

# WM\_W800\_BleWiFi 蓝牙配网

## Android SDK

### V1.0

北京联盛德微电子有限公司 (winner micro)

地址：北京市海淀区阜成路 67 号银都大厦 18 层

电话：+86-10-62161900

公司网址：[www.winnermicro.com](http://www.winnermicro.com)

## 文档修改记录

版本	修订时间	修订记录	作者	审核
V0.1	2019/9/25	[C]创建文档	Wanghf	
V0.2	2020/7/8	统一字体	Cuiyc	
V1.0	2020/8/10	升级版本号	Cuiyc	

## 目录

文档修改记录 .....	2
目录 .....	3
1 引言 .....	5
1.1 概述 .....	5
2 接口定义 .....	5
2.1 BleWiFiClient 类 .....	5
2.1.1 构造函数 .....	5
2.1.2 connect 方法 .....	5
2.1.3 negotiateSecretKey 方法 .....	6
2.1.4 configureSta 方法 .....	6
2.1.5 setBleWiFiCallback 方法 .....	6
2.1.6 close 方法 .....	6
2.2 BleWiFiCallback 接口 .....	7
2.2.1 onConnected 方法 .....	7
2.2.2 onDisconnected 方法 .....	7
2.2.3 onServicesDiscovered 方法 .....	7
2.2.4 onConfigureStaResult 方法 .....	8
2.2.5 onNegotiateSecretKeyResult 方法 .....	8
2.2.6 onError 方法 .....	8
2.3 BleWiFiStaParams 类 .....	9
2.3.1 setSsid 方法 .....	9

2.3.2	setPassword 方法 .....	9
2.3.3	setBssid 方法 .....	9
2.4	BleWiFiBaseResult 类 .....	10
2.4.1	getStatus 方法 .....	10
2.4.2	错误号定义 .....	10
2.5	BleWiFiConfigStaResult 类 .....	11
2.5.1	getMac 方法 .....	11
2.5.2	getIpAddress 方法 .....	11
3	使用示例 .....	12
3.1	实例化 BleWiFiClient .....	12
3.2	设置 BleWiFiCallback .....	12
3.3	连接 Device .....	12
3.4	与 Device 协商数据加密密钥 .....	12
3.5	开始配网 .....	13

## 1 引言

### 1.1 概述

BleWiFi 是一款基于低功耗蓝牙（BLE）通道，实现 WiFi 网络配置功能的协议，适用于 W800 芯片。

本文档介绍了我司提供的 BleWiFi for Android 的 SDK 接口使用方法，方便用户进行 BleWiFi 的二次开发，快速将 BleWiFi 实现并集成到自己的 Android App 中。

## 2 接口定义

### 2.1 BleWiFiClient 类

本类提供了与 Device 通讯的所有 API，这些 API 可以帮助 APP 轻松实现通过蓝牙 BLE 对 Device 进行配置。

#### 2.1.1 构造函数

本构造函数返回 BleWiFiClient 类的实例。

原型：

```
public BleWiFiClient(Context context, BluetoothDevice device);
```

参数：

context：应用上下文 Context；

device：要配置的设备；

#### 2.1.2 connect 方法

本方法建立 client 与 Device 之间的连接，若连接建立成功，BleWiFiCallback 回调 onConnected 方法将被调用。client 将主动扫描 Device 的服务和特征，在发现指定的服务和特征后，BleWiFiCallback 回调 onServicesDiscovered 方法将被调用，此时用户才可

以开始配网过程。

原型：

```
public synchronized void connect();
```

### 2.1.3 negotiateSecretKey 方法

本方法用来与 Device 协商加密密钥，协商结果通过 BleWiFiCallback 回调 onNegotiateSecretKeyResult 方法通知给用户。

原型：

```
public void negotiateSecretKey();
```

### 2.1.4 configureSta 方法

本方法配置 Device 作为 STA 的参数，配置成功后 Device 开始加网，并将加网结果通过 BleWiFiCallback 回调 onConfigureStaResult 方法通知给用户。

原型：

```
public void configureSta(final BleWiFiStaParams params);
```

参数：

params: 配置 STA 参数，包括 AP 的 SSID 和 password 等；

### 2.1.5 setBleWiFiCallback 方法

本方法用来设置 client 的 BleWiFiCallback 回调接口。

原型：

```
public void setBleWiFiCallback(BleWiFiCallback callback);
```

参数：

callback: BleWiFiCallback 回调接口实例；

### 2.1.6 close 方法

本方法用来释放 client 的资源。

原型：

```
public synchronized void close();
```

## 2.2 BleWiFiCallback 接口

本接口用来实现 BleWiFiClient 类的回调。用户的回调类需要继承本接口，并通过 BleWiFiClient 的 setBleWiFiCallback 方法将回调类的实例设置给 client，从而接收通知。

### 2.2.1 onConnected 方法

本方法用来通知用户蓝牙连接建立成功。

原型：

```
void onConnected(BleWiFiClient client);
```

参数：

client: BleWiFiClient 类的实例；

### 2.2.2 onDisconnected 方法

本方法用来通知用户蓝牙连接已断开。

原型：

```
void onDisconnected(BleWiFiClient client);
```

参数：

client: BleWiFiClient 类的实例；

### 2.2.3 onServicesDiscovered 方法

本方法用来通知用户蓝牙 GATT 服务和特征已经发现，用户可以在此回调方法中开始密钥交换过程。如果不需要加密传输配置参数，可以跳过密钥交换过程，在此回调方法中直接开始配网过程。

原型：

`void onServicesDiscovered(BleWiFiClient client);`

参数:

client: BleWiFiClient 类的实例;

#### 2.2.4 onConfigureStaResult 方法

本方法用来通知用户调用 client.configureSta 方法的配网结果。

原型:

`void onConfigureStaResult(BleWiFiClient client, BleWiFiConfigStaResult result);`

参数:

client: BleWiFiClient 类的实例;

result: 配网结果;

#### 2.2.5 onNegotiateSecretKeyResult 方法

本方法用来通知用户调用 client.negotiateSecretKey 方法的密钥交换结果。

原型:

`void onNegotiateSecretKeyResult(BleWiFiClient client, BleWiFiBaseResult result);`

参数:

client: BleWiFiClient 类的实例;

result: 密钥交换结果;

#### 2.2.6 onError 方法

本方法用来通知用户错误。

原型:

`void onError(BleWiFiClient client, int errCode);`

参数:



client: BleWiFiClient 类的实例;

errCode: 错误号;

## 2.3 BleWiFiStaParams 类

本类是 BleWiFiClient 类 configureSta 配网方法的参数类, 包含 SSID、password 和 BSSID 三个参数可以设置, 其中 SSID 和 BSSID 至少设置一个不为空。如果 AP 是 OPEN 模式, password 不设置。

### 2.3.1 setSsid 方法

本方法用来设置 AP 的 SSID。

原型:

```
public void setSsid(String ssid);
```

参数:

ssid: AP 的 SSID;

### 2.3.2 setPassword 方法

本方法用来设置 AP 的密码。

原型:

```
public void setPassword(String password);
```

参数:

password: AP 的密码;

### 2.3.3 setBssid 方法

本方法用来设置 AP 的 BSSID, 本方法适用于隐藏 SSID 的 AP 配网的情况, 可以通过设置 BSSID 和 Password 来配网。

原型:

```
public void setBssid(String bssid);
```

参数:

bssid: AP 的 BSSID;

## 2.4 BleWiFiBaseResult 类

本类是 BleWiFiCallback 回调接口 onNegotiateSecretKeyResult 方法的参数类。也是所有回调接口方法参数类的基类。

### 2.4.1 getStatus 方法

本方法用来获取回调接口方法的状态，通知用户结果。

原型:

```
public int getStatus();
```

返回值:

错误号，具体参见本类错误号定义。

### 2.4.2 错误号定义

本类定义了如下错误号，包含了 getStatus 方法返回的所有可能值，同时也包含了 BleWiFiCallback 回调接口 onError 方法所有的错误。

//成功

```
public static final int STATUS_SUCCESS = 0;
```

//参数错误

```
public static final int STATUS_INVALID_PARAMS = 1;
```

//密码错误

```
public static final int STATUS_PASSWORD = 2;
```

//获取 IP 地址失败

```
public static final int STATUS_DHCP_IP= 3;

//扫描失败

public static final int STATUS_WIFI_SCAN= 4;

//秘钥交换失败

public static final int STATUS_NEGOTIATE_SECRET_KEY=5;

//发送数据失败

public static final int STATUS_GATT_WRITE=6;
```

## 2.5 BleWiFiConfigStaResult 类

本类是 BleWiFiCallback 回调接口 onConfigureStaResult 方法的参数类。本类继承 BleWiFiBaseResult 类，除了 status 外，还有 mac 和 IPAddress 可以获取。

### 2.5.1 getMac 方法

本方法用来获取 Device 的 WiFi Mac 地址，配网后加网成功后返回。

原型：

```
public String getMac();
```

返回值：

Device 的 WiFi Mac 地址。

### 2.5.2 getIpAddress 方法

本方法用来获取 Device 的 IP 地址，配网后加网成功后返回。

原型：

```
public String getIpAddress();
```

返回值：

Device 的 IP 地址。

### 3 使用示例

#### 3.1 实例化 BleWiFiClient

```
mBleWiFiClient = new BleWiFiClient(getApplicationContext(), mDevice);
```

#### 3.2 设置 BleWiFiCallback

```
mBleWiFiClient.setBleWiFiCallback(new MyBleWifiCallback());
```

#### 3.3 连接 Device

```
mBleWiFiClient.connect();  
  
//连接成功在 BleWiFiCallback 回调接口中通知  
  
@Override  
public void onConnected(BleWiFiClient client) {  
    }  
  
//发现服务和特征在 BleWiFiCallback 回调接口中通知  
  
@Override  
public void onServicesDiscovered(BleWiFiClient client) {  
    //发现服务和特征成功  
    onGattServicesDiscovered();  
}
```

#### 3.4 与 Device 协商数据加密密钥

```
//发现服务和特征成功，开始密钥协商  
  
private void onGattServicesDiscovered() {  
    //开始与 Device 协商数据加密密钥
```

```
mBleWiFiClient.negotiateSecretKey();

}

//协商结果在 BleWiFiCallback 回调接口中通知

@Override

public void onNegotiateSecretKeyResult(BleWiFiClient client,
BleWiFiBaseResult result) {

    if(result.getStatus() == BleWiFiBaseResult.STATUS_SUCCESS){

        //协商成功

        onNegotiateSecretKeySuccess();

    }

    else{

    }

}

}
```

### 3.5 开始配网

//协商成功，开始加密传输配网参数

```
private void onNegotiateSecretKeySuccess(){

    runOnUiThread() -> {

        //构造 STA 配网参数

        BleWiFiStaParams params = new BleWiFiStaParams();

        params.setSsid(mTxtSsid.getText().toString());

        params.setPassword(mTxtPassword.getText().toString());

        //开始配网
```

```
mBleWiFiClient.configureSta(params);

});

}

//配网结果在 BleWiFiCallback 回调接口中通知

@Override

public void onConfigureStaResult(BleWiFiClient client, BleWiFiConfigStaResult
result) {

    if(result.getStatus() == BleWiFiBaseResult.STATUS_SUCCESS) {

        //配网成功，显示 WiFi 的 Mac 地址和 Device 的 IP 地址

        ShowMessage(String.format("Mac: %s", result.getMac()));

        ShowMessage(String.format("IP Address: %s",
result.getIpAddress()));

    }

    else{

    }

}

}
```