RAK IoT 模块配置工具移植说明

深圳市瑞科慧联科技有限公司

www.rakwireless.com

info@rakwireless.com

1. 概述

该配置工具主要依赖于库 iot.sdk.aar，方便用户快速上手使用 RAK 公司的 IoT WiFi 模块，包括：RAK413/RAK415/RAK423/RAK425/RAK473/RAK475/RAK476/RAK477，主要功能包括一键配置、AP 配置、参数配置、OTA 升级、透传通信等功能。

2. 本地发现移植说明

2.1 概述

这一部分主要实现当手机与模块处于同一网络时，手机可以获取到该模块的相关信息。

2.2 移植步骤

1.添加 iot.sdk.aar。

2.初始化本地发现接口：

```
Scanner _scanner=new Scanner(getApplicationContext());
```

3.监听扫描结果并获取相关信息：

```
_scanner.setOnScanOverListener(new Scanner.OnScanOverListener() {
    @Override
    public void onResult(Map<InetAddress, ScanInfo> data,InetAddress gatewayAddress) {
        if (data != null) {
            for (Map.Entry<InetAddress, ScanInfo> entry : data.entrySet()) {
                String ip = entry.getKey().getHostAddress();
                _scanInfo.Ip = ip;//获取模块的ip地址
                ScanInfo _scanInfo = entry.getValue();//获取模块的其他信息
                /* 模块信息包括：
                    _scanInfo.GroupName      //模块的组名称
                    _scanInfo.NickName       //模块名称
                    _scanInfo.Mac            //模块的MAC地址
                    _scanInfo.Rssi           //模块的信号值
                    _scanInfo.ProductId      //模块所属产品ID
                    _scanInfo.ClientId       //模块的ID
                */
            }
        }
    }
});
```

4.扫描设备，有以下三种方式：

```
_scanner.scanAll(); //扫描同一局域网下全部设备
_scanner.scan(); //扫描同一局域网下单个设备(第一个回应手机的设备)
_scanner.scan(_ip); //指定ip地址扫描，_ip：模块的ip地址
```

3. AP 配置移植说明

3.1 概述

这一部分主要实现使用 AP 配置的方式把模块配置到指定的路由器，这种配置方式非常稳定可靠。

3.2 移植步骤

移植步骤：初始化>>>获取网络列表，选择其中一个网络>>>配置模块>>>复位模块。

1.添加 iot.sdk.aar。

2.初始化 AP 配置接口：

```
ApConfig _apSdkAPI=new ApConfig(_deviceIp,_devicePassword,_moduleType);
//_deviceIp：模块 ip    _devicePassword：模块密码   _moduleType：模块类型
```

3.设置 AP 配置监听：

```
_apSdkAPI.setOnResultListener(new ApConfig.OnResultListener() {
    @Override
    public void onResult(ApConfig.Response result) {
        if (result.type== ApConfig.GET_SSID_FROM_DEVICE){
            //获取网络列表返回的内容
            if (result.statusCode==200){
                ssidList=_apSdkAPI.decodeSsidFromDevice(result.body);//解析网络列表
                if (ssidList.size()>0){
                    rssiList=_apSdkAPI.getRssiList();//获取网络列表的信号值
                    //显示获取到的网络列表
                }
            }
        }
        else if (result.type== ApConfig.CONFIGURE_DEVICE_TO_NETWORK){
            //配置模块联网返回的内容
            if (result.statusCode==200){
                //配置完成
            }
        }
        else if (result.type== ApConfig.RESET_DEVICE){//配置模块联网返回的内容
            if (result.statusCode==200){//复位成功
            }
        }
    }
});
```

4.获取模块扫描到的网络列表

```
_apSdkAPI.getSsidFromDevice();
```

5.配置模块联网

```
_apSdkAPI.configureDeviceToNetwork(Ssid, Psk); //Ssid：无线网络名称   Psk：无线网络密码
```

6.复位模块

```
_apSdkAPI.resetDevice();
```

4. 一键配置移植说明

4.1 概述

这一部分主要实现如何利用手机 APP，使用一键配置功能快速把模块配置到指定的路由器。

4.2 移植步骤

1.添加库 iot.sdk.aar。

2.初始化一键配置接口：

```
EasyConfig _easyConfig = new EasyConfig(_ctx, _moduleType);
//_ctx：上下文 Context        _moduleType：模块类型
```

3.设置相关监听

```
//监听停止配置事件
_easyConfig.setOnStopListener(new EasyConfig.OnStopListener() {
    @Override
    public void onStop() {
        ......
    }
});

//监听配置进度
_easyConfig.setOnProgressListener(new EasyConfig.OnProgressListener() {
    @Override
    public void onData(final int progress, final String ip, final String mac) {
        ..... //progress：配置进度   ip：模块 ip   mac：模块 mac
              (当 ip 和 mac 不为 null 时即配置成功)
    }
});
```

4.开始配置

```
_easyConfig.start(Ssid, Psk); //Ssid：无线网络名称   Psk：无线网络密码
```

5.停止配置

```
_easyConfig.stop();
```

5. 参数配置移植说明

5.1 概述

这一部分实现配置模块的所有参数。

5.2 移植步骤

1.添加库 iot.sdk.aar。

2.初始化参数配置接口

```
ParametersSettings _parametersSettings=new ParametersSettings(_deviceIp,_moduleType);
//_deviceIp：模块 ip   _moduleType：模块类型
```

3.设置相关监听

```
_parametersSettings.setOnRecvDataListener(new ParametersSettings.OnRecvDataListener() {
    @Override
    public void onRecvData(final int statusCode, final String resultData, final int cmdType) {
        runOnUiThread(new Runnable() {
            @Override
            public void run() {
                if (statusCode==200){
                    if (cmdType==ParametersSettings.CERTIFICATE_CMD){
                            //模块认证返回的信息
                    }
                    else if(cmdType==ParametersSettings.GET_VERSION_CMD){
                            //获取模块的固件版本
                    }
                    else if(cmdType==ParametersSettings.GET_PARAMETERS_CMD){
                            //获取模块的配置参数
                    }
                    else if(cmdType==ParametersSettings.SET_PARAMETERS_CMD){
                            //设置模块的配置参数
                    }
                    else if(cmdType==ParametersSettings.RESET_CMD){
                            //复位模块返回的信息
                    }
                    else if(cmdType==ParametersSettings.FAC_RESET_CMD){
                            //模块恢复出厂返回的信息
                    }
                    else if(cmdType==ParametersSettings.GET_SSID_FROM_DEVICE){
                        //解析网络列表
                        ssidList=_parametersSettings.decodeSsidFromDevice(resultData);
                        if (ssidList.size()>0){
                            //获取网络列表的信号值
                           rssiList=_parametersSettings.getRssiFromDevice();
                            //显示获取到的网络列表
```

```
                                    }
                                }
                            }
                        }
                    });
                }});
```

4.模块认证

```
_parametersSettings.Certificate(Username, Password); //Username：模块用户名 Password：模块密码
```

5.获取模块版本号

```
_parametersSettings.GetVersion();
```

6.获取模块配置参数

```
_parametersSettings.GetParameters();
```

7.解析模块配置参数

```
ModuleParameters.DecodeModuleParameters(parameters);
```

解析完的模块参数保存到 ModuleParameters 类里，所有参数如表 5-1 所示：

表 5-1 模块参数值说明

参数类型	参数名称	参数说明	参数值	备注
模式设置	power_mode_value	功耗模式	full：全功耗 save：节省功耗	
	wlan_mode_value	网络模式	STA：STA 模式 AP：AP 模式 AP+STA：共存	仅 RAK47X 支持共存模式
AP 参数设置	ap_ssid_value	AP 热点名称		
	ap_bdcast_value	AP 热点广播	0：关闭 1：开启	
	ap_max_clts_value	允许最大连接	取值：0~4 0 表示无限制	仅 RAK47X 支持多个连接
	ap_channel_value	AP 信道	取值：1~13	
	ap_sec_mode_value	AP 加密模式	0：不加密 1：加密	
	ap_psk_value	AP 密码		
	ap_ip_value	AP 的 IP 地址		
STA 参数设置	sta_ssid_value	STA 网络名称		
	sta_sec_mode_value	STA 加密模式	0：不加密 1：加密	
	sta_psk_value	STA 网络密码		
	sta_dhcp_value	DHCP 设置	0：不使能 1：使能	

	sta_ip_value	STA 时模块 IP		
	sta_netmask_value	STA 时模块掩码		
	sta_gateway_value	STA 时模块网关		
	sta_dns1_value	STA 时模块 dns1		
	sta_dns2_value	STA 时模块 dns2		
UART 参数设置	uart_baudrate_value	UART 波特率	9600~921600	
	uart_datalen_value	UART 数据位	5~8	RAK47X 只支持 7 和 8
	uart_parity_en_value	UART 校验位	0：无校验 1：奇校验 2：偶校验	
	uart_stoplen_value	UART 停止位	0：1 位停止位 1：2 位停止位	
	uart_rtscts_en_value	UART 流控	0：不使能 1：使能 2：使能 485 流控	RAK47X 不支持 485 流控
	uart_timeout_value	UART 字节超时		
	uart_recvlenout_value	UART 字节间隔		
SOCKET 参数设置	socket_multi_en_value	双 socket 设置	0：不使能 1：使能	
	socketA_type_value	socketA 类型	tcp：TCP 客户端 ltcp：TCP 服务器 udp：UDP 客户端 ludp：UDP 服务器	
	socketA_max_clts_value	socketA 最大连接数	取值：0~4	仅 RAK47X 支持
	socketA_destip_value	socketA 目标 ip		
	socketA_destport_value	socketA 目标端口		
	socketA_localport_value	socketA 本地端口		
	socketA_tcp_timeout_value	socketA TCP 超时时间		
	socketA_tcp_reconval_value	socketA TCP 重连时间间隔		仅 RAK47X 支持
	socketB_type_value	socketB 类型	tcp：TCP 客户端 ltcp：TCP 服务器 udp：UDP 客户端 ludp：UDP 服务器	
	socketB_max_clts_value	socketB 最大连接数	取值：0~4	仅 RAK47X 支持
	socketB_destip_value	socketB 目标 ip		
	socketB_destport_value	socketB 目标端口		

	socketB_localport_value	socketB 本地端口		
	socketB_tcp_timeout_value	socketB TCP 超时时间		
	socketB_tcp_reconval_value	sockeB TCP 重连时间间隔		仅 RAK47X 支持
用户参数设置	user_name_value	模块用户名		
	user_password_value	模块用户密码		
	module_name_value	模块名称		
	module_group_value	模块组名称		

8.设置模块配置参数

(1) _parametersSettings.SetParameters();//设置模块所有参数

(2) _parametersSettings.SetParameters(_type);//设置模块指定类型的参数

类型包括：

ParametersSettings.SET_PARAMETERS_MODE //设置模式相关参数
ParametersSettings.SET_PARAMETERS_AP //设置 AP 相关参数
ParametersSettings.SET_PARAMETERS_STA //设置 STA 相关参数
ParametersSettings.SET_PARAMETERS_UART //设置 UART 相关参数
ParametersSettings.SET_PARAMETERS_SOCKET //设置 SOCKET 相关参数
ParametersSettings.SET_PARAMETERS_USER //设置用户相关参数

注意：以上两种参数设置方式配置前需要先获取模块的配置参数，然后对表 5-1 中需要修改的参数进行修改，然后选择调用上面两个接口，即可完成配置。

(3) _parametersSettings.SetParameters(_config);//设置模块任意参数 _config：配置信息

这个接口需要用户自己拼接配置参数，拼接格式如下：

关键字=值&关键字=值&关键字=值&关键字=值&关键字=值.....

wlan_mode=0&power_mode=0&ap_ssid=RAK477_AP&ap_psk=&ap_bdcast=1&ap_sec_mode=0&ap_max_clts=3&ap_channel=6&.....

RAK IoT 模块的关键字信息在 ModuleParameters 类里已有定义，可以直接引用，如表 5-2：

表 5-2 模块关键字说明

关键字类型	关键字名称	关键字说明	关键字的值	备注
模式设置	power_mode	功耗模式	RAK47X： 0：全功耗 1：节省功耗 RAK41X、42X： full：全功耗 save：节省功耗	
	wlan_mode	RAK47X 网络模式	0：STA 模式 1：AP 模式	

			2：AP+STA 共存	
	wifi_mode	RAK41X、42X 网络模式	STA：STA 模式 AP：AP 模式	
AP 参数设置	ap_ssid	AP 热点名称		
	ap_bdcast	RAK47X： AP 热点广播	0：关闭 1：开启	
	ap_bdcast_en	RAK41X、42X：AP 热点广播	0：关闭 1：开启	
	ap_max_clts	允许最大连接	取值：0~4 0 表示无限制	仅 RAK47X 支持多个连接
	ap_channel	AP 信道	取值：1~13	
	ap_sec_mode	RAK47X： AP 加密模式	0：不加密 1：加密	
	ap_secu_en	RAK41X、42X： AP 加密模式	0：不加密 1：加密	
	ap_psk	AP 密码		
	ap_ip	RAK47X： AP 的 IP 地址		
	ap_ipaddr	RAK41X、42X： AP 的 IP 地址		
STA 参数设置	sta_ssid	STA 网络名称		
	sta_sec_mode	RAK47X：STA 加密模式	0：不加密 1：加密	
	sta_secu_en	RAK41X、42X：STA 加密模式	0：不加密 1：加密	
	sta_psk	STA 网络密码		
	sta_dhcp	RAK47X：DHCP 设置	0：不使能 1：使能	
	sta_dhcp_en	RAK41X 、 42X ： DHCP 设置	0：不使能 1：使能	
	sta_ip	RAK47X：STA 时模块 IP		
	sta_ipaddr	RAK41X、42X：STA 时模块 IP		
	sta_netmask	STA 时模块掩码		
	sta_gateway	STA 时模块网关		
	sta_dns1	RAK47X：STA 时模块 dns1		
	sta_dnssever1	RAK41X、42X：STA 时模块 dns1		

	sta_dns2	RAK47X：STA 时模块 dns2		
	sta_dnssever2	RAK41X、42X：STA 时模块 dns2		
UART 参数设置	uart_baudrate	UART 波特率	9600~921600	
	uart_datalen	UART 数据位	5~8	RAK47X 只支持 7 和 8
	uart_parity_en	UART 校验位	0：无校验 1：奇校验 2：偶校验	
	uart_stoplen	UART 停止位	0：1 位停止位 1：2 位停止位	
	uart_rtscts_en	UART 流控	0：不使能 1：使能 2：使能 485 流控	RAK47X 不支持 485 流控
	uart_timeout	UART 字节超时		
	uart_recvlenout	UART 字节间隔		
SOCKET 参数设置	socket_multi_en	双 socket 设置	0：不使能 1：使能	
	socketA_type	socketA 类型	RAK47X： 0：TCP 客户端 1：TCP 服务器 2：UDP 客户端 3：UDP 服务器 RAK41X、42X： tcp：TCP 客户端 ltcp：TCP 服务器 udp：UDP 客户端 ludp：UDP 服务器	
	socketA_max_clts	socketA 最大连接数	取值：0~4	仅 RAK47X 支持
	socketA_destip	socketA 目标 ip		
	socketA_destport	socketA 目标端口		
	socketA_localport	socketA 本地端口		
	socketA_tcp_timeout	socketA TCP 超时时间		
	socketA_tcp_reconval	socketA TCP 重连时间间隔		仅 RAK47X 支持
	socketB_type	socketB 类型	RAK47X： 0：TCP 客户端	

			1：TCP 服务器 2：UDP 客户端 3：UDP 服务器 RAK41X、42X： tcp：TCP 客户端 ltcp：TCP 服务器 udp：UDP 客户端 ludp：UDP 服务器	
	socketB_max_clts	socketB 最大连接数	取值：0~4	仅 RAK47X 支持
	socketB_destip	socketB 目标 ip		
	socketB_destport	socketB 目标端口		
	socketB_localport	socketB 本地端口		
	socketB_tcp_timeout	socketB TCP 超时时间		
	socketB_tcp_reconval	sockeB TCP 重连时间间隔		仅 RAK47X 支持
用户参数设置	user_name	模块用户名		
	user_password	模块用户密码		
	module_name	模块名称		
	module_group	模块组名称		

注意：参数配置完成后不会立即生效，需要复位模块才生效。调用复位接口或者硬件复位均可。

9.复位模块

```
_parametersSettings.Reset();
```

10.模块恢复出厂设置

```
_parametersSettings.FACReset();
```

11.获取模块扫描到的网络列表

```
_parametersSettings.getSsidFromDevice();
```

12.解析模块扫描到的网络列表

```
ArrayList<String> ssidList=_parametersSettings.decodeSsidFromDevice(resultData);
//resultData：获取到的网络信息  ssidList：解析后的网络列表
```

13.关闭配置，释放配置接口

```
_parametersSettings.Close();
```

6. 透传移植说明

6.1 概述

这一部分主要实现模块通过 Socket 透传收发数据的功能。

6.2 移植步骤

1.添加库 iot.sdk.aar。

2.初始化升级接口

(1) SocketControl _socketControl=new SocketControl(_destIp,_destPort,_localPort, _type);

(2) SocketControl _socketControl=new SocketControl(_destIp,_destPort, _type);

(3) SocketControl _socketControl=new SocketControl(_localPort, _type);

//_destIp：目标 ip　_destPort：目标端口　_localPort：本地端口　_type：Socket 类型

Socket 类型:

```
SocketControl.TCP      //TCP 客户端
SocketControl.LTCP     //TCP 服务器
SocketControl.UDP      //UDP 客户端
SocketControl.LUDP     //UDP 服务器
```

3.Socket 连接

```
//设置 Socket 连接监听
_socketControl.setOnConnectStateListener(new SocketControl.OnConnectStateListener() {
    @Override
    public void OnConnectState(final boolean connect) {
        //connect：true 表示 Socket 连接成功　false 表示 Socket 连接失败
    }
});
_socketControl.SocketConnect();//Socket 连接
```

4.Socket 发送数据

```
//设置 Socket 发送监听
_socketControl.setOnSendDataStateListener(new SocketControl.OnSendDataStateListener() {
    @Override
    public void OnSendDataState(final boolean send) {
        //send：true 表示发送成功　false 表示 Socket 发送失败
    }
    });
_socketControl.SocketSend(data, offset, length);//Socket 发送字节数组
_socketControl.SocketSend(str);//Socket 发送字符串数据
```

5.Socket 接收数据

```
//设置 Socket 接收 TCP 数据监听
_socketControl.setOnRecvTCPDataListener(new SocketControl.OnRecvTCPDataListener() {
    @Override
    public void onRecvTCPData(final byte[] data, final int size) {
        //TCP 数据
    }
});

//设置 Socket 接收 UDP 数据监听
_socketControl.setOnRecvUDPDataListener(new SocketControl.OnRecvUDPDataListener() {
    @Override
    public void onRecvUDPData(final byte[] data, final int size) {
        //UDP 数据
    }
});
```

6.关闭 Socket

```
_socketControl.SocketClose();
```

7. OTA 移植说明

7.1 概述

这一部分主要实现无线升级固件的功能。

7.2 移植步骤

整个升级过程如下：

创建文件夹存放要升级的固件 >> 手机扫描设备，选择一个设备升级 >> 获取模块和手机中存放固件的版本号 >>建立连接并开始升级 >>显示升级结果

1.添加库 iot.sdk.aar。

2.初始化升级接口

```
OTAInterface _otaInterface=new OTAInterface(_deviceIp,_username,_password,_moduleType);
//_deviceIp：模块 ip    _username：模块用户名   _password：模块密码   _moduleType：模块类型
```

3.设置相关监听

```
_otaInterface.setOnResultaListener(new OTAInterface.OnResultListener() {
            @Override
            public void onResult(final int percent, final int resultData) {
                //percent：升级进度
                runOnUiThread(new Runnable() {
                    @Override
                    public void run() {
                        if (resultData==OTAInterface.OTA_CONNECT_FAILED){
                            //连接失败
                        }
                        else if (resultData==OTAInterface.OTA_RESPONSE_FAILED){
                            //升级失败
                        }
                        else if (resultData==OTAInterface.OTA_UPGRADE_SUCCESS){
                            //升级成功
                        }
                    }
                });
            }
        });
```

4.开始升级

```
_otaInterface.upgradeFirmwareToDevice(_firmwarePath);//firmwarePath：升级固件的路径
```

5.停止升级

```
_otaInterface.stopUpgradeFirmware();
```

8. 相关权限

8.1 相关权限

```
  <!-- wifi usage -->
      <uses-permission android:name="android.permission.CHANGE_WIFI_MULTICAST_STATE"/>
      <uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"></uses-permission>
      <uses-permission android:name="android.permission.CHANGE_WIFI_STATE"></uses-permission>
      <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"></uses-permission>
      <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"></uses-permission>
      <uses-permission android:name="android.permission.INTERNET" ></uses-permission>
     <uses-permission android:name="android.permission.READ_SMS"></uses-permission>
 <!-- Camera usage -->
     <uses-permission android:name="android.permission.VIBRATE" />
      <uses-permission android:name="android.permission.CAMERA" />
      <uses-permission android:name="android.permission.FLASHLIGHT" />
      <uses-feature android:name="android.hardware.camera" />
      <uses-feature android:name="android.hardware.camera.autofocus" />
<!-- file and SD Card usage -->
      <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
      <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
      <uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
```

9. 修改记录

版本	作者	时间	修改内容
V1.0	瞿瑾	2017/02/14	创建文档
V1.0.1	瞿瑾	2017/02/20	添加获取扫描网络列表的信号值