

# RakVideo 视频测试工具

## 移植说明 V1.4

深圳市瑞科慧联科技有限公司

[www.rakwireless.com](http://www.rakwireless.com)

[info@rakwireless.com](mailto:info@rakwireless.com)

© 2015 瑞科慧联对于此文件保留所有权利。

本文所提及的实际公司和产品名称，均为其各自所有者商标。

本文档的任何部分不得转载，不得存储在任何检索系统，

或以任何未经过瑞科慧联书面同意的形式传送。

本文件在更新新版本后，恕不另行通知。

## 1. 简介

### 1.1 概述

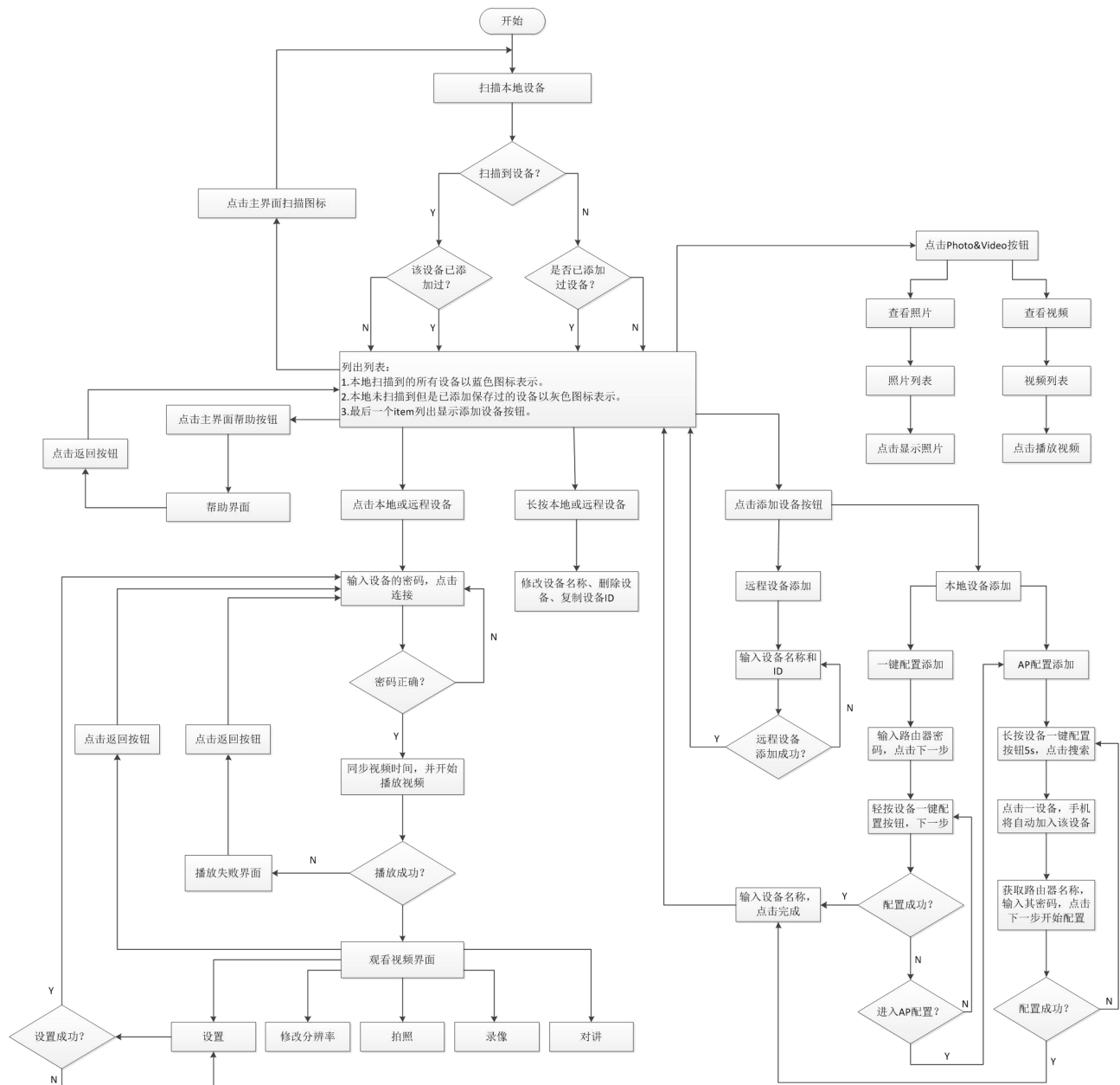
RakVideo 实现以下功能：

- 1) 本地直连观看视频。
- 2) 通过 AP 配置和一键配置将模块配置到路由器。
- 3) 保存、修改、删除已经配置添加的模块。
- 4) 手机本地观看视频或者手机打开数据流量远程观看视频。
- 5) 观看视频界面实现拍照、录像、对讲、分辨率切换等功能。
- 6) 查看已拍下的照片和录制的视频。
- 7) 中英文自动切换功能（手机系统为中文则中文显示，否则为英文显示）。

V1.4

## 1.2 程序整体设计流程

RakVideo 程序整体设计流程如下：



## 2. 移植说明

### 2.1 概述

程序移植主要包括一键配置、AP 配置、本地观看视频和远程观看视频等。

若是移植 Android Studio 工程则依赖于 wiseview.sdk.aar、jason-all.jar、libnabto\_client\_api\_jni.so 几个库和 nabto 资源文件。

若是移植 Eclipse 工程则依赖于 wiseview.sdk.jar、libdemo.sdk.so、jason-all.jar、libnabto\_client\_api\_jni.so 几个库和 nabto 资源文件。

Android Studio 工程中的 wiseview.sdk.aar 相当于 Eclipse 工程中的 wiseview.sdk.jar、libdemo.sdk.so。

下面的介绍都以 Android Studio 移植为例子。

### 2.2 一键配置移植

Easyconfig 配置是一种将设备快速加入路由器的配置方式，依赖于 wiseview.sdk.aar。

使用方法如下：

1. EasyConfig \_easyConfig = new EasyConfig(); //定义一个类对象，用于调用 EasyConfig 相关接口

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

```
3. _easyConfig.setOnProgressListener(new EasyConfig.OnProgressListener() {  
    @Override  
    public void onData(int progress, String ip, String mac,String id) {  
        ..... //监听配置过程中数据返回，若配置成功返回模块的 ip、mac、id 信息  
    }  
});
```

4. \_easyConfig.start(\_password);//开始配置，传入参数为路由器的密码

5. \_easyConfig.stop();//停止配置

具体请参考工程 RakVideo 工程 AddDeviceStep3.java。

## 2.3 AP 配置移植

AP 配置是一种设备建立热点，手机加入设备发送配置信息将设备配置到路由器的配置方式，依赖于 wiseview.sdk.aar、WLANAPI.java 和 LX520.java。

使用方法如下：

1. lx520.Get\_Ssid\_List();//获取设备获取到的网络列表
2. Scanner \_scanner = new Scanner(AddDeviceStep3.this); //定义一个类，用于调用 Scanner 相关接口  
\_scanner.setOnScanOverListener(new Scanner.OnScanOverListener() {  
    @Override  
    public void onResult(Map<InetAddress, String> data, InetAddress gatewayAddress) {  
        ..... //监听扫描完成事件，若发现设备即可再次获得设备信息，这里主要是为了记录正在配置的这个设备的 id  
    }  
});  
\_scanner.scanAll(); //开始扫描
3. lx520.joinWifi(\_ssid, \_password); //配置设备加入路由器，传入路由器的名称和密码
4. mWifiAdmin.addNetWork(mWifiAdmin.CreateWifiInfo(\_ssid, \_password, 1)); //手机自动切换到刚才配置的路由器，传入路由器的名称、密码和加密方式
5. 再次调用本地扫描 Scanner，找到刚才配置的设备。

具体请参考工程 RakVideo 工程 AddDeviceStep2AP.java 和 AddDeviceStep3.java。

## 2.4 视频播放移植

视频播放部分主要是将音视频数据流解码显示的过程，依赖于 wiseview.sdk.aar。

使用方法如下：

### 1. 设置对接模块相关参数

```
Module _module = new Module(this);
_module.setLogLevel(Enums.LogLevel.VERBOSE); //设置log打印方式
_module.setUsername("admin"); //设置用户名
_module.setPassword("admin"); //设置密码
_module.setPlayerPort(554); //设置播放视频的目标端口，本地播放端口为554，远程时为映射端口
_module.setModuleIp(_moduleIp); //设置播放视频目标 IP，本地播放 ip 为设备的 ip，远程时为
“127.0.0.1”
```

### 2. 设置播放视频相关参数

```
Player _player = _module.getPlayer();
_player.setTimeout(10000); //设置超时时间，单位：ms
_player.setRecordFrameRate(10); //设置录制视频的帧率
_player.setAudioOutput(false); //设置开启或关闭声音
_player.setDisplayView(context, _displayView, _displayView2, _viewType); //设置播放画布
参数描述：
(1) _displayView、_displayView2 //显示视频的画布
(2) _viewType //为0：SurfaceView(软解) 1：TextureView0：(软解) 2：SurfaceView(硬解)
```

使用说明：

(1) 若使用单屏显示时，将\_displayView、\_displayView2其中一个设置为null即可。如：

```
_player.setDisplayView(context, null, _displayView2, 0);
_player.setDisplayView(context, _displayView, null, 0);
_player.setDisplayView(context, null, _displayView2, 1);
_player.setDisplayView(context, _displayView, null, 1);
_player.setDisplayView(context, null, _displayView2, 2);
_player.setDisplayView(context, _displayView, null, 2);
```

(2) 若使用TextureView，则可以通过下面方法获取该TextureView，再做相应的变换。

//获取\_displayView对应的TextureView

```
TextureView _textureView = _displayView.getGLTextureView();
if(_textureView != null){
    _textureView.setRotation(45.0f); //顺时针旋转45°
}
```

//获取\_displayView2对应的TextureView2

```
TextureView _textureView2 = _displayView2.getGLTextureView2();
if(_textureView2 != null){
    _textureView2.setRotation(-45.0f); //逆时针旋转45°
}
```

V1.4

**PS:** *SurfaceView*的工作方式是创建一个置于应用窗口之后的新窗口, 这种方式的效率非常高, 因为*SurfaceView* 窗口刷新的时候不需要重绘应用程序的窗口, 但是*SurfaceView*也有一些非常不便的限制。因为*SurfaceView*的内容不在应用窗口上, 所以不能使用变换(平移、缩放、旋转等), 也难以放在*ListView*或者*ScrollView*中, 不能使用UI控件的一些特性比如设置透明度*View.setAlpha()*。为了解决这个问题 *Android 4.0*中引入了*TextureView*。

*SurfaceView*(硬解)具有更优的解码效率, 需*Android 4.1.2*以上才支持。

`_player.getState();`//获取视频播放状态

状态描述:

- (1) *Enums.State.IDLE*//空闲状态
- (2) *Enums.State.PLAYING*//正在播放
- (3) *Enums.State.PREPARING*//准备播放
- (4) *Enums.State.STOPPED*//已停止播放

`_player.play(_pipe, Enums.Transport.UDP);`//通过UDP获取视频并播放

`_player.play(_pipe, Enums.Transport.TCP);`//通过TCP获取视频并播放

`_player.stop();`//停止播放视频

`boolean _recording = _player.isRecording();`//判断是否正在录制视频

`_player.beginRecord(String path, String name);`//采用mp4v2录制, 不占内存, 录制稳定性一般

`_player.beginRecord0(String path, String name);`//采用ffmpeg录制, 占内存, 录制稳定性好

`_player.endRecord();`//结束录制

`Bitmap photo = _player.takePhoto();`//拍照

`_player.setOnStateChangedListener();`//监听播放视频时状态变化

`_player.setOnRecordStateChangedListener();`//监听录制视频时状态变化

`_player.setOnTimeoutListener();`//监听播放视频超时

### 3. 设置默认视频分辨率

`Controller _controller = _module.getController();`

`_player.changePipe(_pipe);`//设置手机获取视频的分辨率

*\_pipe* 参数描述:

- (1) `_pipe = Enums.Pipe.H264_PRIMARY`//设置手机获取视频为 H264 格式, 高清
- (2) `_pipe = Enums.Pipe.H264_SECONDARY`//设置手机获取视频为 H264 格式, 标清
- (3) `_pipe = Enums.Pipe.MJPEG_PRIMARY`//设置手机获取视频为 MJPEG 格式, 高清
- (4) `_pipe = Enums.Pipe.MJPEG_SECONDARY`//设置手机获取视频为 MJPEG 格式, 标清

`_player.setOnVideoSizeChangedListener();`//监听播放视频时尺寸变化

### 4. 设置播放视频画布

`DisplayView _displayView;`//设置播放视频画布

`_displayView = (DisplayView)findViewById(R.id.sview);`

`_displayView.setFullScreen(true);`//设置视频充满画布

布局文件如下：

```
<com.demo.sdk.DisplayView
    android:id="@+id/video_view"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
/>
```

#### 5. 获取视频 YUV 数据

```
_player.startGetYUVData(true); //使能获取视频 YUV 数据
_player.setOnGetYUVDataListener(new Player.OnGetYUVDataListener() {
    @Override
    public void onResult(int width, int height, byte[] yData, byte[] uData, byte[] vData) {
        //监听获取视频 YUV 数据
    }
});
```

具体请参考工程 RakVideo 工程 VideoPlay.java。



V1.4

## 2.5 远程 nabto 移植

远程 nabto 部分用于远程通道打通, 实现远程播放视频, 依赖于 libnabto\_client\_api\_jni.so、com.nabto.api 包中所有文件和 assets 下 nabto 资源文件。

1. RemoteTunnel \_remoteTunnel=new RemoteTunnel(getApplicationContext());
2. \_remoteTunnel.openTunnel(0,(getApplicationContext(), 5555, 554, \_deviceId); //5555:映射视频播放端口号; 554:视频默认端口 ; \_deviceId:设备id
3. \_remoteTunnel.setOnResultListener(new OnResultListener() //监听  
{  
    @Override  
    public void onResult(int id, String result)  
    {  
        // TODO Auto-generated method stub  
        if(result.equals("CONNECT\_TIMEOUT"))|| //远程连接超时  
        result.equals("NTCS\_CLOSED"))|| //远程设备不在线  
        result.equals("NTCS\_UNKNOWN"))|| //远程连接出现未知错误  
        result.equals("FAILED")) //远程连接失败  
        {  
        }  
        else  
        {  
            //远程连接成功, 映射后的IP为“127.0.0.1”, 端口号为“5555”  
        }  
    }  
});
4. \_remoteTunnel.openTunnel(0,(getApplicationContext(), 3333 80, \_deviceId); //3333:映射控制端口号;  
80:控制默认端口 ; \_deviceId:设备id
5. \_remoteTunnel.closeTunnels(); //关闭远程连接

### 6. 注意:

本地时: 目标 ip 为模块的 ip, 视频播放端口为 554, 控制端口为 80。

远程时: 目标 ip 为“127.0.0.1”, 视频播放端口为远程连接时对 554 映射后的端口, 控制端口为远程连接时对 80 映射后的端口。

具体请参考工程 RakVideo 工程 VideoPlay.java, DeviceUart.java。

V1.4

## 2.6 视频回放移植

视频回放实现下载播放视频模块录制到 TF 卡中的视频文件。

1. 获取 TF 卡中视频文件夹列表

```
Lx520 lx520=new Lx520(_ip+":_"+controlPort,_psk);
lx520.setOnResultListener(new Lx520.OnResultListener() {
    @Override
    public void onResult(Lx520.Response result) {
        if(result.statusCode==200){
            if(result.type==19){
                //TF 卡中视频文件夹列表
            }
        }
    }
});
lx520.Get_Video_Folder_List();
```

2. 获取其中一个文件夹中的视频列表

```
Lx520 lx520=new Lx520(_ip+":_"+controlPort,_psk);
lx520.setOnResultListener(new Lx520.OnResultListener() {
    @Override
    public void onResult(Lx520.Response result) {
        if(result.statusCode==200){
            if(result.type==20){
                //TF 卡中其中一个文件夹中的视频列表
            }
        }
    }
});
lx520.Get_Video_List(path); //path 为 TF 卡中任意文件夹的路径
```

3. 根据获取到的视频文件夹和视频路径，播放视频

**Mp4Download.playMp4File**(url, \_psk, savePath, videoHandler);

参数描述:

**url**: 回放视频的路径, 例如: [http://admin:admin@192.168.100.1/link//mnt/rec\\_folder/video/pipe0/1970Y01M04D15H/NVTDV19700104\\_150156.mp4](http://admin:admin@192.168.100.1/link//mnt/rec_folder/video/pipe0/1970Y01M04D15H/NVTDV19700104_150156.mp4)

**savePath**: 将会放的视频保存到手机指定的路径。

**videoHandler**: 视频回放时的相关状态返回。

4. 注意:

\_psk 为模块密码，默认是 admin。

本地时: \_ip 为模块的 ip, controlPort 为 80。

远程时: \_ip 为 “127. 0. 0. 1”， controlPort 为远程连接时对 80 映射后的端口。

具体请参考 RakVideo 工程 PlayBackVideoActivity.java,  
PlayBackFolderListActivity.java, PlayBackVideoListActivity.java。

V1.4

## 2.7 透传移植

透传部分主要实现手机与模块实时通信的功能。

RAK 的产品有些模块透传是通过建立 TCP 连接，目标端口号为 80，如 LX520 模块；有些模块是通过建立 UDP 连接，目标端口号为 1008，如 RAK566，具体见对应产品的规格书等文档。

### 1. TCP 透传

#### (1) 创建 TCP 连接

```
Socket _socket = new Socket(_deviceIp, _sendPort);  
_socket.setKeepAlive(true);  
dataStream = new DataOutputStream(_socket.getOutputStream());
```

#### (2) TCP 发送数据

```
dataStream.write(message);
```

#### (3) TCP 接收数据

```
_socket.getInputStream().read(buffer);
```

#### (4) 关闭 TCP 连接

```
_socket.close();  
dataStream.close();
```

### 2. UDP 透传

#### (1) 创建 UDP 连接

```
DatagramSocket udp_socket = new DatagramSocket(25000);
```

#### (2) UDP 发送数据

```
InetAddress serverAddress = InetAddress.getByName(_deviceIp);  
DatagramPacket sendPackage = new DatagramPacket(data, data.length, serverAddress,  
_sendPort);  
udp_socket.send(sendPackage);
```

#### (3) UDP 接收数据

```
DatagramPacket recvPackage = new DatagramPacket(buffer, buffer.length);  
udp_socket.receive(recvPackage);
```

#### (4) 关闭 UDP 连接

```
udp_socket.close();
```

### 3. 注意：

发送数据均以 0x01 0x55 开头，接收到的数据模块内部会自动添加 0x01 0x55。即：

发送数据时：0x01 0x55 [要发送的数据内容](#)

接收数据时：0x01 0x55 [要接收的数据内容](#)

本地时：\_deviceIp 为模块的 ip，\_sendPort 为 80。

远程时：\_deviceIp 为“127.0.0.1”，\_sendPort 为远程连接时对 80 映射后的端口。

具体请参考工程 RakVideo 工程 VideoPlay.java，DeviceUart.java。

### 3. 相关权限

RakVideo视频测试工具需要用到的权限如下：

```
<uses-permission android:name="android.permission.CHANGE_WIFI_MULTICAST_STATE"></uses-permission>
<uses-permission android:name="android.permission.INTERNET"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"></uses-permission>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"></uses-permission>
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"></uses-permission>
<uses-permission android:name="android.permission.WAKE_LOCK"></uses-permission>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"></uses-permission>
<uses-permission android:name="android.permission.CALL_PHONE"></uses-permission>
<uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"></uses-permission>
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="com.google.android.maps"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_LOCATION"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"></uses-permission>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"></uses-permission>
<uses-permission android:name="android.permission.DISABLE_KEYGUARD"></uses-permission>
<uses-permission android:name="android.permission.RESTART_PACKAGES" />
<uses-permission android:name="android.permission.KILL_BACKGROUND_PROCESSES" />
<uses-permission android:name="android.permission.CHANGE_CONFIGURATION" />
<uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS" />
<uses-permission android:name="android.permission.GET_TASKS" />
<uses-permission android:name="android.permission.BROADCAST_STICKY" />
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW" />
```

## 4. 修改记录

版本	作者	时间	修改内容
V1.0	瞿瑾	2016/03/05	创建文档
V1.1	瞿瑾	2016/05/17	1. 更新设置视频画布接口，可以单屏或双屏显示视频，并可以选择 SurfaceView 还是 TextureView 显示。 2. 添加获取视频 YUV 数据接口。
V1.2	瞿瑾	2016/08/05	1. 添加硬件解码。
V1.3	瞿瑾	2016/12/01	1.优化本地扫描。 2.保留 ffmpeg 和 mp4v2 两种录制方式。 3.添加图片处理功能。 4.规避播放 5275 闪退的问题。 5.添加视频回放功能。 6.添加透传功能。
V1.4	瞿瑾	2016/02/24	1.优化视频录制。