

RakVideoTest 视频测试工具

移植说明 V1.2

深圳市瑞科慧联科技有限公司

www.rakwireless.com

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

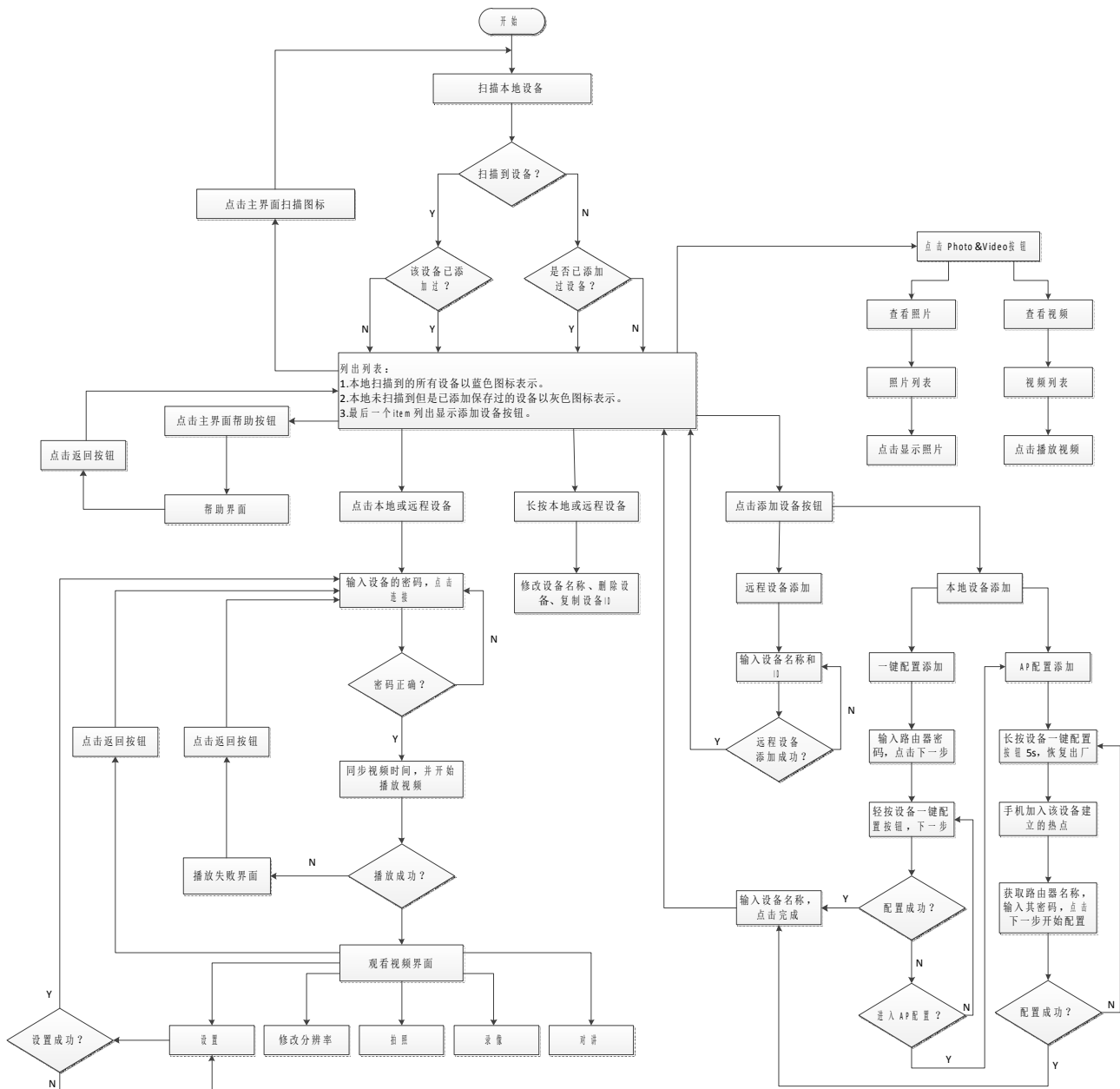
RakVideoTest 实现以下功能：

- 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）。

V1.2

1.2 程序整体设计流程

RakVideoTest 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。

依赖于 HTTP1、Audio、LX520、Easyconfig、3rdParty、common 等几个文件夹下的库和资源文件。

2.2 一键配置移植

Easyconfig 配置依赖于 EasyConfig 包。

主要有以下几个接口：

1) EasyConfig __EasyConfig = [[EasyConfig alloc] init:self];//初始化

2) - (void)SendDataWithPsk:(NSString *)psk andSSID:(NSString*)SSID;//开始配置，路由器 SSID 不隐藏时只要传入路由器密码即可，否则需要传入路由器密码和名称

3) - (void)stop_send;//停止配置

4) 设置好代理 EasyConfigDelegate 之后，用下面接口接收返回信息：

- (void)RecvWithPacket:(RecvPacket *)recvPacket;//配置成功后接收到的返回信息

recvPacket 中主要包括 recvPacket.module_name、recvPacket.module_mac、recvPacket.module_ip。

具体请参考工程 RakVideoTest 工程 AddDeviceStep3.m。

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 HTTP1 包和 3rdParty 包中的 deviceScan 包。使用方法如下：

```
1. +(HttpRequest*)HTTPRequestWithURL:(NSString*)HttpURL andData:(NSString*)PostData
    andMethod:(NSString*)Method andUserName:(NSString*)UserName
    andPassword:(NSString*)Password;
```

//通过 HTTP 请求去实现：1.获取模块扫描到的网络列表；2.配置模块；3.复位模块。

```
2. Lx52x_Device_Info *result = [_device_Scan_Find ScanDeviceWithTime:1.5f];//开始扫描1.5s
```

```
- (void)scanDeviceOver:(Lx52x_Device_Info *)result{
```

```
..... //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
```

```
}
```

具体请参考工程 RakVideoTest 工程 AddDeviceStep2AP.m。

V1.2

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 LX520 包。

主要接口如下：

1. 初始化视频播放界面

(1) 单屏播放

```
_videoView = [[LX520View alloc] initWithFrame:CGRectMake(0, 0, 1280, 720)];  
[_videoView set_log_level:4]; // 设置 log 打印方式  
[_videoView delegate:self]; // 设置代理  
[self.view addSubview:_videoView]; // 添加视频显示的 view
```

(2) 双屏播放

```
_videoView = [[LX520View alloc] initWithFrame2:frame:frame1:frame2]; // frame: 是 _videoView 整个的  
frame; frame1: _videoView 上第一个视频的 frame; frame2: _videoView 上第二个视频的 frame  
[_videoView set_log_level:4]; // 设置 log 打印方式  
[_videoView delegate:self]; // 设置代理  
[self.view addSubview:_videoView]; // 添加视频显示的 view
```

相关接口：

- (void)setView1Frame:(CGRect)frame; // 设置第一个视频的 frame
- (void)setView2Frame:(CGRect)frame; // 设置第二个视频的 frame
- (CGRect)getView1Frame; // 获取第一个视频的 frame
- (CGRect)getView2Frame; // 获取第二个视频的 frame
- (void)setView1Hidden:(BOOL)isHidden; // 设置第一个视频是隐藏
- (void)setView2Hidden:(BOOL)isHidden; // 设置第二个视频是隐藏

2. 设置视频播放参数

NSString *url ;

url = [NSString stringWithFormat:@"rtsp://admin:%@%@%:@%d/cam1/%@", _psk, _ip, _port, _pipe];

_psk: 设备连接密码

_ip: 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”

_port: 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

_pipe 参数描述：

- (1) _pipe = @"h264" // 设置手机获取视频为 H264 格式，高清
- (2) _pipe = @"h264-1" // 设置手机获取视频为 H264 格式，标清
- (3) _pipe = @"mpeg4" // 设置手机获取视频为 MJPEG 格式，高清

V1.2

(4) `_pipe =@"mpeg-1"`//设置手机获取视频为 MJPEG 格式，标清

```
[_videoView sound:NO];//设置开启或关闭声音
[_videoView set_record_frame_rate:10];//设置录制视频的帧率
[_videoView play:url useTcp:NO];//通过 UDP 或 TCP 获取视频并播放
[_videoView stop];//停止播放视频
```

3. 拍照与录像

```
[_videoView take_photo];//拍照
[_videoView begin_record:type];//开始录制,type: 0 ffmpeg 录制 1 mp4v2 录制
[_videoView begin_record2:type :path];//开始录制视频到指定的路径， path 为路径
[_videoView end_record];//结束录制
```

4. 监听视频播放状态

```
- (void)state_changed:(int)state {
    switch (state) {
        case 0: //空闲状态
        {
            break;
        }
        case 1: //准备播放
        {
            break;
        }
        case 2: ///正在播放
        {
            break;
        }
        case 3: //已停止播放
        {
            break;
        }
        default:
            break;
    }
}
```

V1.2

```
    }  
}  
  
- (void)video_info:(NSString *)codecName codecLongName:(NSString *)codecLongName {  
    //监听播放的视频信息  
}  
  
- (void)audio_info:(NSString *)codecName codecLongName:(NSString *)codecLongName  
    sampleRate:(int)sampleRate channels:(int)channels {  
    //监听播放的音频信息  
}
```

5. 获取视频解码后的 YUV 数据

```
- (void)startGetYUVData:(BOOL)start; //使能获取视频解码后的 YUV 数据  
- (void)GetYUVData:(int)width :(int)height  
    :(Byte*)yData :(Byte*)uData :(Byte*)vData  
    :(int)ySize :(int)uSize :(int)vSize; //监听视频解码后的 YUV 数据
```

具体请参考工程 RakVideoTest 工程 DevicePlay.m。

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 common 包和 3rdParty 包中的 nabto 包。

1. NabtoLibraryInit();//初始化nabto
2. Async_ConnectDeviceWithTunnel(&videoTunnel ,deviceId,554 ,5555);//videoTunnel: 视频通道，
5555:映射视频播放端口号； 554:视频默认端口 ； _deviceId:设备id
3. Async_ConnectDeviceWithTunnel(&httpTunnel ,deviceId,554 ,3333);//httpTunnel: 控制(透传)
通道， 3333:映射控制端口号； 80:控制默认端口 ； _deviceId:设备id
- 4.int status = CheckConnectStatus(&videoTunnel);//status为0表示远程连接成功，映射后的IP为
“127.0.0.1”，端口号为“5555”
int status = CheckConnectStatus(&httpTunnel);//status为0表示远程连接成功，映射后的IP为
“127.0.0.1”，端口号为“3333”
5. CloseTunnel(&videoTunnel);//关闭视频通道
CloseTunnel(&httpTunnel);//关闭控制(透传)通道
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 RakVideoTest 工程 DevicePlay.m。

V1.2

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

```
NSString *URL=[[NSString alloc]initWithFormat:@"http://%@:%d/param.cgi?
    action=list&group=videodir&fmt=link&pipe=0&type=0",_ip,controlPort];
HttpRequest* http_request = [HttpRequest HTTPRequestWithUrl:URL andData:nil
    andMethod:@"GET" andUserName:@"admin" andPassword:_psk];
if(http_request.StatusCode==200)
{
    //TF 卡中视频文件夹列表
}
```

2. 获取其中一个文件夹中的视频列表

```
NSString *URL=[[NSString alloc]initWithFormat:@"http://%@:%d/param.cgi?
    action=list&group=file&fmt=link&pipe=0&type=0&folder=%@",_ip,controlPort,folderName];
HttpRequest* http_request = [HttpRequest HTTPRequestWithUrl:URL andData:nil
    andMethod:@"GET" andUserName:@"admin" andPassword:_psk];
if(http_request.StatusCode==200)
{
    //TF 卡中其中一个文件夹中的视频列表
}
```

3. 根据获取到的视频文件夹和视频路径，播放视频

MPMoviePlayerController _moviePlayer=[[MPMoviePlayerController alloc]initWithContentURL:url];

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

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 RakVideoTest 工程 PlayBackFolderList.m, PlayBackVideoList.m, PlayBackViewController.m。

V1.2

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

```
GCDUartSocket = [[GCDAsyncSocket alloc] initWithDelegate:self  
delegateQueue:dispatch_get_main_queue()];  
[GCDUartSocket connectToHost:_deviceIp onPort:_sendPort error:nil];
```

(2) TCP 发送数据

```
[GCDUartSocket writeData:data withTimeout:1.0 tag:100];
```

(3) TCP 接收数据

```
[GCDUartSocket readDataWithTimeout:-1 tag:0];  
-(void)socket:(GCDAsyncSocket *)sock didReadData:(NSData *)data withTag:(long)tag {  
    if([sock isEqual:GCDUartSocket]){  
        //接收到的数据  
        [GCDUartSocket readDataWithTimeout:-1 tag:0];  
    }  
}
```

(4) 关闭 TCP 连接

```
if (GCDUartSocket != nil) {  
    [GCDUartSocket disconnect];  
    GCDUartSocket = nil;  
}
```

2. UDP 透传

(1) 创建 UDP 连接

```
GCDUdpSocket = [[GCDAsyncUdpSocket alloc] initWithDelegate:self  
delegateQueue:dispatch_get_main_queue()];  
[GCDUdpSocket bindToPort :25000 error:nil];
```

(2) UDP 发送数据

```
[GCDUdpSocket sendData:data toHost:_deviceIp port:_sendPort withTimeout:1.0 tag:100];
```

(3) UDP 接收数据

```
[GCDUdpSocket beginReceiving:&err];  
- (void)udpSocket:(GCDAsyncUdpSocket *)sock didReceiveData:(NSData *)data  
    fromAddress:(NSData *)address withFilterContext:(id)filterContext {  
    if([sock isEqual:GCDUdpSocket]){  
        //接收到的数据
```

V1.2

```
}  
}
```

(4) 关闭 UDP 连接

```
if (GCDUdpSocket != nil) {  
    [GCDUdpSocket close];  
    GCDUdpSocket = nil;  
}
```

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 RakVideoTest 工程 DevicePlay.m，DeviceUart.m。

3. 相关 Frameworks

RakVideoTest视频测试工具需要用到的Frameworks:

CoreGraphics.framework

AVFoundation.framework

CoreVideo.framework

Foundation.framework

UIKit.framework

CFNetwork.framework

SystemConfiguration.framework

OpenAL.framework

AssetsLibrary.framework

libbz2.tbd

libbz.tbd

libiconv.tbd

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	2016/07/07	1.添加分屏显示。 2.添加录像到指定路径。 3.添加获取解码后的 YUV 数据。
V1.2	瞿瑾	2016/12/02	1.保留 ffmpeg 和 mp4v2 两种录制方式。 2.规避播放 5275 闪退的问题。 3.添加视频回放功能。 4.添加透传功能。