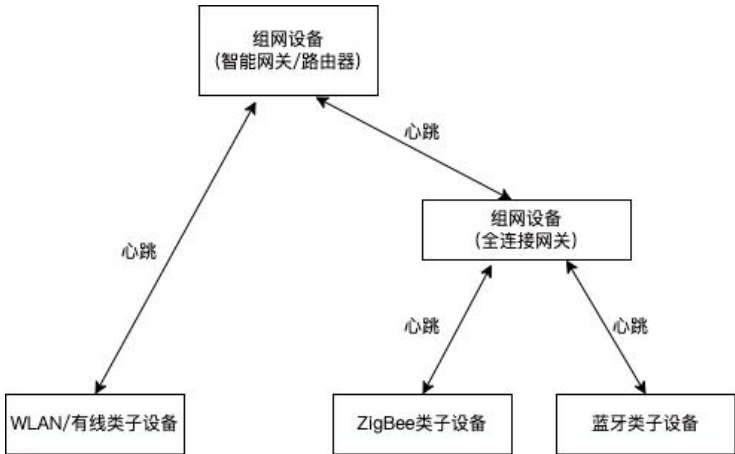


图9 局域网内设备间的心跳交互示意图

从图9可以看出，WLAN/有线类/蜂窝类子设备、全连接网关直接与智能网关/路由器进行心跳消息交互，Zigbee类和蓝牙类子设备之间与全连接网关进行心跳消息交互。其中，WLAN/有线类/蜂窝类子设备、全连接网关向智能网关/路由器发送的心跳消息参数定义如表19所示。

表19 向智能网关/路由器发送的心跳消息参数定义

参数名	描述	备注
ID	随机生成的 16 位 nonce 值	String 类型
RPCMethod	请求方法	填写固定 String 类型值，如“heart_beat”
version	消息版本号	整型值，如 1
deviceSn	设备 SN	String 类型
deviceMac	设备 MAC	String 类型
timestamp	当前时间戳，单位毫秒	长整型

智能网关/路由器针对该心跳消息返回的消息参数定义如表20所示。

表20 智能网关/路由器返回的消息参数定义

参数名	描述	备注
ID	随机生成的 16 位 nonce 值	String 类型
respCode	响应码	int 类型，0-成功，其他-失败
respCont	响应描述	String 类型
subDevInfo	下挂设备信息	String 类型，多个设备信息用分号分割，如“WLANDev123;EthDev456”

蓝牙子设备向全连接网关发送的心跳消息参数定义如表21所示。

表21 蓝牙子设备发送的心跳消息参数定义

序号	长度 (byte)	字段	说明
0	2	帧头	0x55
1	2	帧头	0xAA
2	1	版本号	0x00
3	1	命令字 CMD	0x00
4	2	数据长度	0x00
5	2	数据长度	0x00
6	1	CRC8	从帧头开始按字节求和得出的结果对 256 求余

全连接网关针对该心跳消息返回的消息参数定义如表 22 所示。

表22 全连接网关返回的消息参数定义

序号	长度 (byte)	字段	说明
0	2	帧头	0x55

表 22 全连接网关返回的消息参数定义（续）

序号	长度（byte）	字段	说明
1	2	帧头	0xAA
2	1	版本号	0x00
3	1	命令字 CMD	0x00
4	2	数据长度	0x00
5	2	数据长度	0x00
6	1	Status	见下表
7	256	已接入的子设备类型	有则返回设备类型，无则返回空
8	1	CRC8	从帧头开始按字节求和得出的结果对 256 求余

Status 返回值说明如表 23 所示。

表23 Status返回值说明

Status 返回值	说明
0x00	仅全连接网关第一次收到子设备的心跳包后的返回值，全连接网关也可依据这一点判断工作过程中子设备是否重启
0x01	除子设备重启后第一次返回 0x00 外，其余均返回 0x01

Zigbee子设备向全连接网关发送的心跳消息参数定义如表24所示。

表24 Zigbee子设备发送的心跳消息参数定义

Cluster (ID)	Attributes (ID)	DefaultValue	Commands (ID) /Remarks
Identify (0x0003)	IdentifyTime (0x0000)	0x0000	(I) Identify (0x00)
Heart Beat (0x0500)	Message Type (0x0001)	0x01	心跳类型： 0x01-周期心跳 0x02-一次性心跳
	Message Period (0x0007)	0x02	心跳周期，默认 2 秒

全连接网关针对该心跳消息返回的消息参数定义如表 25 所示。

表25 全连接网关返回的消息参数定义

Cluster (ID)	Attributes (ID)	DefaultValue	Commands (ID) /Remarks
Identify (0x0003)	IdentifyTime (0x0000)	0x0000	(I) Identify (0x00)
Heart Beat (0x0500)	Message Type (0x0001)	0x01	心跳类型： 0x01-周期心跳 0x02-一次性心跳
	subDevInfo (0x0007)	—	已接入的子设备类型，有则返回具体值，无则返回空

9 数据模型

数据模型的格式是 JSON，用于设备端和物联网平台的双向通信，更便捷地实现和规范了设备端和物联网平台之间的业务数据交互。物模型基础功能分为三类：属性、服务、事件。物模型基础功能说明见表 26。

表26 物模型基础功能

功能类型	说明
属性	用于描述设备的动态特征，包括运行时的状态，应用可发起对属性的读取和设置请求。
服务	用于描述终端设备可被外部调用的能力，可设置输入参数和输出参数。服务可实现复杂的业务逻辑，例如执行某项特定的任务；支持同步或异步返回结果。
事件	设备运行时可以被触发的上行消息，如设备运行的记录信息，设备异常时发出的告警、故障信息等；可包含多个输出参数。

数据类型支持整数型、浮点型、时间类型、布尔型、字符型、枚举型、位图型、数组类型、结构图等，具体支持的数据类型格式见表 27。

表27 数据类型支持

类型	标识符	说明
整数型	int32、int64	整数、长整数
浮点型	float、double	单精度浮点、双精度浮点
时间类型	date	长整数的扩展类型，整数类型 int64 的 UTC 时间戳（毫秒）
布尔型	bool	true 或 false
字符型	string	字符串，文本类型
枚举型	enum	枚举类型，枚举值为整数
位图型	bitMap	位图，多用于多个故障信号同时上送，非传统意义的图片数据
数组类型	array	数组类型，元素类型支持：int32、int64、float、double、string、date、struct
结构图	struct	结构图类型，仅支持一层嵌套，成员类型不支持数组

10 设备控制技术要求

10.1 端云互通控制接口

平台中，服务端和设备端通过通信主题topic实现消息通信，设备可以通过发布消息到系统 topic 调用服务接口，也可以订阅系统 topic 用于接收服务消息通知，服务提供的系统topic见表28。

表28 topic列表

功能	类别	行为	描述	topic 类	操作权限
物模型通信 topic	属性	设备属性上报	请求	\$sys/{pid}/{device-name}/thing/property/post	发布
			响应	\$sys/{pid}/{device-name}/thing/property/post/reply	订阅
		设备属性设置	请求	\$sys/{pid}/{device-name}/thing/property/set	订阅
			响应	\$sys/{pid}/{device-name}/thing/property/set_reply	发布
		设备获取属性期望值	请求	\$sys/{pid}/{device-name}/thing/property/desired/get	发布
			响应	\$sys/{pid}/{device-name}/thing/property/desired/get/reply	订阅
		清除属性期望值	请求	\$sys/{pid}/{device-name}/thing/property/desired/delete	发布
			响应	\$sys/{pid}/{device-name}/thing/property/desired/delete/reply	订阅
		设备属性获取	请求	\$sys/{pid}/{device-name}/thing/property/get	订阅
			响应	\$sys/{pid}/{device-name}/thing/property/get_reply	发布
	事件	设备事件上报	请求	\$sys/{pid}/{device-name}/thing/event/post	发布
			响应	\$sys/{pid}/{device-name}/thing/event/post/reply	订阅
	服务	设备服务调用	请求	\$sys/{pid}/{device-name}/thing/service/{identifier}/invoke	订阅
			响应	\$sys/{pid}/{device-name}/thing/service/{identifier}/invoke_reply	发布
网关与子设备通信 topic	上下线	子设备上线	请求	\$sys/{pid}/{device-name}/thing/sub/login	发布
			响应	\$sys/{pid}/{device-name}/thing/sub/login/reply	订阅
		子设备下线	请求	\$sys/{pid}/{device-name}/thing/sub/logout	发布
			响应	\$sys/{pid}/{device-name}/thing/sub/logout/reply	订阅
	属性和事件	批量上报属性和事件（网关上报或代理子设备上报）	请求	\$sys/{pid}/{device-name}/thing/pack/post	发布
			响应	\$sys/{pid}/{device-name}/thing/pack/post/reply	订阅

表28 topic列表（续）

功能	类别	行为	描述	topic 类	操作权限
网关与子设备通信 topic	属性和事件	子设备属性获取	请求	\$sys/{pid}/{device-name}/thing/sub/property/get	订阅
			响应	\$sys/{pid}/{device-name}/thing/sub/property/get_reply	发布
		历史属性和事件上报（网关上报或代理子设备上报）	请求	\$sys/{pid}/{device-name}/thing/history/post	发布
			响应	\$sys/{pid}/{device-name}/thing/history/post/reply	订阅
	服务	子设备服务调用	请求	\$sys/{pid}/{device-name}/thing/sub/service/invoke	订阅
			响应	\$sys/{pid}/{device-name}/thing/sub/service/invoke_reply	发布
	拓扑关系	添加子设备	请求	\$sys/{pid}/{device-name}/thing/sub/topo/add	发布
			响应	\$sys/{pid}/{device-name}/thing/sub/topo/add/reply	订阅
		删除子设备	请求	\$sys/{pid}/{device-name}/thing/sub/topo/delete	发布
			响应	\$sys/{pid}/{device-name}/thing/sub/topo/delete/reply	订阅
		获取拓扑关系	请求	\$sys/{pid}/{device-name}/thing/sub/topo/get	发布
			响应	\$sys/{pid}/{device-name}/thing/sub/topo/get/reply	订阅
			网关同步结果响应	\$sys/{pid}/{device-name}/thing/sub/topo/get/result	发布
		通知网关拓扑关系变化	请求	\$sys/{pid}/{device-name}/thing/sub/topo/change	订阅
			响应	\$sys/{pid}/{device-name}/thing/sub/topo/change_reply	发布

10.2 局域网互通控制

在局域网内,组网设备(如智能网关)可以周期性地采集并保存下挂子设备的信息,当子设备1(WLAN/有线/Zigbee/蓝牙子设备等)和子设备2(WLAN/有线/Zigbee/蓝牙子设备等)之间进行互通控制时,对应的流程如图10所示。

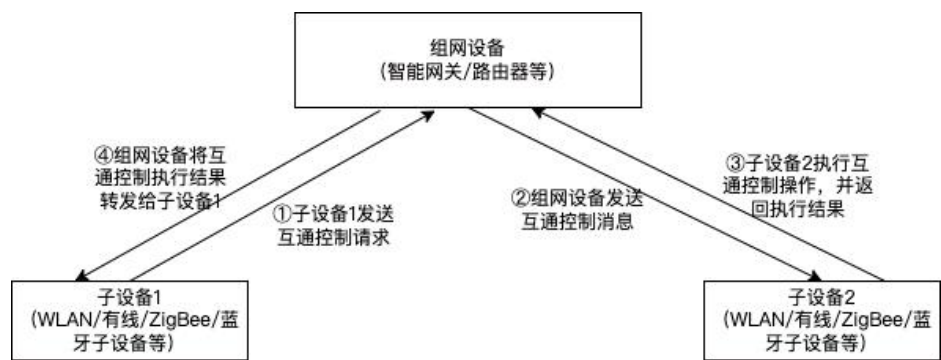


图10 局域网内子设备间的互通控制流程

局域网内的子设备间的互通控制流程如下：

- 子设备 1 通过 8.5 节的心跳接口发现局域网内的设备，当子设备 1（WLAN/有线/Zigbee/蓝牙子设备等）需要与子设备 2（WLAN/有线/Zigbee/蓝牙子设备等）进行互通控制时，子设备 1 将相关的互通控制请求发送给组网设备。如 WLAN 子设备 1 想要和 WLAN 子设备 2 进行互通控制，那么 WLAN 子设备 1 就将互通控制请求发送给智能网关或路由器；
- 组网设备接收子设备 1 的互通控制请求，解析出具体的操作请求类型，然后与第 10.1 节表 24 中的“网关与子设备通信 topic”进行匹配，找到具体的子设备操作指令，并组装成互通控制消息发送给子设备 2。如子设备 1 想查看子设备 2 的在线情况，则组网设备匹配到表 24 中“网关与子设备通信 topic”的“上下线”消息类别，并将对应的指令发送给子设备 2；
- 子设备 2 接收到组网设备发送的互通控制消息，执行对应的互通控制操作，并将操作执行结果返回给组网设备。如子设备 2 解析到子设备 1 想查看自己的在线情况，则将自己的状态消息封装成执行结果消息，并返回给组网设备。如果子设备 2 不支持该操作，则直接返回不支持；
- 组网设备将子设备 2 的响应消息封装成互通控制执行结果消息转发给子设备 1，子设备 1 即完成了与子设备 2 的互通控制。

上述流程是针对子设备间互通控制的情况，如果组网设备想要和子设备进行互通控制，则直接发送消息给对于子设备即可，控制接口参考第10.1节表24中的“网关与子设备通信topic”。

10.3 设备上报属性或事件

设备上报属性或事件流程见图11，具体步骤如下：

- 设备使用 JSON 指定主题，上报数据；
- 平台对数据进行业务处理，包含格式验证、数据存储等，如果配置了规则引擎，数据将流转到用户配置的消息目的地，推送方式支持 HTTP(s)、MQ 等；
- 平台返回数据上报结果；
- 用户也可以通过控制台或公开 API 查询上报的数据。

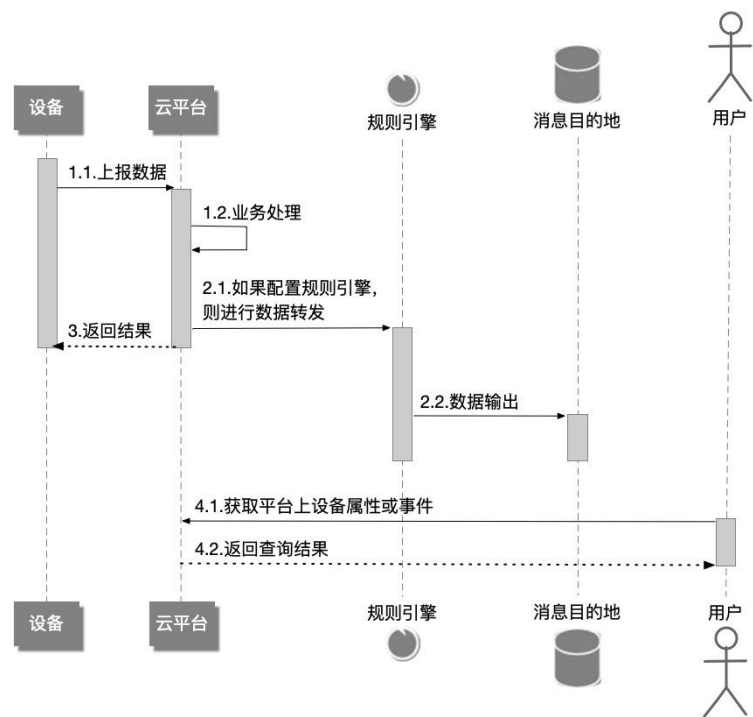


图11 设备上报属性或事件流程

10.4 设备属性获取

- 设备上报属性或事件流程见图12，具体步骤如下：
- a) 用户通过公开 API 或控制台发起获取属性请求；
 - b) 平台将根据物模型定义验证请求参数；
 - c) 平台将数据请求发给设备；
 - d) 平台等待设备响应，如果等待超时，将返回超时错误信息；
 - e) 设备处理完请求之后，把需要的设备属性返回给平台；
 - f) 平台收到设备最新属性后，对设备属性进行校验，将结果返回给用户。

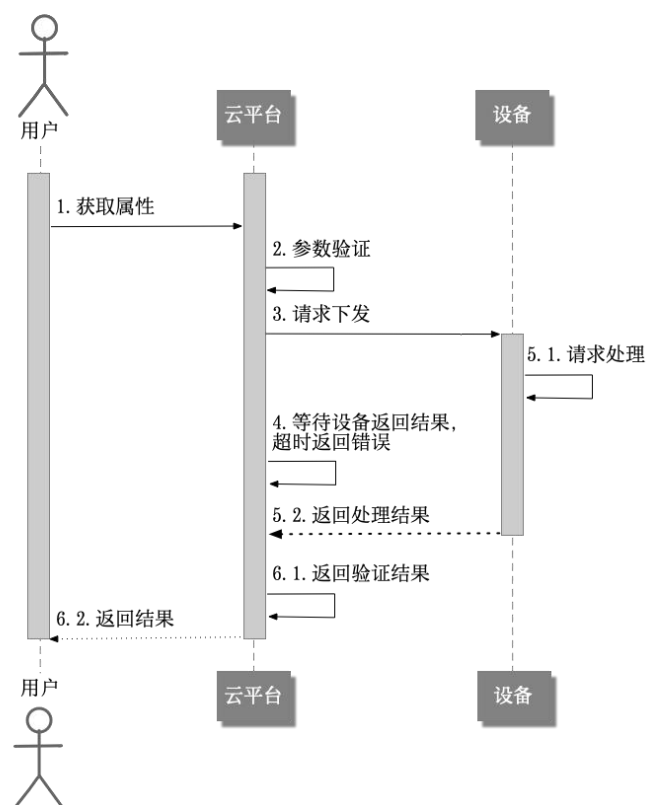


图12 设备属性获取流程

10.5 设备服务调用及属性设置

同步服务调用流程图见图 13，具体步骤如下：

- a) 通过调用对应 API 接口来调用同步服务（定义服务时，调用方式选择为同步的服务即为同步调用）；
- b) 平台将根据物模型定义验证输入参数；
- c) 平台将数据下发给设备；
- d) 平台等待设备响应，如果等待超时，将返回相应错误信息；
- e) 设备处理完数据之后，把处理结果返回给平台；
- f) 平台收到设备处理结果后，对设备输出参数进行校验，将结果返回给用户。

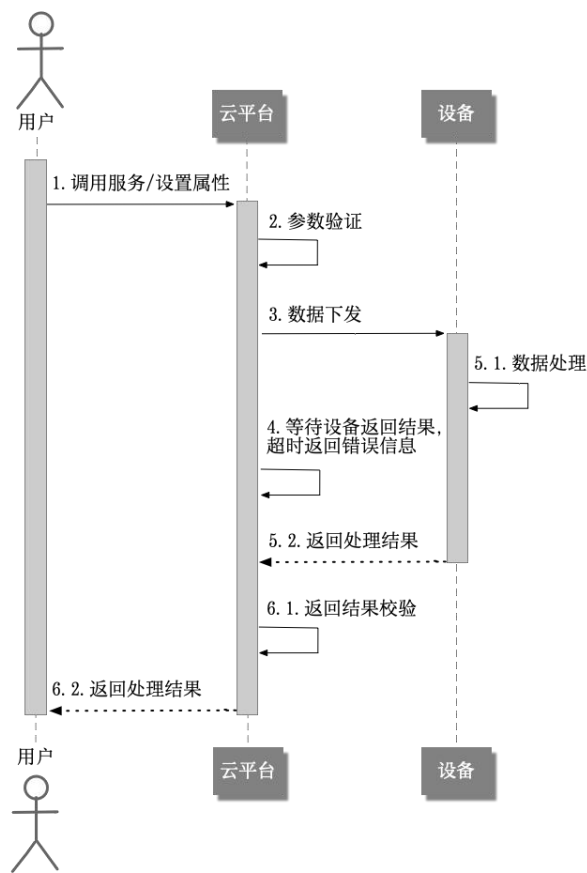


图13 同步服务调用流程

异步服务调用流程图见图 14，具体步骤如下：

- a) 通过调用对应 API 接口来调用同步服务（定义服务时，调用方式选择为异步的服务即为异步调用）；
- b) 平台对输出参数进行校验，并返回处理结果给用户；
- c) 平台采用异步方式将数据下发；
- d) 设备收到数据后，进行业务处理；
- e) 设备完成业务处理后，返回处理结果给平台；
- f) 平台对设备返回的输出参数进行验证；
- g) 如果配置了规则引擎，数据将流转 to 用户配置的消息目的地，推送方式支持 HTTPS、MQ 等。

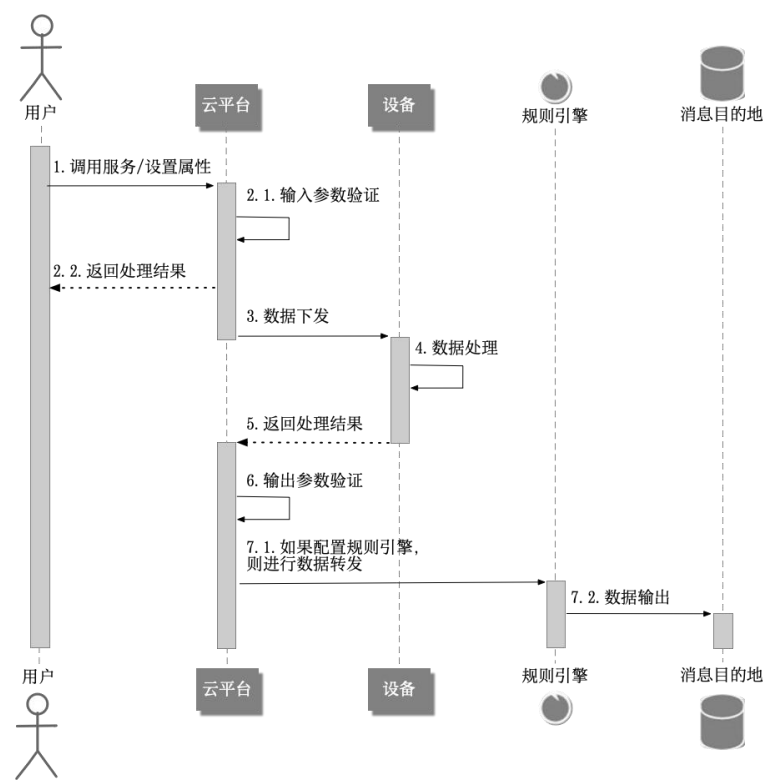


图14 异步服务调用流程

10.6 设备绑定与解绑

用户将设备与平台绑定或解绑后，云平台下发设备拓扑关系至网关，网关同步拓扑关系并返回消息给云平台，云平台下发返回结果给用户，设备与云平台、网关建立拓扑关系的流程见图15。

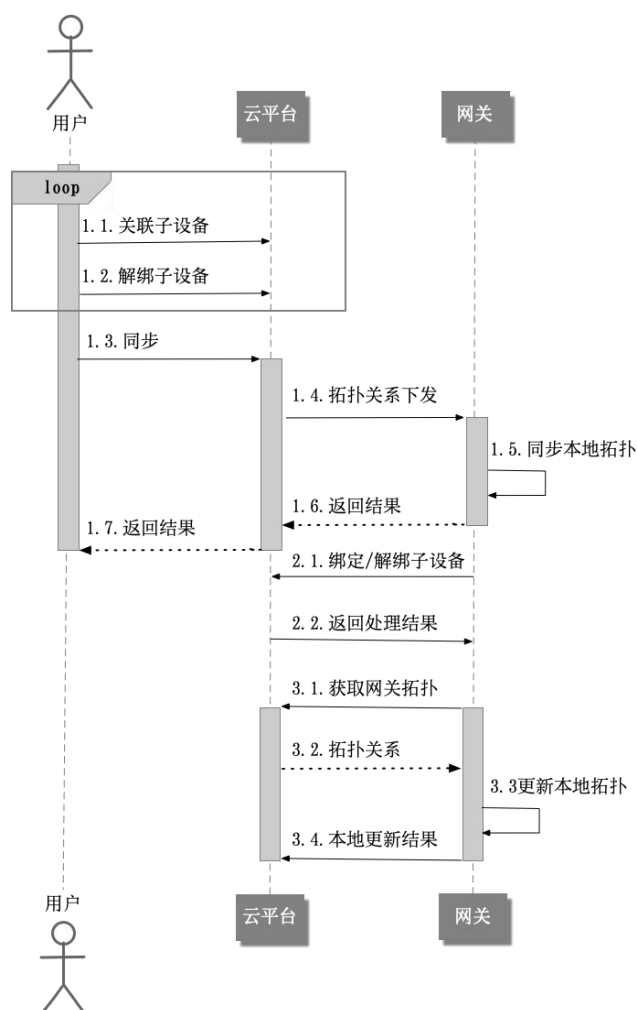


图15 拓扑关系示意图

网关和子设备绑定拓扑关系后，子设备可借助网关通道进行上下线、数据上传、指令接收等操作。网关和子设备绑定与解绑可以通过两种方式变更：

- a) 基于云平台的变更
 - 用户在控制台绑定或解绑子设备；
 - 用户点击同步操作，将拓扑关系变更通知到网关；
 - 网关收到拓扑关系后，进行网关本地拓扑关系同步；
 - 同步结果返回平台；
 - 平台处理后，将最终结果返回给用户。
- b) 基于网关的更改
 - 网关侧发起关联或解绑子设备操作；
 - 平台处理完网关请求后，将处理结果返回给网关设备。
- c) 拓扑关系获取
 - 网关可主动向平台获取拓扑关系；
 - 平台下发拓扑关系到网关；
 - 网关本地同步云平台存储的拓扑关系到本地；
 - 网关回复本地同步结果。

10.7 设备属性/事件

10.7.1 设备属性上报

上行请求 topic: \$sys/{pid}/{device-name}/thing/property/post, 请求参数描述见表 29, JSON 数据格式如下:

```
{
  "id": "123",
  "version": "1.0",
  "params": {
    "Power": {
      "value": "on",
      "time": 1524448722123
    },
    "WF": {
      "value": 23.6,
      "time": 1524448722123
    }
  }
}
```

表29 请求参数描述

参数	类型	描述
id	String	消息 id 号, 用户自定义, String 类型的数字, 长度限制不超过 13 位。
version	String	物模型版本号, 可选字段, 不填默认为 1.0。
params	JsonObject	请求参数, 用户自定义, 标准 json 格式。如以上示例中, 设备上报了的两个属性 Power 和 WF。具体属性信息, 包含属性上报时间 (time) 和上报的属性值 (value)。
time	Long	属性值生成时间。该参数为可选字段, 到毫秒级。根据您的业务场景决定消息中是否带时间戳。如果消息频繁, 需根据时间戳判断消息顺序, 建议消息中带有时间戳。
value	Object	上报的属性值

响应 topic: \$sys/{pid}/{device-name}/thing/property/post/reply, 响应参数见表 30, JSON 数据格式如下:

```
{
  "id": "123",
  "code": 200,
  "msg": "xxxx"
}
```

表30 响应参数描述

参数	类型	描述
Id	String	消息 id 号, 用户自定义, String 类型的数字, 长度限制不超过 13 位。

表 30 响应参数描述（续）

参数	类型	描述
code	Integer	结果状态码
msg	String	错误信息

10.7.2 设备属性设置

下行请求 topic: \$sys/{pid}/{device-name}/thing/property/set，请求参数描述见表 31，JSON 数据格式如下：

```
{
  "id": "123",
  "version": "1.0",
  "params": {
    "temperature": "30.5"
  }
}
```

表31 请求参数描述

参数	类型	描述
id	String	消息 id 号，平台定义，String 类型的数字，长度限制 13。
version	String	物模型版本号，可选字段，不填默认为 1.0。
params	JsonObject	属性设置参数。如以上示例中，设置属性：{"temperature": "30.5"}。

响应 topic: \$sys/{pid}/{device-name}/thing/property/set_reply，具体响应参数描述见表 32，JSON 数据格式如下：

```
{
  "id": "123",
  "code": 200,
  "msg": "xxxx"
}
```

表32 响应参数描述

参数	类型	描述
Id	String	消息 id 号，平台定义，String 类型的数字，长度限制不超过 13 位。
code	Integer	结果状态码
msg	String	错误信息

10.7.3 设备属性获取

下行请求 topic: \$sys/{pid}/{device-name}/thing/property/get, 请求参数描述见表 33, JSON 数据格式如下:

```
{
  "id": "123",
  "version": "1.0",
  "params": [
    "temperature",
    "humidity"
  ]
}
```

表33 请求参数描述

参数	类型	描述
id	String	消息 id 号, 用户自定义, String 类型的数字, 长度限制不超过 13 位。
version	String	物模型版本号, 可选字段, 不填默认为 1.0。
params	JsonObject	获取属性最新值的标识符列表。

响应 topic: \$sys/{pid}/{device-name}/thing/property/get_reply, 响应参数描述见表 34, JSON 数据格式如下:

```
{
  "id": "123",
  "code": 200,
  "msg": "xxx",
  "data": {
    "temperature": 39.5,
    "humidity": 20
  }
}
```

表34 响应参数描述

参数	类型	描述
id	String	消息 id 号, 用户自定义, String 类型的数字, 长度限制不超过 13 位。
code	Integer	结果状态码, 用户自定义。
msg	String	错误信息
data	JsonObject	获取属性最新值列表

10.7.4 设备事件上报

上行请求 topic: \$sys/{pid}/{device-name}/thing/event/post，请求参数描述见表 35，JSON 数据格式如下：

```
{
  "id": "123",
  "version": "1.0",
  "params": {
    "identifier1": {
      "value": {
        "Power": "on",
        "WF": "2"
      },
      "time": 1524448722123
    },
    "identifier2": {
      "value": {
        "Power": "on",
        "WF": "2"
      },
      "time": 1524448722123
    }
  }
}
```

表35 请求参数描述

参数	类型	描述
version	String	物模型版本号，可选字段，不填默认为 1.0。
params	JsonObject	上报事件的参数
time	Long	属性值生成时间。该参数为可选字段，到毫秒级。根据您的业务场景决定消息中是否带时间戳。如果消息频繁，需根据时间戳判断消息顺序，建议消息中带有时间戳。
value	Object	具体的事件信息。如以上示例中{ "Power": "on", "WF": "2" }。

响应 topic: \$sys/{pid}/{device-name}/thing/event/post/reply，响应参数描述见表 36，JSON 数据格式如下：

```
{
  "id": "123",
  "code": 200,
  "msg": "xxxx"
}
```

表36 响应参数描述

参数	类型	描述
Id	String	消息 id 号，用户自定义，String 类型的数字，长度限制不超过 13 位。
code	Integer	结果状态码
msg	String	错误信息

10.7.5 批量数据上报

上行请求 topic: \$sys/{pid}/{device-name}/thing/pack/pos,每次批量上送不超过 10 个子设备，每个设备上送数据点数量不超过 100 个。批量数据上报请求参数格式见表 37。

表37 请求参数说明

参数	类型	说明
id	String	消息 id 号，用户自定义，String 类型的数字，长度限制不超过 13 位。
version	String	物模型版本号，可选字段，不填默认为 1.0。
params	Object	请求参数
properties	Object	属性，包含属性标识符、属性值 value 和属性生成的时间 time。其中，time 参数为可选字段。根据您的业务场景决定消息中是否带时间戳。如果消息频繁，需根据时间戳判断消息顺序，建议消息中带有时间戳。
events	Object	事件，包含事件标识符、事件输出参数 value 和事件生成的时间 time。其中，time 参数为可选字段。根据您的业务场景决定消息中是否带时间戳。如果消息频繁，需根据时间戳判断消息顺序，建议消息中带有时间戳。
identity	Object	设备信息，可选字段，如果有此字段，内容为子设备信息。
productID	String	子设备产品 key
deviceName	String	子设备名称

响应 topic: \$sys/{pid}/{device-name}/thing/pack/post/reply, 响应消息参数见表 38, JSON 数据格式如下：

```
{
  "id": "123",
  "code": 200,
  "msg": "xxxx"
}
```

表38 响应参数说明

参数	类型	说明
id	String	消息 id 号，用户自定义，String 类型的数字，长度限制不超过 13 位。
code	Integer	结果状态码

表 38 响应参数说明（续）

参数	类型	说明
msg	String	错误消息

10.7.6 历史数据上报

上行请求 topic: \$sys/{pid}/{device-name}/thing/history/post，每次批量上送不超过 10 个子设备，每个设备上送数据点数量不超过 100 个，每个功能点历史数据不超过 100 个。历史数据上报请求参数说明见表 39。

表39 请求参数说明

参数	类型	说明
id	String	消息 id 号，用户自定义，String 类型的数字，长度限制不超过 13 位。
version	String	物模型版本号，可选字段，不填默认为 1.0。
params	Object	请求参数
identity	String	数据所属设备的身份证书信息，包含参数 productID 和 deviceName。**可选字段，**默认为 topic 信息中对应的设备数据；说明 直连设备仅能上报自己的物模型数据。网关设备可以上报其子设备的物模型数据。网关上报子设备历史数据时，identity 为子设备的信息。
properties	Object	属性，包含属性标识符、属性值 value 和属性生成的时间 time。
events	Object	事件，包含事件标识符、事件输出参数 value 和事件生成的时间 time。
productID	String	子设备产品 key
deviceName	String	子设备名称

响应 topic: \$sys/{pid}/{device-name}/thing/history/post/reply，JSON 数据格式如下：

```
{
  "id": "123",
  "code": 200,
  "msg": "xxxx"
}
```

历史数据上报响应消息格式见表 40。

表40 响应参数说明

参数	类型	说明
id	String	消息 id 号，用户自定义，String 类型的数字，长度限制不超过 13 位。
code	Integer	结果状态码
msg	String	错误消息

10.7.7 设备服务调用

物联网平台支持同步调用和异步调用，同步调用超时时间 5s，异步调用无超时时间。请求 topic: \$sys/{pid}/{device-name}/thing/service/{identifier}/invoke，JSON 数据格式：

```
{
  "id": "123",
  "version": "1.0",
  "params": {
    "Power1": "on",
    "WF1": "2"
  }
}
```

设备服务调用请求参数描述见表 41。

表41 请求参数描述

参数	类型	描述
identifier	String	功能点唯一标识符（产品下唯一）
id	String	消息 id 号，用户自定义，String 类型的数字，长度限制不超过 13 位。
version	String	物模型版本号，可选字段，不填默认为 1.0
params	Object	服务的请求参数，参数类型具体见物模型对应的“服务”定义（输入参数部分） { "Power1": "on", "WF1": "2" }

响应 topic: \$sys/{pid}/{device-name}/thing/service/{identifier}/invoke_reply，JSON 数据格式如下：

```
{
  "id": "123",
  "code": 200,
  "msg": "xxxx",
  "data": {
    "result1": "on",
    "result2": "2"
  }
}
```

设备服务调用响应参数描述见表 42。

表42 响应参数描述

参数	类型	描述
id	String	消息 id 号，用户自定义，String 类型的数字，长度限制不超过 13 位。

表 42 响应参数描述（续）

参数	类型	描述
code	Integer	结果状态码
msg	String	错误消息
data	Object	服务的响应参数，参数类型具体见物模型对应的“服务”定义（输出参数部分） { "result1": "on", "result2": "2" }

同步调用，返回以上结果，异步调用，平台收到服务调用请求后立即返回以上结果给应用，但是不包含 data 字段；平台收到设备响应后，按以上结果（含 data 字段）将数据通过规则引擎流转 MQ、推送或数据存储，用户通过 MQ、HTTP 推送等服务获取设备服务执行结果，也可通过「设备详情」-「服务记录」查看相关执行结果，亦可通过 API 查询获取执行结果。



电信终端产业协会团体标准
应用于政企场景的物联网产品互通控制技术要求

T/TAF 215—2024

*

版权所有 侵权必究

电信终端产业协会印发
地址：北京市西城区新街口外大街 28 号
电话：010-82052809
电子版发行网址：www.taf.org.cn