WisView Video SDK Porting

User Guide V2.0

Shenzhen Rakwireless Technology Co., Ltd

www.rakwireless.com   info@rakwireless.com

Content

1. Brief Introduction

WisView Video SDK achieve the following functions：

1) Support local scan and get information of the video module.

2) Support AP Configuration ,configure the device to router.

3) Support watch video by local and remote open data traffic.

4) Support functions as take-photo,record,change pipe,intercom and so on.

5) Support setting various parameters of the video module.

6) Support VR，and watch the video with split screen.

7) Support software and hardware decoding.

8) Support decoding format:H264 and MJPEG.

2. Porting Instructions

2.1 Local Scan Porting

Scanner used to scan and get the reference information of the local module, it depends on:wisview.sdk.aar。

Instruction：

1. Initialize local scan interface.

```
Scanner _scanner = new Scanner(AddDeviceStep3.this); //used to call Scanner interfaces
```

2. Set On Scan Over Listener.

```
_scanner.setOnScanOverListener(new Scanner.OnScanOverListener() {
        @Override
        public void onResult(Map<InetAddress, String> data, InetAddress gatewayAddress) {
            // Listen to scan completed events，get device information once find the device
            if (data != null) {
                for (Map.Entry<InetAddress, String> entry : data.entrySet()) {
                    String id = entry.getValue(); //Scan the device ID
                    String ip = entry.getKey().getHostAddress(); //Scan the device ip address
                }
            }
        }
});
```

3. Start Local Scan.

```
_scanner.scanAll(); //start to scan
```

2.2 Porting ApConfig

ApConfig is the way let device to establish an AP, then mobile phone to the AP and send configuration information to the device, it's depends on wisview.sdk.aar and ParametersConfig.java。

Instruction：

1. Initialize parameter configuration interface

ParametersConfig _parametersConfig=new ParametersConfig(); //used to call Parameters Config interface

2. Set parameters to listen

```
_parametersConfig.setOnResultListener(new ParametersConfig.OnResultListener() {
    @Override
    public void onResult(ParametersConfig.Response result) {
        if(result.statusCode == 200) {
            if (result.type == ParametersConfig.GET_SSID_LIST) {
                //get network list of the device
            }
            else if (result.type == ParametersConfig.JOIN_WIFI) {
                //APConfig succeed
            }
        }
    }
});
```

3. _parametersConfig.getSsidList(); //Get network list of the device

4. _parametersConfig.joinWifi(_ssid, _password); // Configure the device and connect to a router,Incoming with the router's name and password.

s

2.3 Porting play video

Video play part is the process of decoding the audio and video data stream, it's depends on wisview.sdk.aar.

Instructions：

1. Set up the related parameters of docking module

　　　　Module _module_ = new Module(this);

　　　　module.setLogLevel(Enums.LogLevel.*VERBOSE*);// Set log print mode

　　　　module.setUsername("admin");// Set a user name

　　　　module.setPassword("admin");// Set a password

　　　　module.setPlayerPort(554);//Set target port of playing video, local port is 554，remote is the mapped

port.

　　　　　　module.setModuleIp(_moduleIp_);//Set target IP，the ip of local video is the device's ip，remote

　　　　　　ip is "127.0.0.1"

2.Set Video Play Reference data

　　　　Player _player_ = _module_.getPlayer();

　　　　player.setTimeout(10000);//Set timeout，unit：ms

　　　　player.setRecordFrameRate(10);//Set the frame rate when recording

　　　　player.setAudioOutput(false);//Set audio of open and close.

player.setDisplayView(context, _displayView, _displayView2, _viewType);//Set the decoding mode of playback

canvas.

　　　　Parameter Descriptions：

　　　　　　(1) _displayView、_displayView2//Show the canvas of the video

　　　　　　(2) _viewType//When it's 0：SurfaceView,　1：TextureView,　2：SurfaceView

　　　　Instructions：

　　　　(1) If it is single screen，set _displayView or _displayView2 to null. For example：

　　　　　　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) If it is TextureView，can get TextureView by following method，make corresponding changes.

　　　　//Get the corresponding TextureView of _displayView.

　　　　　　TextureView _textureView=_displayView.getGLTextureView();

```
if(_textureView!=null){
    _textureView.setRotation(45.0f);// Turn 45°clockwise
}
```

//Get the corresponding TextureView2 of _displayView2.

```
TextureView _textureView2=_displayView2.getGLTextureView2();
if(_textureView2!=null){
    _textureView2.setRotation(-45.0f);// Turn 45°counterclockwise
}
```

PS：*SurfaceView is to create a new window behind the application window. it is highly efficient ,because when refresh SurfaceView window without redraw the application window. But there are also some limitations about SurfaceView.The content of SurfaceView does not apply on the window，so that it can't transform（translation, zoom, rotation, etc.）, at the same time it's hard to put in ListView or ScrollView，Also can't set some characteristic with UI controls ,for example: Transparency View.setAlpha().In order to solve the problem Android 4.0 presents TextureView。*

SurfaceViewis with better decoding efficiency，supported by Android 4.1.2 or above.

_player.getState();//Get video play status

Status Description：

    (1) Enums.State.IDLE//Idle status

    (2) Enums.State.PLAYING//Playing

    (3) Enums.State.PREPARING//Ready for play

    (4) Enums.State.STOPPED//Stop playing the video

_player.play(_pipe, Enums.Transport.*UDP*);//Get and play the video through UDP,it's unavailable when remote play the video.

_player.play(_pipe, Enums.Transport.*TCP*);//Get and play the video through TCP，it's available when remote play the video

_player.stop();//Stop playing the video

boolean _recording = _player.isRecording();//Judge it is recording or not

_player.beginRecord(String path, String name);//Recording with mp4v2 format, does not occupy memory, record only with h264 format.

_player.beginRecord0(String path, String name);//Record with ffmpeg format,occupy memory,record with h264 and mjpeg format

_player.endRecord();//Finish recording

Bitmap photo = _player.takePhoto();//Take photos

_player.setOnVideoSizeChangedListener();//Monitor the size of the video when playing video

_player.setOnStateChangedListener();//Monitor the status of the video when playing video

_player.setOnRecordStateChangedListener();//Monitor the status of the video when recording video

_player.setOnTimeoutListener();//Monitor video play timeout

3.Set Default Video Resolution

Two ways to modify video resolution：

1）For modules with two video streams，modify the resolution by select the video stream.

_player.changePipe(_pipe);//Set mobile phone to get the video resolution

_pipe parameter description：

(1) _pipe =Enums.Pipe.H264_PRIMARY//set mobile phone to get first stream H264 format，HD

(2) _pipe =Enums.Pipe.H264_SECONDARY// set mobile phone to get second stream H264 format，SD

(3) _pipe =Enums.Pipe.MJPEG_PRIMARY// set mobile phone to get first stream MJPEG format,HD

(4) _pipe =Enums.Pipe.MJPEG_SECONDARY// set mobile phone to get second stream MJPEG

format，SD

2）For modules with one video stream，modify the resolution by set parameters.

_parametersConfig.setResolution(int type, int resolution);//set the resolution of the video module

resolution parameter description：

0--QVGA(320X240)

1--VGA(640X480)

2--720P(1280X720)

3--1080P(1920X1080)

4. Set the canvas of the video

DisplayView _displayView;//set canvas of the playing video

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

_displayView.setFullScreen(true);//set video full of canvas

layout files as below：

<com.demo.sdk.DisplayView

android:id="@+id/video_view"

android:layout_width="fill_parent"

android:layout_height="fill_parent"

/>

5.For video processing，get the raw data of the video through YUV, then processed.

```
_player.startGetYUVData(true);// Enable get YUV data of the video
_player.setOnGetYUVDataListener(new Player.OnGetYUVDataListener() {
    @Override
    public void onResult(int width, int height, byte[] yData, byte[] uData, byte[] vData) {
        // listen to data of the video get through YUV
    }
});
```

2.4 Video Parameter Configuration Porting

Video parameter configuration is to get and set reference parameter of the video, it depends on ParametersConfig.java。

Set interface	Function Description	Incoming parameters	Return value	
updateUsernameAndPassword	Update the user name and password of the module	User name	Succeed	{"value": "0"}
		password	Failed	Other
getUsernameAndPassword	Get the user name and password of th module	Null	Module's user name and password	
getSsidList	Get wireless network list	Null	Wireless network list	
joinWifi	Connect the module to the router	Router's name	Succeed	{"value": "0"}
		Router's password	Failed	Other
getVersion	Get module version	Null	Module version	
setResolution	Set resolution of the module	Type： 0：local video 1：remote video	Succeed	{"value": "0"}
		resolution： 0：320X240 1：640X480 2：1280X720 3：1920X1080	Failed	Other
getResolution	Get resolution of the module	Type： 0：local video 1：remote video	320X240	{"value": "0"}
			640X480	{"value": "1"}
			1280X720	{"value": "2"}
			1920X1080	{"value": "3"}
setFps	Set Fps of the module	Type： 0：local video 1：remote video	Succeed	{"value": "0"}
		Fps (1~30)	Failed	Other
getFps	Get Fps of the module	Type： 0：local video 1：remote video	Fps of the module	
setQuality	Set the quality of the video	Type： 0：local video 1：remote video	Succeed	{"value": "0"}
		Quality（0~139）	Failed	Other
getQuality	Get the quality of the video	Type： 0：local video 1：remote video	Video quality	
setGOP	Set GOP of the module	gop（0~100）	Succeed	{"value": "0"}
			Failed	Other
getGOP	Get GOP of the module	Null	GOP of the module	

startSdRecord	Start recording with SD card	Type： 0：local video 1：remote video	Succeed	{"value": "0"}
			Busy	{"value": "-4"}
			Space not enough	{"value": "-22"}
			Failed	Other
stopSdRecord	Stop record with SD card	Type： 0：local video 1：remote video	Succeed	{"value": "0"}
			Failed	Other
getSdRecordStatus	Get the record status of SD card	Type： 0：local video 1：remote video	Free	{"value": "0"}
			Busy	{"value": "1"}
setModuleRtcTime	Set RTC time of the module	Date 、 hour 、 minute seconds 、 time zone	Succeed	{"value": "0"}
			Failed	Other
getVideoFolderList	Get record video folder lists in SD card	Null	SD card record video folder list	
getVideoList	Get video list from the record video folder in SD card	SD card video folder path	SD card video folder list	
getSignal	Get signal of the video module	Null	Router's name which the module connected	
			Signal value of the module	

NOTE：

As there are too many module and interface，above is some of the commonly use module configuration interface,so this part we completely open source.You can add any video parameters according to you application.

Any questions, pls contact：steven.tang@rakwireless.com

2.5 Remote nabto Porting

The nabto part is used to open the remote channel to realize the remote video playback, it is depends on libnabto_client_api_jni.so、com.nabto.api package and nabto source files in assets.

1. RemoteTunnel _remoteTunnel=new RemoteTunnel(getApplicationContext());

2. _remoteTunnel.openTunnel(0,getApplicationContext(), 5555, 554, _deviceId);//5555:Mapped video port; 554:default video port ; _deviceId: device id

3. _remoteTunnel.setOnResultListener(new OnResultListener() //Monitor
    {
        @Override
        public void onResult(int id, String result)
        {
            // TODO Auto-generated method stub
            if(result.equals("CONNECT_TIMEOUT")|| // Remote connection timeout
            result.equals("NTCS_CLOSED")|| // Remote device offline
            result.equals("NTCS_UNKNOWN")|| // Unknown error in remote connection
            result.equals("FAILED"))   // Remote connection failed
            {
            }
            else
            {
                // Remote connection succeed，mapped IP is"127.0.0.1"，port is"5555"
            }
    });

4. _remoteTunnel.openTunnel(0,getApplicationContext(), 3333 80, _deviceId);//3333:mapped control port number;   80:default control port ; _deviceId:device id

5._remoteTunnel.closeTunnels();//Close remote connection

6. Attention：
        Local：Target ip is the ip of the module,video play port is 554，control port is 80.
        Remote：Target ip is"127.0.0.1"， the video play port is the port 554 after mapping,the control port is the port 80 after mapping.

2.6 Voice Intercom Porting

It realize module intercom function，it depends on SendAudio.java。

1. Collect Audio Data of PCM format

This part use the audio collection interface of Android.

2. Initialize the voice intercom interface.

SendAudio _sendAudio=new SendAudio();

3. Convert PCM-formatted audio data to PCMU format.

byte[] PCMU_Data=_sendAudio.PCMToPCMU(PCM_Data, PCM_Data_Len);

parameter description：

```
PCM_Data :    PCM audio data content
PCM_Data_Len:  PCM audio data length
```

Return value：

```
PCMU audio data
```

4. Send intercom Audio Data.

_sendAudio.sendAudio(_deviceIp, _voicePort, buf, len);

Paremeter description：

```
_deviceIp :  IP address of the module
_voicePort:  Audio intercom interface of the module
buf:  PCMU audio data connect
len:  PCMU audio data length
```

5. Close Audio Intercom.

_sendAudio.closeSocket();

Attention：

Local：_ip is the ip of the module，_voicePort is 80

Remote：_ip is"127.0.0.1"，_voicePort is the remote connection port 80 after mapping.

2.7 Video play Porting

Download and play the recorded video in the module's SD card.

1.Get the folder list from TF card

```
ParametersConfig _parametersConfig=new ParametersConfig (_ip+":"+_controlPort,_psk);
_parametersConfig.setOnResultListener(new ParametersConfig .OnResultListener() {
    @Override
    public void onResult(ParametersConfig .Response result) {
        if(result.statusCode==200){
            if(result.type==ParametersConfig .GET_VIDEO_FOLDER_LIST){
                //Folder list from TF card
            }
        }
    }
});
_parametersConfig.Get_Video_Folder_List();
```

2.Get video list from one of the folder.

```
ParametersConfig _parametersConfig=new ParametersConfig (_ip+":"+_controlPort,_psk);
_parametersConfig.setOnResultListener(new ParametersConfig .OnResultListener() {
    @Override
    public void onResult(ParametersConfig .Response result) {
        if(result.statusCode==200){
            if(result.type==ParametersConfig .GET_VIDEO_LIST){
                //Get the video list of a folder from TF card
            }
        }
    }
});
_parametersConfig.Get_Video_List(path);// Path is a folder path of module's TF Card
```

3.Play the video according to the path of the floder and video.

```
Mp4Download.playMp4File(url, _psk, savePath, videoHandler);
```

Parameter Description：

url：Path of the video playback ,for example： http://admin:admin@192.168.100.1/link//mnt/rec_folder/video/pipe0/1970Y01M04D15H/NVTDV19700104_150156.mp4

savePath：save the video to specified path of the mobile phone .

videoHandler：return related status when play the video.

Attention：

_psk is the password of the module，default password is admin。

Local：_ip is the ip of the module，controlPort is 80.

Remote ：_ip is "127.0.0.1"，controlPort is the port 80 after mapping when remote connection.

2.8 Transparent Transmission Porting

Transparent transmission is mainly to realize the function of real-time communication between mobile phone and module.

Some of the products realize transparent transmission by TCP, the target port is 80；some by UDP, and the target port is 1008,more details needed please refer to the documents such as specification。

1.TCP Transparent transmission

（1）Create TCP connection

Socket _socket = new Socket(_deviceIp, _sendPort);

_socket.setKeepAlive(true);

dataStream =new DataOutputStream(_socket.getOutputStream());

（2）TCP send data

dataStream.write(message);

（3）TCP recieve data

_socket.getInputStream().read(buffer);

（4）Close TCPc connection

_socket.close();

dataStream.close();

2.UDP transparent transmission

（1）Create UDP connection

DatagramSocket udp_socket=new DatagramSocket(25000);

（2）UDP send data

InetAddress serverAddress = InetAddress.getByName(_deviceIp);

DatagramPacket sendPackage = new DatagramPacket(data , data.length , serverAddress , _sendPort);

udp_socket.send(sendPackage);

（3）UDP recieve data

DatagramPacket recvPackage = new DatagramPacket(buffer , buffer.length);

udp_socket.receive(recvPackage);

（4）Close UDP connection

udp_socket.close();

3.Attention：

Send data begins with 0x01 0x55，the module will add 0x01 0x55 automatically when receive the data . That is：

When sending data：0x01 0x55 content data needs to be sent

When receiving data：0x01 0x55 content data needs to be received

Local：_deviceIp is the ip of the module，_sendPort is 80。

Remote：_deviceIp is "127.0.0.1"，_sendPort is the port for 80 after mapping when remote connection.

3. Related Permission

WisView SDK need following permissions：

```
<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="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. Revision History

Version	Author	Date	Modification
V1.0	Qu Jin	2016/03/05	Initial Draft
V1.1	Qu Jin	2016/05/17	1. Update set the video canvas interface，can display video video on signal or double screen，also can choose video display method SurfaceView or TextureView. 2. Add the interface to get the video YUV data.
V1.2	Qu Jin	2016/08/05	1. Add hardware decoding
V1.3	Qu Jin	2016/12/01	1.Optimize local scan. 2.Keep two kinds of recording format:ffmpeg and mp4v2. 3.Add the function of edit photos 4.Solve the problem of flash back when playing AAC audio. 5.Add the function of video playback. 6.Add the function of transparent transmission.
V1.4	Qu Jin	2017/02/24	1 Optimize video record.
V2.0	Qu Jin	2017/04/11	1.Sort out and open the interface of setting parameters. 2.Sort out and open interface of intercom. 3.Sort out SDK。