**LLSYNC广播协议**

未经授权，禁止扩散

目录

[1.引言 3](#_Toc82112701)

[1.1 背景 3](#_Toc82112702)

[1.2 目的 3](#_Toc82112703)

[2.技术概述 3](#_Toc82112704)

[2.1 AD Type说明 3](#_Toc82112705)

[2.2 数据格式 4](#_Toc82112706)

[3.设备配网 4](#_Toc82112707)

[3.1 未配网广播 6](#_Toc82112708)

[3.2 配网邀请广播 7](#_Toc82112709)

[3.3 配网邀请确认广播 7](#_Toc82112710)

[3.4 配网数据下发广播 8](#_Toc82112711)

[3.5 配网完成广播 9](#_Toc82112712)

[3.6 配网完成确认广播 10](#_Toc82112713)

[4.设备控制 10](#_Toc82112714)

[5.属性上报 12](#_Toc82112715)

[6.设备删除 14](#_Toc82112716)

[7.注意事项 15](#_Toc82112717)

# 1.引言

## 1.1 背景

LLSYNC ADV SDK是使用BLE广播与扫描(以下简称为广播)，使腾讯连连小程序与设备端之间进行通信的技术。双方使用BLE广播进行设备配对、设备绑定、设备控制、设备删除等动作，整个过程不需要蓝牙连接等方式参与，具有交互方式易理解、设备资源要求低、移植适配简单等优点。

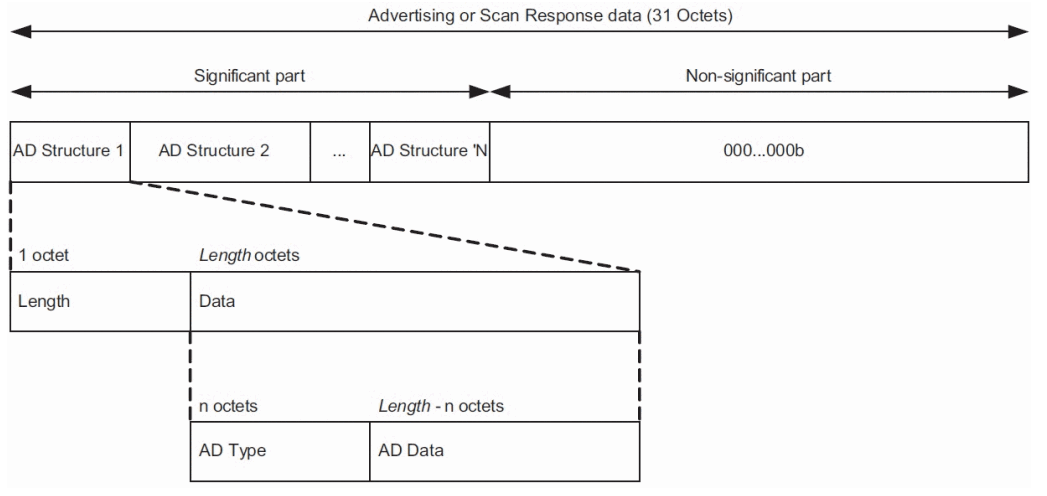
## 1.2 目的

本文档旨在对LLSYNC ADV SDK协议进行细节说明，通过设备配网、设备控制、设备属性上报、设备删除四个方面对协议进行展开。

# 2.技术概述

## 2.1 AD Type说明

LLSYNC ADV SDK协议使用AD Type为0xff «Manufacturer Specific Data»和0x07 «Complete List of 128-bit Service Class UUIDs»(**注意在使用0x07类型时必须保证每次发送数据为128-bit，数据不足使用0x00填充**)进行数据传输，其中0xff主要是设备未配网时广播使用，广播数据中携带设备名称、PID等信息，具体参见设备配网章节，0x07为后期其他状态的数据交互使用。



## 2.2 数据格式

腾讯云物联网为接入平台定义一套数据模板协议，将设备的接入形式通过JSON模板标准化。多数BLE设备受资源限制，较难承载JSON格式的数据交互，针对此定义了TLV格式的二进制数据包来表示数据模板，最大程度的减少资源占用，LLSYNC ADV SDK协议设备控制和属性上报数据内容使用TLV格式进行传输。

**LLSync TLV格式:**

|  |  |  |
| --- | --- | --- |
| 字段 | Type | Value |
| 长度 | 1 Byte | N Bytes |
| 说明 | Type字段定义 | 无 |

**Type字段说明:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| 字段 | 数据类型定义 | | | RFU | | | | |

**数据类型定义**：

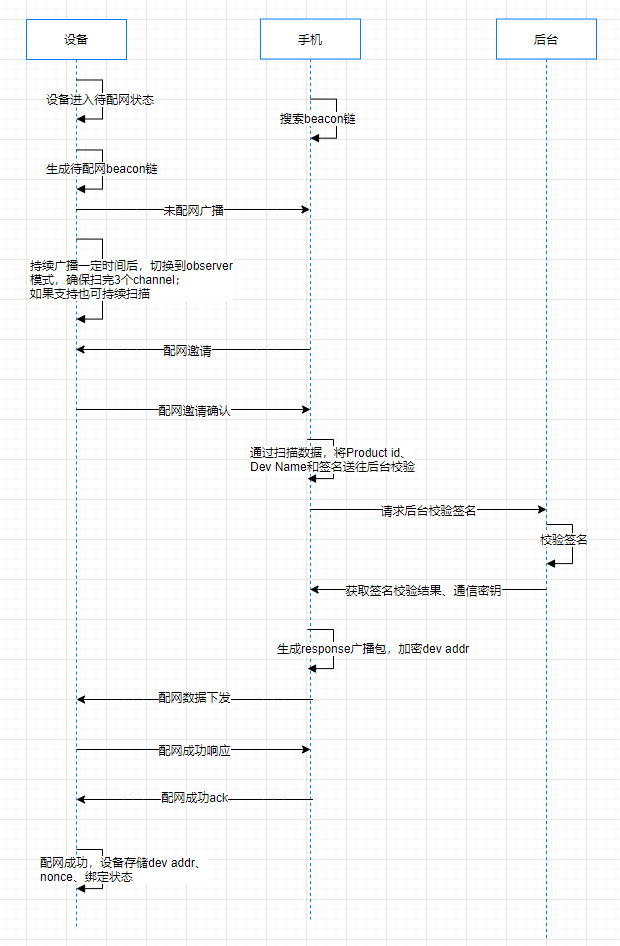
|  |  |  |  |
| --- | --- | --- | --- |
| 数据类型 | 值 | 数据长度 | 数据范围 |
| 布尔 | 0 | 1 Byte | 0/1 |
| 整数 | 1 | 4 Bytes | -2^31 ~ 2^31 - 1 |
| 枚举 | 4 | 2 Bytes | 0 ~ 2^16 - 1 |

TLV数据类型示例：

* 00 01：第1个Byte高3bits为0表示布尔类型，01为具体value值；
* 20 00 00 00 01: 第1个Byte高3bits为1表示整数类型，01为具体value值；
* 80 00 01：第1个Byte高3bits为4表示枚举类型，01为具体value值。

# 3.设备配网

设备配网阶段时序图如下所示：



数据格式定义如下(未配网除外)：

| **Field** | **Size(Octets)** | **Notes** |
| --- | --- | --- |
| CID | 2 | 0xFEBA |
| PackageNum | 1 | 高 4 Bits 表示总包数，低 4 Bits 表示总包分片后本包序号 |
| Length | 1 | 后续数据有效长度 |
| MsgHeader | 1 | 高 4 Bits 表示消息ID，从0x01 到0x0F循环递增，低 4Bits表示MsgType。 |
| MAC | 2 | 配网过程中携带的 2 字节MAC，设备通过此来筛选配网邀请 |
| Payload | N | N <= 9 Bytes |

Payload字段由MsgType决定，MsgType用来区表示配网过程中数据包的作用：

| **MsgType** | **Value** | **Description** |
| --- | --- | --- |
| Provisioning Invite | 0x00 | 配网邀请 |
| Provisioning Invite ACK | 0x01 | 配网邀请确认 |
| Provisioning Data | 0x02 | 配网数据下发 |
| Provisioning Complete | 0x03 | 配网完成 |
| Provisioning Complete ACK | 0x04 | 配网完成确认 |
| Provisioning Delete | 0x05 | 配网解除 |
| RFU | 0x06 - 0x07 | - |

## 3.1 未配网

未配网广播使用AD Type为0xff的类型进行数据交互。

| **Field** | **Size(Octet)** | **Notes** |
| --- | --- | --- |
| CID | 2 | Compnay Identify，固定为0xFEBA |
| Frame Control | 2 | 控制位 |
| MAC[4:5] | 2 | 设备MAC地址最后2个字节 |
| PID | 10 | Product ID，一型一号 |
| DevNameLen | 1 | 设备名称长度 |
| DevName | N | Device name，物联网开发平台中设备名称，最长为8字节 |
| SplitCycle | 1 | 小程序分片周期，单位10ms |

* Frame Control预留2字节，用来表示版本号和设备端支持的功能。

| **Bit** | **Notes** |
| --- | --- |
| 3 ～ 0 | 协议版本号，目前为0 |
| 11 ～ 4 | RFU 协议功能 |
| 14 ～ 12 | RFU 协议类型 |
| 15 | 标准广播协议 |

* 未配网设备beacon示例：

| **Length** | **AD Type** | **CID** | **Frame Control** | **MAC[4:5]** |
| --- | --- | --- | --- | --- |
| 0x18 | 0xFF | 0xFEBA | 0x8000 | 0x3132 |
| **PID** | **NameLen** | **DevName** | **SplitCycle** | **-** |
| 0123456789 | 0x07 | Tencent | 0x28 | - |

## 3.2 配网邀请

* 小程序向设备发出配网邀请，设备响应后开始配网。

| **Field** | **Size(Octets)** | **Notes** |
| --- | --- | --- |
| CID | 2 | 0xFEBA |
| PackageNum | 1 | 高 4 Bits 表示总包数，低 4 Bits 表示总包分片后本包序号 |
| Length | 1 | 后续数据有效长度 |
| MsgHeader | 1 | 高 4 Bits 表示消息ID，从0x01 到0x0F循环递增，低 4Bits表示MsgType。 |
| MAC | 2 | 未配网beacon中携带的 2 字节MAC，设备通过此来筛选配网邀请 |
| Nonce | 2 | 小程序生成的随机数(**大端发送**)。用于设备认证计算。取值:0x0000 ~ 0xFFFF |

* 配网邀请广播配网示例：

| **Length** | **AD Type** | **CID** | **PackageNum** | **Length** | **MsgHeader** | **MAC** | **Nonce** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x11 | 0x07 | 0xFEBA | 0x11 | 0x05 | 0x00 | 0x3132 | 0x1234 |

## 3.3 配网邀请确认

|  |  |  |
| --- | --- | --- |
| **Field** | **Size(Octets)** | **Notes** |
| CID | 2 | 0xFEBA |
| Package Num | 1 | 高 4 Bits 表示总包数，低 4 Bits 表示总包分片后本包序号 |
| Length | 1 | 后续数据有效长度 |
| MsgHeader | 1 | 高 4 Bits 消息ID为Provisioning Invite的ID，低 4Bits表示MsgType。 |
| MAC | 2 | 未配网beacon中携带的 2 字节MAC |
| ACK | 1 | 设备是否同意配网，不同意配网时无Sign字段，配网结果(0成功，1失败) |
| Sign | 8 | 鉴权信息 |

Sign = MD5(AES128-ecb(MD5(psk xor nonce)，provisioning data))

1.MD5(psk xor nonce)：设备psk和配网邀请nonce进行异或，nonce位数不足，循环使用;

2.provisioning data：产品ID+设备名称+nonce;

3.AES128-ecb:加密使用AES-ecb模式，PKCS7 Padding填充;

4.Sign = MD5(AES128-ecb(xxxxx)): md5前8字节和后8字节进行xor操作，确保将sign信息放在一个数据包内，不用分包。

* 用户拒绝配网示例：

| **Length** | **AD Type** | **CID** | **PackageNum** | **Length** | **MsgHeader** | **MAC** | **ACK** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x11 | 0x07 | 0xFEBA | 0x11 | 0x04 | 0x01 | 0x3132 | 0x01 |

* 用户同意配网示例：

| **Length** | **AD Type** | **CID** | **PackageNum** | **Length** | **MsgHeader** | **MAC** | **ACK** | **Sign** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0x11 | 0x07 | 0xFEBA | 0x11 | 0x0C | 0x01 | 0x3132 | 0x00 | 0xaf5df8da4bc264b5 |

Sign计算过程如下(此处提供一组结果供参考)：

PRODUCT ID：9ESYQHPQF3

Dev Name:“wo”

PSK：“uE4To+1zc\*\*\*\*t6CIEIYSA==”

Nonce：0xf54f

Sign=0xaf5df8da4bc264b5

## 3.4 配网数据下发

小程序收到配网邀请确认包后，将设备的Product id，Device name，Nonce上传到后台进行验证。校验通过后，小程序生成配网数据，使用通信密钥加密后发送给设备。

| **Field** | **Size(Octets)** | **Notes** |
| --- | --- | --- |
| CID | 2 | 0xFEBA |
| Package Num | 1 | 高 4 Bits 表示总包数，低 4 Bits 表示总包分片后本包序号 |
| Length | 1 | 后续数据有效长度 |
| MsgHeader | 1 | 高 4 Bits 表示消息ID，从0x01 到0x0F循环递增，低 4Bits表示MsgType。 |
| MAC | 2 | 未配网beacon中携带的 2 字节MAC |
| Encrypt provisioning data | N | 加密的配网数据 |

Encrypt provisioning data = AES128-ecb（Key，Device Address）

| **Field** | **Size(Octets)** | **Notes** |
| --- | --- | --- |
| Device Address | 2 | 设备被分配的广播地址，后续控制报文使用该地址识别设备 |

## 3.5 配网完成

设备收到小程序的配网数据后，对配网数据解密后进行存储，并回复配网成功与否。配网数据在后续控制过程中使用。

| **Field** | **Size(Octets)** | **Notes** |
| --- | --- | --- |
| CID | 2 | 0xFEBA |
| Package Num | 1 | 高 4 Bits 表示总包数，低 4 Bits 表示总包分片后本包序号 |
| Length | 1 | 后续数据有效长度 |
| MsgHeader | 1 | 高 4 Bits 消息ID为Provisioning Data的ID，低 4Bits表示MsgType。 |
| Device Address | 2 | 配网成功时的设备地址，同一个用户小程序内确保不重复 |
| ACK | 1 | 配网结果(0成功，1失败) |
| Sign | 8 | 成功携带，使用最新dec addr进行计算，算法同配网邀请确认保持一致 |

配网成功数据示例：

| **Length** | **AD Type** | **CID** | **PackageNum** | **Length** | **MsgHeader** | **Device Address** | **ACK** | **Sign** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 0x11 | 0x07 | 0xFEBA | 0x11 | 0x0C | 0x03 | 0x3132 | 00 | 0xB00A42E83E11BE00 |

解密失败则配网失败示例：

| **Length** | **AD Type** | **CID** | **PackageNum** | **Length** | **MsgHeader** | **MAC** | **ACK** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x11 | 0x07 | 0xFEBA | 0x11 | 0x04 | 0x03 | 0x3132 | 01 |

## 3.6 配网完成确认

小程序发送给设备，设备收到后表示配网成功的数据小程序已收到，此时将Nonce、Dev addr和绑定状态等信息存入flash或其他非易失性存储介质。

| **Field** | **Size(Octets)** | **Notes** |
| --- | --- | --- |
| CID | 2 | 0xFEBA |
| Package Num | 1 | 高 4 Bits 表示总包数，低 4 Bits 表示总包分片后本包序号 |
| Length | 1 | 后续数据有效长度 |
| MsgHeader | 1 | 高 4 Bits 消息ID为Provisioning Data的ID，低 4Bits表示MsgType。 |
| Device Address | 2 | 配网成功时的设备地址，同一个用户小程序内确保不重复 |
| ACK | 1 | 配网结果(0成功，1失败) |

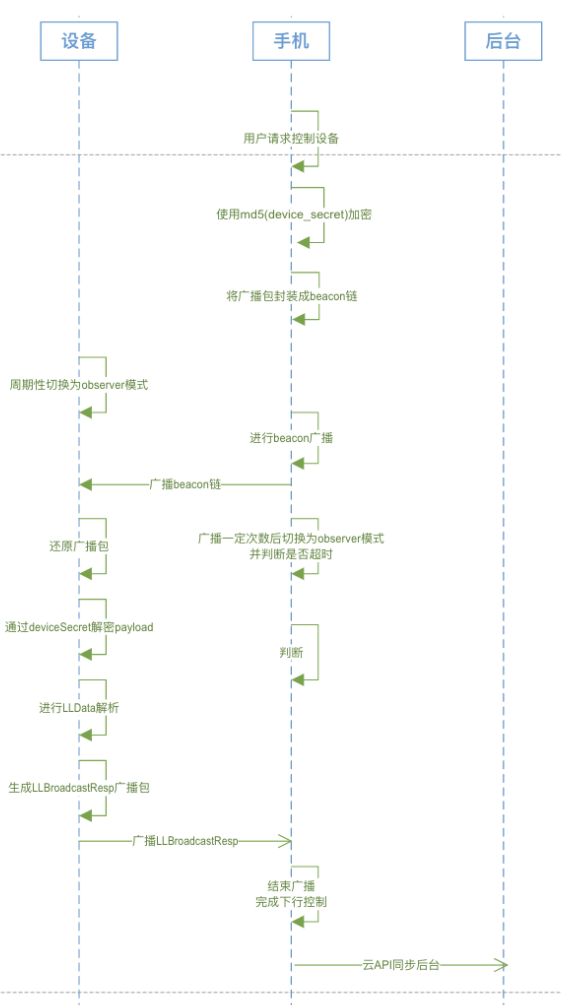
配网完成ack示例：

| **Length** | **AD Type** | **CID** | **PackageNum** | **Length** | **MsgType** | **Device Address** | **ACK** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x11 | 0x07 | 0xFEBA | 0x11 | 0x03 | 0x04 | 0x1234 | 00 |

到此处设备配网完成。

# 4.设备控制

设备控制时序图如下：



配网成功后设备和小程序绑定，小程序可以控制设备，设备可以上报数据到小程序。

控制报文仍然包含在128 bits的UUID中，UUID格式如下：

| **Field** | **Size(Octets)** | **Notes** |
| --- | --- | --- |
| CID | 2 | 0xFEBA |
| PackageNum | 1 | 高 4 Bits 表示总包数，低 4 Bits 表示总包分片后本包序号 |
| Length | 1 | 后续数据有效长度 |
| MsgHeader | 1 | 高 4 Bits 表示消息ID，从0x01 到0x0F循环递增，低 4Bits表示MsgType |
| Device Address | 2 | 设备地址 |
| Payload | N | N <= 9 Bytes (使用AES-ecb加密, PKCS7 Padding) |

MsgType用来区分控制报文。

| **MsgType** | **Value** | **Description** |
| --- | --- | --- |
| Control | 0x08 | 小程序控制设备 |
| Report | 0x09 | 设备上报小程序 |
| RFU | 0x0A ~ 0x0E | - |
| ACK | 0x0F | 确认报文 |

Payload取值参考具体MsgType定义。

小程序控制报文报文格式：

| **Field** | **Size(Octets)** | **Notes** |
| --- | --- | --- |
| CID | 2 | 0xFEBA |
| PackageNum | 1 | 高 4 Bits 表示总包数，低 4 Bits 表示总包分片后本包序号 |
| Length | 1 | 后续数据有效长度 |
| MsgHeader | 1 | 高 4 Bits 表示消息ID，从0x01 到0x0F循环递增，低 4Bits表示MsgType |
| Device Address | 2 | 设备地址 |
| Payload | N | N <= 9 Bytes (使用AES-ecb加密, PKCS7 Padding) |

控制报文示例：

| **Length** | **AD Type** | **CID** | **PackageNum** | **Length** | **MsgHeader** | **Device Address** | **Payload** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x11 | 0x07 | 0xFEBA | 0x11 | 0x04 | 0x08 | 0x1234 | 0x00 |

设备端在接收到控制报文后，解密数据，完成对应动作，然后进行ack响应。

设备端回复报文：

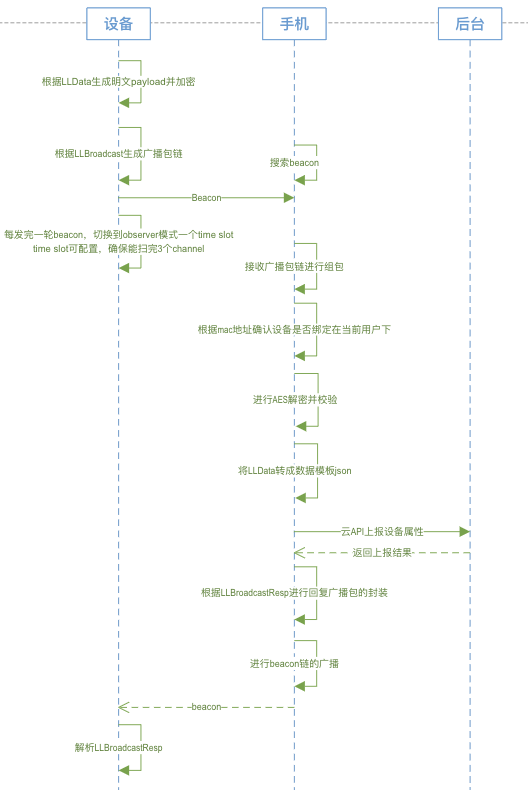
| **Field** | **Size(Octets)** | **Notes** |
| --- | --- | --- |
| CID | 2 | 0xFEBA |
| PackageNum | 1 | 高 4 Bits 表示总包数，低 4 Bits 表示总包分片后本包序号 |
| Length | 1 | 后续数据有效长度 |
| MsgHeader | 1 | 高 4 Bits消息ID为控制报文ID，低 4Bits表示MsgType |
| Device Address | 2 | 设备地址 |
| Payload | 2 | result |

控制数据包确认报文示例：

| **Length** | **AD Type** | **CID** | **PackageNum** | **Length** | **MsgHeader** | **Device Address** | **Payload** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x11 | 0x07 | 0xFEBA | 0x11 | 0x04 | 0x0F | 0x1234 | 0x01 |

# 5.属性上报

设备属性上报时序图如下所示：



设备属性上报报文格式：

| **Field** | **Size(Octets)** | **Notes** |
| --- | --- | --- |
| CID | 2 | 0xFEBA |
| PackageNum | 1 | 高 4 Bits 表示总包数，低 4 Bits 表示总包分片后本包序号 |
| Length | 1 | 后续数据有效长度 |
| MsgHeader | 1 | 高 4 Bits消息ID为控制报文ID，低 4Bits表示MsgType |
| Device Address | 2 | 设备地址 |
| Payload | N | N <= 9 Bytes |

属性上报报文示例：

| **Length** | **AD Type** | **CID** | **PackageNum** | **Length** | **MsgHeader** | **Device Address** | **Payload** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x11 | 0x07 | 0xFEBA | 0x11 | 0x04 | 0x09 | 0x1234 | 0x00 |

小程序收到设备上报报文后，会进行确认包回复.设备端可以利用此报文决定是否进行数据重传。

| **Field** | **Size(Octets)** | **Notes** |
| --- | --- | --- |
| CID | 2 | 0xFEBA |
| PackageNum | 1 | 高 4 Bits 表示总包数，低 4 Bits 表示总包分片后本包序号 |
| Length | 1 | 后续数据有效长度 |
| MsgHeader | 1 | 高 4 Bits消息ID为属性上报ID，低 4Bits表示MsgType |
| Device Address | 2 | 设备地址 |
| Payload | N | N <= 9 Bytes |

* 确认报文MsgHeader内的MID表示是对收到的哪个数据包的ACK，不消息设备当前的MID。
* Payload为设备回去确认的报文类型。

确认报文报文示例：

| **Length** | **AD Type** | **CID** | **PackageNum** | **Length** | **MsgHeader** | **Device Address** | **Payload** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x11 | 0x07 | 0xFEBA | 0x11 | 0x04 | 0x0F | 0x1234 | 0x09 |

# 6.设备删除

小程序可以主动解除设备绑定，以便设备重新配网。

| **Field** | **Size(Octets)** | **Notes** |
| --- | --- | --- |
| CID | 2 | 0xFEBA |
| Package Num | 1 | 高 4 Bits 表示总包数，低 4 Bits 表示总包分片后本包序号 |
| Length | 1 | 后续数据有效长度 |
| MsgHeader | 1 | 0x05 |
| Device Address | 2 | 配网成功时的设备地址，同一个用户小程序内确保不重复 |

配网删除示例数据：

| **Length** | **AD Type** | **CID** | **PackageNum** | **Length** | **MsgType** | **Device Address** |
| --- | --- | --- | --- | --- | --- | --- |
| 0x11 | 0x07 | 0xFEBA | 0x11 | 0x03 | 0x05 | 0x1234 |

* Provisioning Delete ACK

设备收到delete信息后进行广播上报响应

| **Field** | **Size(Octets)** | **Notes** |
| --- | --- | --- |
| CID | 2 | 0xFEBA |
| Package Num | 1 | 高 4 Bits 表示总包数，低 4 Bits 表示总包分片后本包序号 |
| Length | 1 | 后续数据有效长度 |
| MsgHeader | 1 | 高 4 Bits 消息ID为Provisioning Delete的ID，低 4Bits表示MsgType。 |
| Device Address | 2 | 配网成功时的设备地址，同一个用户小程序内确保不重复 |
| ACK | 1 | 0x00 |

删除配网响应事例：

| **Length** | **AD Type** | **CID** | **PackageNum** | **Length** | **MsgTYPE** | **MAC** | **ACK** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 0x11 | 0x07 | 0xFEBA | 0x11 | 0x02 | 0x05 | 0x3132 | 0x00 |

# 7.注意事项

7.1在腾讯云平台进行设备创建时，注意device name的长度不能超过8 Bytes；  
7.2双向广播支持的数据类型为：布尔、整型和枚举，故在创建产品时注意选择相应的数据类型；

7.3双向广播支持的最大数据长度为124 Bytes；  
7.4广播类型需要设置为不可连接、不可扫描和非定向型广播。