

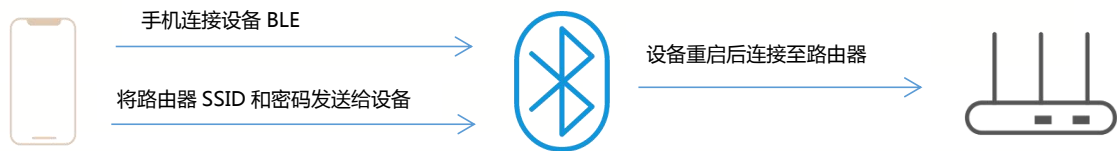
第三方小程序调用蓝牙 BLE 配网小程序使用指南

v19042401

1. 蓝牙 BLE 配网概述

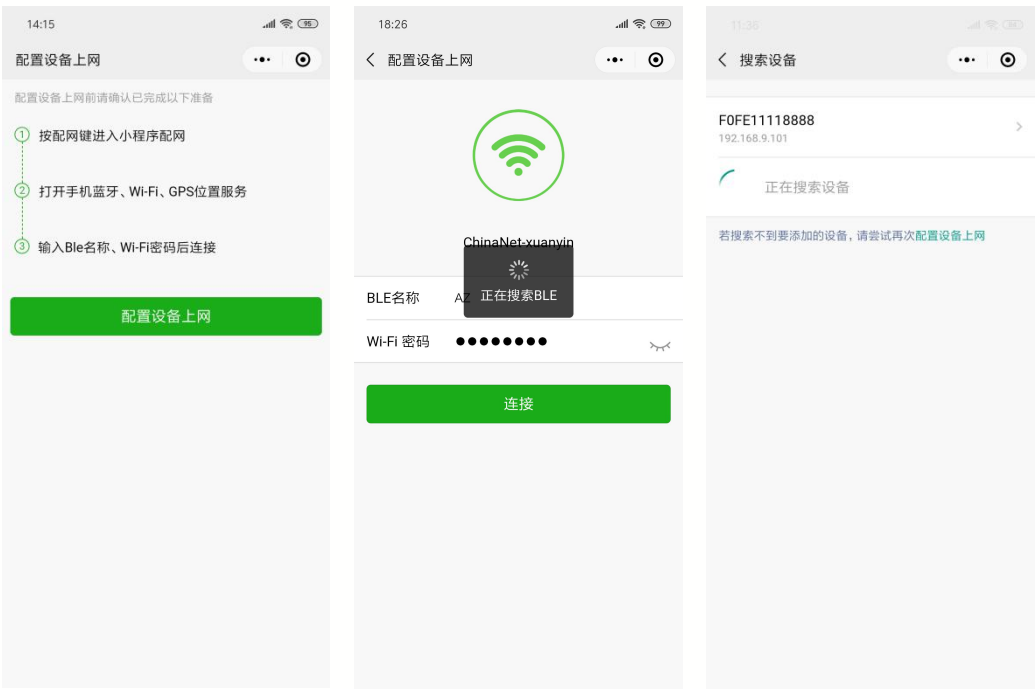
蓝牙 BLE 配网是指使设备进入 BLE 配网模式，手机连接到设备 BLE 后将公网路由器的 SSID 和密码发送给设备以完成设备联网的一种配网方式。

基本工作原理如下图所示：



2. 蓝牙 BLE 配网小程序概述

蓝牙 BLE 配网小程序是由上海汉枫电子科技有限公司开发的针对该司产品的 BLE 配网微信端小程序。该小程序操作体验上与微信 Airkiss 配网类似。



【使用限制】

微信客户端小程序基础库版本 $\geq 2.4.0$

3. 蓝牙 BLE 配网小程序的 AppId

wx4beabf3f3ca2bb43

4. 调用蓝牙 BLE 配网小程序

4.1 调用小程序的 Api

`wx.navigateToMiniProgram(Object object)`

基础库 1.3.0 开始支持，低版本需做兼容处理。

打开另一个小程序

参数

Object object

属性	类型	默认值	必填	说明
appId	string		是	要打开的小程序 appId
path	string		否	打开的页面路径，如果为空则打开首页。path 中 ? 后面的部分会成为 query，在小程序的 <code>App.onLaunch</code> 、 <code>App.onShow</code> 和 <code>Page.onLoad</code> 的回调函数或小程序的 <code>wx.onShow</code> 回调函数、 <code>wx.getLaunchOptionsSync</code> 中可以获取到 query 数据。对于小游戏，可以只传入 query 部分，来实现传参效果，如：传入 <code>"?foo=bar"</code> 。
extraData	object		否	需要传递给目标小程序的数据，目标小程序可在 <code>App.onLaunch</code> 、 <code>App.onShow</code> 中获取到这份数据。如果跳转的是小游戏，可以在 <code>wx.onShow</code> 、 <code>wx.getLaunchOptionsSync</code> 中可以获取到这份数据数据。
envVersion	string	release	否	要打开的小程序版本。仅在当前小程序为开发版或体验版时此参数有效。如果当前小程序是正式版，则打开的小程序必定是正式版。
success	function		否	接口调用成功的回调函数
fail	function		否	接口调用失败的回调函数
complete	function		否	接口调用结束的回调函数（调用成功、失败都会执行）

该接口的使用方法及限制条件请具体参考微信官方文档：

<https://developers.weixin.qq.com/miniprogram/dev/api/wx.navigateToMiniProgram.html>

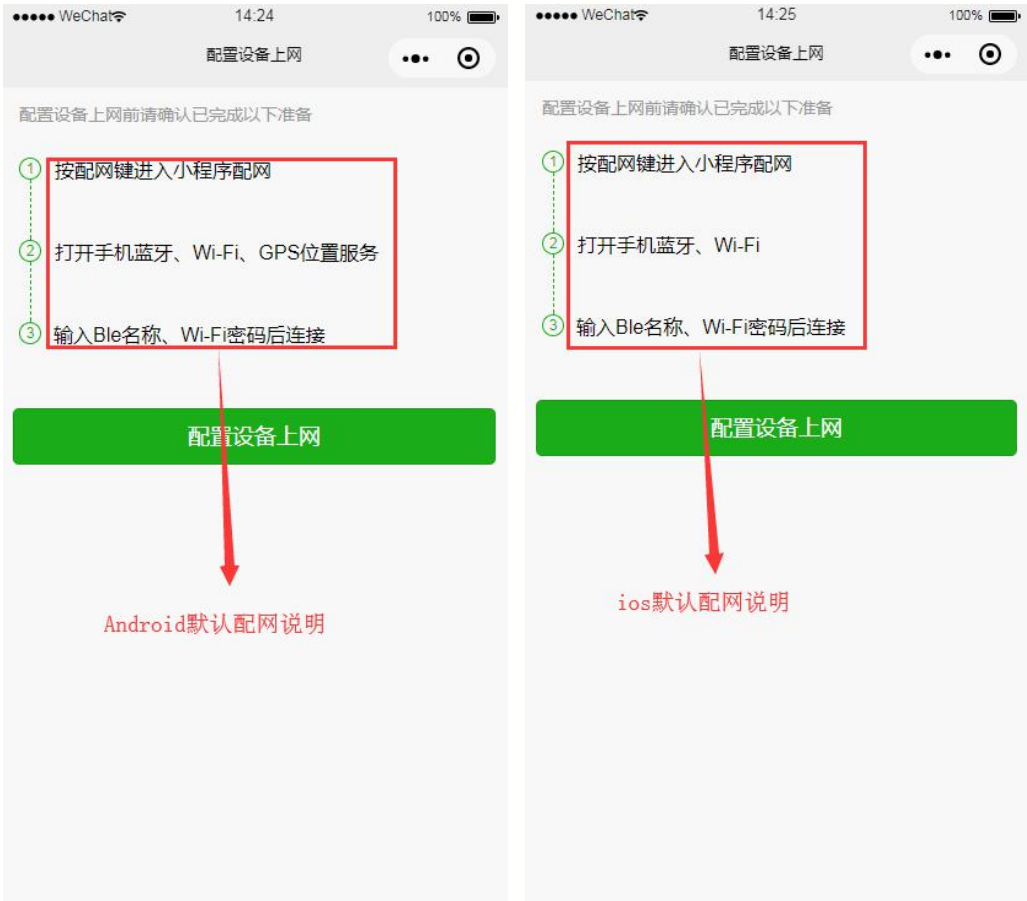
4.2 参数说明

针对使用 4.1 中所示 api 在调用蓝牙 BLE 配网小程序时，需要填入的参数说明如下：

属性		类型	默认值	必填	说明
appId		string		是	蓝牙 BLE 配网小程序的 appId: wx4beabf3f3ca2bb43
extraData	bleName	string	AZ	是	蓝牙 BLE 配网时的设备 BLE 名称； 必须为完整 BLE 名称，不支持部分 BLE 名称；
	hideBleInfoInputs	boolean	false	否	是否隐藏配网界面的 BLE 信息输入框(BLE 名称)
	guide	object		否	配网说明页的文字
	userData	string		否	蓝牙 BLE 配网时额外需要发送给设备的数据

4.3 修改配网说明

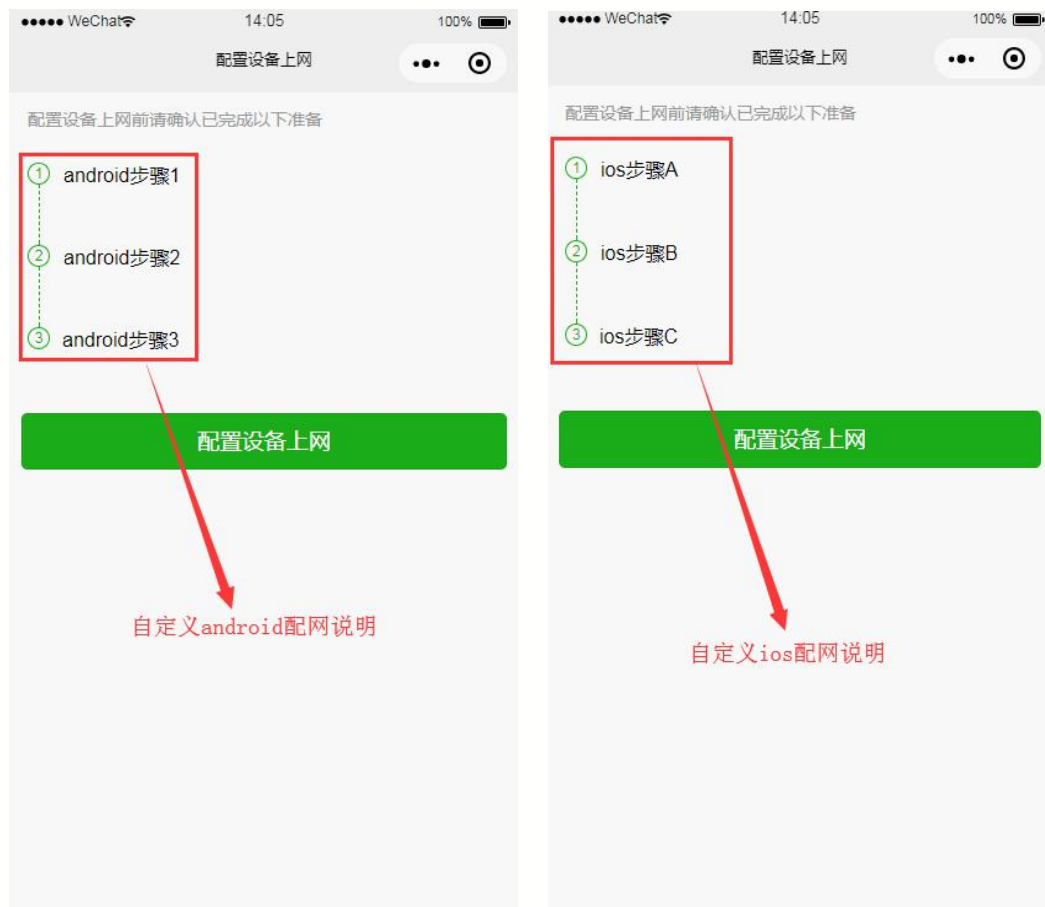
配网说明分 android、ios 两个平台，默认如下：



第三方可根据自己设备的实际配网情况通过 `extraData.guide` 参数修改默认的配网说明，格式如下：

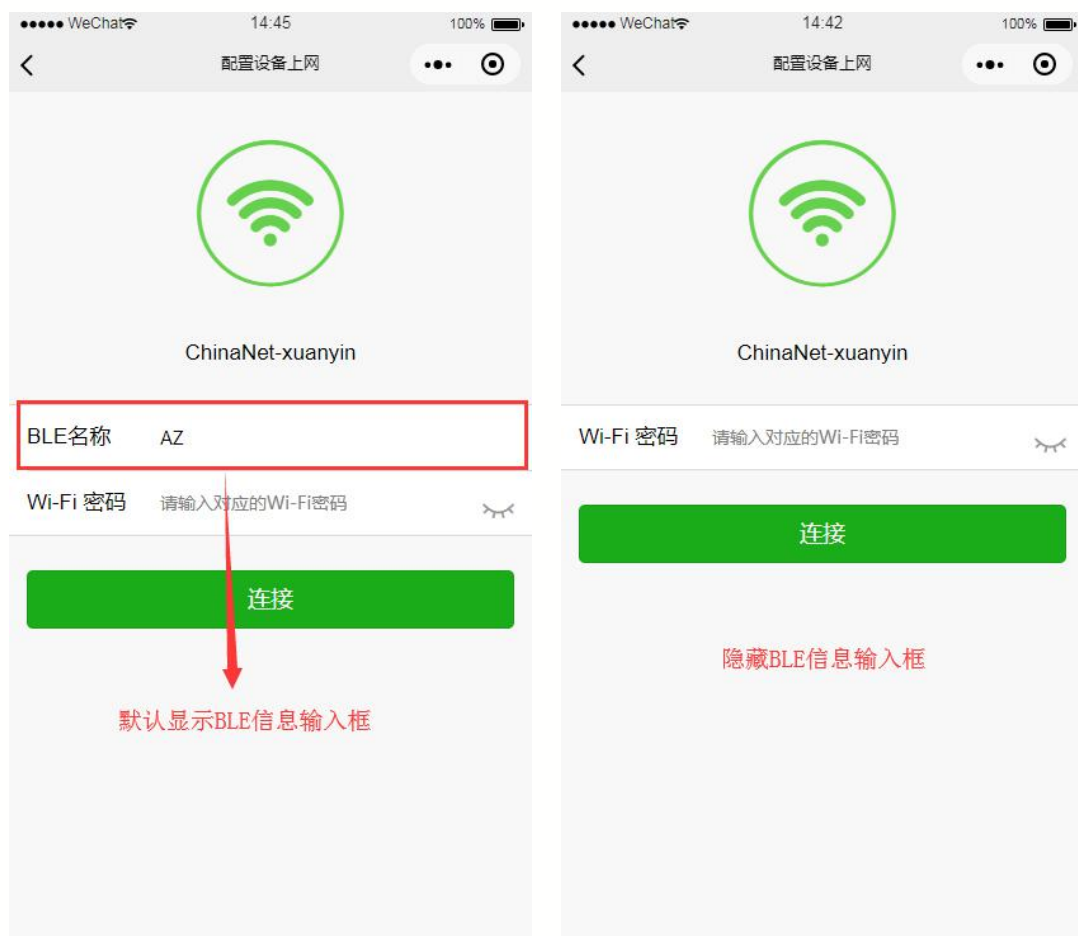
```
{  
  android: ["android 步骤 1", "android 步骤 2", "android 步骤 3"],  
  ios: ["ios 步骤 A", "ios 步骤 B", "ios 步骤 C"]  
}
```

修改后的效果如下：



4.4 隐藏配网界面 BLE 名称输入框

蓝牙 BLE 配网小程序默认显示 BLE 名称输入框，但是对于第三方小程序来说通常不需要显示。可通过 `extraData.hideBleInfoInputs` 来隐藏/显示该输入框。



当 BLE 名称输入框可见时，终端用户可手动修改 BLE 名称并自动缓存。

对于开发阶段，为方便调试可显示 BLE 名称；上线后建议隐藏。

4.5 参考代码

```
wx.navigateToMiniProgram({
  appId: 'wx4beabf3f3ca2bb43',
  extraData: {
    bleName: 'myble',
    hideBleInfoInputs: true,
    guide: {
      android: ["android 步骤 1", "android 步骤 2", "android 步骤 3"],
      ios: ["ios 步骤 A", "ios 步骤 B", "ios 步骤 C"]
    },
    userData: null
  },
},
```

```
success: function(res) {},  
fail: function(res) {},  
complete: function(res) {},  
});
```

5. 接收配网结果

调用蓝牙 BLE 配网小程序成功配网并发现设备后，当终端用户点击选中某个设备时，蓝牙 BLE 配网小程序通过调用 `wx.navigateBackMiniProgram` 自动返回至发起调用的第三方小程序，并同时传送已选中的设备信息。第三方小程序可通过 `App.onShow` 中获取到这份数据。



<https://developers.weixin.qq.com/miniprogram/dev/api/wx.navigateBackMiniProgram.html>

5.1 设备信息

mac : 设备 MAC 地址

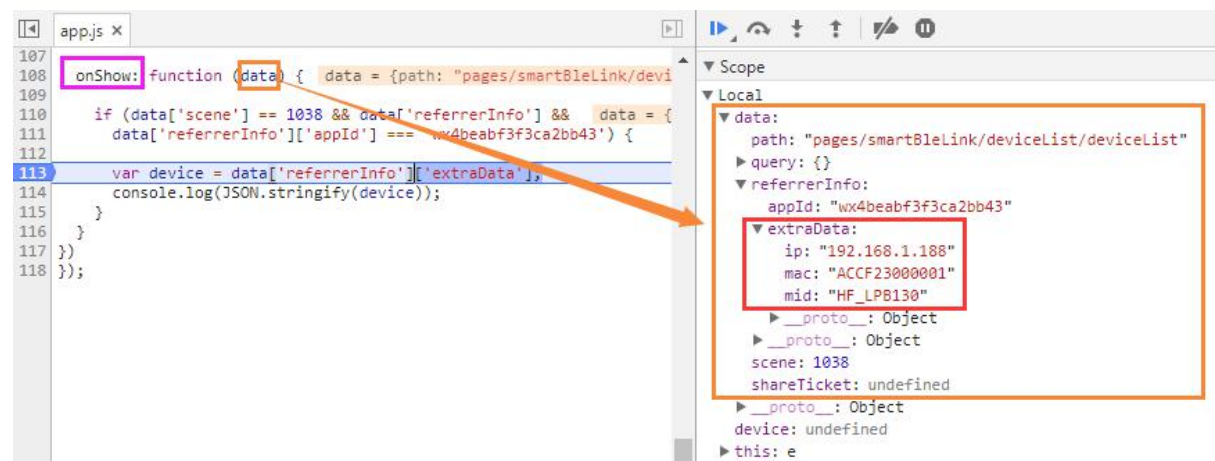
ip : 设备 IP 地址

mid : 设备 ID (通常为设备型号之类的信息)

5.2 参考代码

app.js

```
onShow: function (data) {  
  
    if (data['scene'] == 1038 && data['referrerInfo'] &&  
        data['referrerInfo']['appId'] === 'wx4beabf3f3ca2bb43') {  
  
        var device = data['referrerInfo']['extraData'];  
        console.log(JSON.stringify(device));  
    }  
}
```



【说明】

data['scene']为 1038 表示从先前打开的小程序返回；

wx4beabf3f3ca2bb43 为蓝牙 BLE 配网小程序的 AppId。

6. BLE 名称不固定的解决方案

蓝牙 BLE 配网小程序的参数要求明确指定 BLE 名称，对于设备 BLE 名称不固定的情况可采用先扫二维码然后再调用蓝牙 BLE 配网小程序的方式，具体思路大体如下(仅供参考)：

(1) 设备生产前为每个设备生成一个 uuid 并录入自有系统；

(2) 组织 uuid、BLE 名称、MAC 地址等信息并生成二维码；

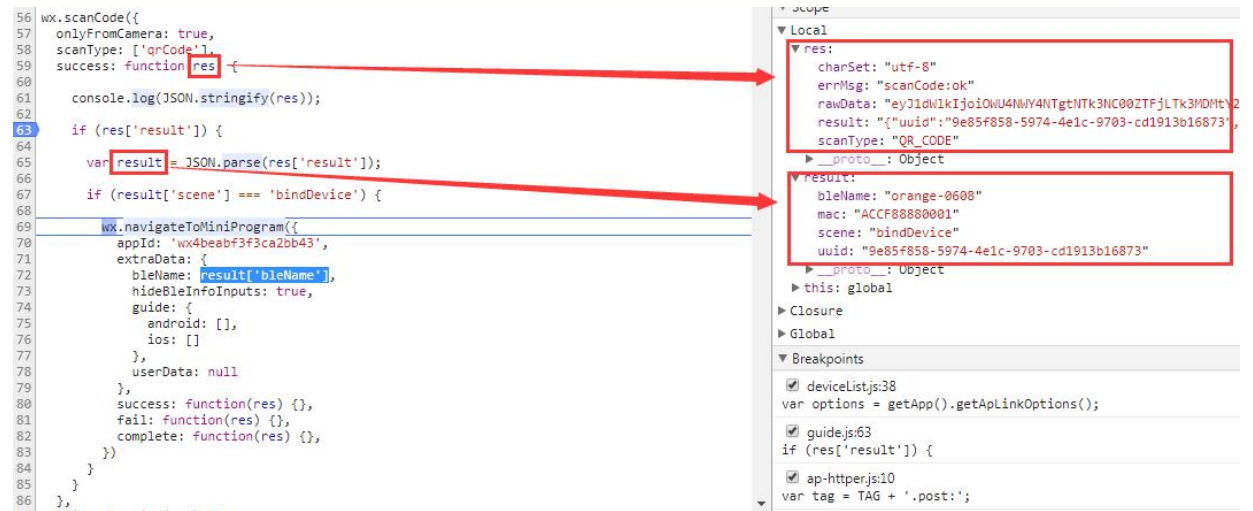
```
{ "uuid": "9e85f858-5974-4e1c-9703-cd1913b16873", "bleName": "orange-0608", "mac": "ACCF88880001", "scene": "bindDevice" }
```

这里 BLE 名称以 orange-xxxx 格式为例，xxxx 为设备 MAC 地址的后四位



(3) 印刷制作纸质二维码并粘贴至设备外壳或产品外包装上；

(4) 第三方小程序自行开发扫码功能，解析扫码结果，根据扫码结果决定跳转至蓝牙 BLE 配网小程序（微信限制 wx.navigateToMiniProgram 只能由用户点击触发，因此以下代码在实际情况中需要做微调）；



(5) 第三方小程序接收蓝牙 BLE 配网小程序返回的设备信息 , 完成自有系统的设备绑定动

作。

onShow: function (data) {

```

if (data['scene'] === 1038 && data['referrerInfo'] &&
    data['referrerInfo']['appId'] === 'wx4beabf3f3ca2bb43') {

```

```

    var device = data['referrerInfo']['extraData'];
    console.log(JSON.stringify(device));

```

```

    if(device['mac'] === 'ACCF88880001') {
      bindDevice(openId, deviceUuid);
    }

```

```

  }
}

```