

WisView Video IOS SDK Porting

User Guide V2.0

Shenzhen Rakwireless Technology Co., Ltd

www.rakwireless.com info@rakwireless.com

© 2017 Rakwireless all rights reserved .

Mentioned in this document , the actual company and
product names, trademarks are their respective owners.

After update the new version, this document without prior
notice.

Content

1. Introduction	3
1.1 Summary.....	3
2. Transplant Description.....	4
2.1 Local Scan Transplant.....	4
2.2 Transplant APConfig	5
2.3 Transplant play video.....	6
2.4 Transplant Video parameter config.....	10
2.5 Transplant remote nabto.....	13
2.6 Transplant Voice Intercom	14
2.7 Transplant Video Playback	15
2.8 Transplant Transparent Transmission	16
3. Reference Frameworks	18
4. Revision History	19

1. Introduction

1.1 Summary

WisView video SDK mainly achieve the following functions::

- 1) Support local scan, and get the reference information about the video module
- 2) Support APConfig,configure the module to the router.
- 3) Support smart-phone watch the video locally,and remotely through data traffic.
- 4) Support functions as: take-photo、record、intercom、change pipe.
- 5) Support set various parameters of the video module.
- 6) Support VR and split-screen watch the video.
- 7) Support hardware and software decoding.
- 8) Support decoding H264 and MJPEG format.

2. Transplant Description

2.1 Local Scan Transplant

Scanner is used to scan reference information of the module locally, it depends on libWisView.a, Scanner.h.

Use method as below:

1. Initialize local scan interface.

```
#import "Scanner.h" //Cite the header file  
Scanner *_scanner = [[Scanner alloc] init]; //Used to call Scanner reference interface.
```

2. Set local scan monitor.

```
- (void)scanDeviceOver:(Scanner *)result;  
{  
    if (result.Device_ID_Arr.count > 0) {  
        //Listen to scan complete event,if find the device then can get the device's information  
        [result.Device_ID_Arr enumerateObjectsUsingBlock:^(id obj, NSUInteger idx, BOOL *stop) {  
            NSString *deviceIp = [result.Device_IP_Arr objectAtIndex:idx]; //The id of the device to be scanned  
            NSString *deviceId = [result.Device_ID_Arr objectAtIndex:idx]; //The ip address of the  
device to be scanned  
        }  
    }  
}
```

3. Start local scan.

```
Scanner *result = [_device_Scan ScanDeviceWithTime:2.0f]; //Start scanning, scanning time is 2.0s  
[self performSelectorOnMainThread:@selector(scanDeviceOver:) withObject:result waitUntilDone:NO];
```

2.2 Transplant APConfig

APConfig is the way to configure the device to the router. That the device establish an hot spot, then smart-phone connect the device and set configure information. It is depends on libWisView.a and ParametersConfig.h.

Use method as below:

1. Initialize parameter config interface of the module

```
#import "Scanner.h" //Cite the header file  
<ParametersConfigDelegate> //Add a proxy, used to get configure return data  
//_ip: ip address of the module _psk: password of the module initialization used to call Parameters  
Configure reference interface
```

```
ParametersConfig *_parametersConfigStep2AP=[[ParametersConfig alloc] init:self ip:_ip password:_psk];
```

2. Set the parameters to configure the interface to monitor

```
- (void)setOnResultListener:(int)statusCode :(NSString*)body :(int)type{  
    if (type==JOIN_WIFI) {  
        if(statusCode==200){ //APConfig succeed  
        }  
    }  
    else if (type==GET_SSID_LIST) {  
        if(statusCode==200){ //Get the list of network the device gets  
        }  
    }  
}
```

3. [_parametersConfigStep2AP getSsidList]; //Get the list of network the device gets

4. [_parametersConfigStep2AP joinWifi:_ssid _psk]; //Config the device to join the router, and incoming the router name and password

2.3 Transplant play video

Video play part is mainly the process of decoding the audio and video data stream, it is depends on libWisView.a, WisView.h.

Main interface as below:

1. Initialize the video play interface

(1)Single screen play

```
_videoView = [[WisView alloc] initWithFrame:CGRectMake(0, 0,w,h)];  
[_videoView setView1Frame:frame]; //Set the frame of video play view  
[_videoView set_log_level:4]; //Set log print method  
[_videoView delegate:self]; //Set a proxy  
[self.view addSubview:_videoView]; //Add the view of the video shows
```

(2)Dual screen play

```
_videoView = [[WisView alloc] initWithFrame2:frame:frame1:frame2];  
// frame1: the frame of the first video on _videoView; frame2: the frame of the second video on  
_videoView  
[_videoView setView1Frame:frame1]; //Set the frame1 of the video play view1  
[_videoView setView2Frame:frame2]; //Set the frame2 of the video play view2  
[_videoView set_log_level:4]; //Set log print method  
[_videoView delegate:self]; //Set a proxy  
[self.view addSubview:_videoView]; //Add the view the video shows
```

Reference interface:

- (void)setView1Frame:(CGRect)frame; //Set the frame of the first video
- (void)setView2Frame:(CGRect)frame; //Set the frame of the second video
- (CGRect)getView1Frame; //Get the frame of the first video
- (CGRect)getView2Frame; //Get the frame of the second video
- (void)setView1Hidden:(BOOL)isHidden; // Set the first video to be hidden
- (void)setView2Hidden:(BOOL)isHidden; // Set the second video to be hidden

2. Set play video parameters

```
NSString *url ;  
url = [NSString stringWithFormat:@"rtsp://admin:% @ % @ :%d/cam1/% @",_psk,_ip,_port,_pipe];  
_psk: Set connection password  
_ip: target IP of the play video, local play ip is the ip of the device,when remotely is "127.0.0.1"  
_port: target port of the play video,local play port is 554,when remotely the port is the mapping port
```

Two ways to modify the video resolution

1) For modules with two way video, select one video to switch the resolution.

_pipe parameter description:

(1) `_pipe = @ "h264" // Set smart-phone to get first H264 video, HD`

(2) `_pipe = @ "h264-1" // Set smart-phone to get second H264 video, SD`

(3) `_pipe = @ "mpeg4" // Set the smart-phone to get the first MJPEG video, HD`

(4) `_pipe = @ "mpeg4-1" // Set the smart-phone to get the second MJPEG video, SD`

2) For modules with only one way video, modify the resolution through parameters to set the interface

`[_parametersConfig setResolution: type :resolution] // Set the resolution of the video module`

Resolution parameter description:

`0--QVGA(320X240)`

`1--VGA(640X480)`

`2--720P(1280X720)`

`3--1080P(1920X1080)`

`[_videoView sound:NO] // Set open or close voice`

`[_videoView set_record_frame_rate:10] // Set the frame rate when recording video`

`[_videoView play:url useTcp:NO] // Get and play the video through UDP or TCP`

`[_videoView stop] // Stop playing the video`

3. Take-photo and record

`[_videoView take_photo] // Take-photo`

`[_videoView begin_record:type] // Start recording, type: 0 ffmpeg record 1 mp4v2 record`

`[_videoView begin_record2:type :path] // Start recording the video to the specified path, the path is path`

`[_videoView end_record] // Finish recording`

4. Monitor the video play status

```
- (void)state_changed:(int)state{
    switch (state) {
        case 0: //Idle state{
            break;
        }
        case 1: //Ready to play{
            break;
        }
        case 2: ///Playing{
            break;
        }
        case 3: //Have stopped playing{
            break;
        }
        default:
            break;
    }
}

- (void)video_info:(NSString *)codecName codecLongName:(NSString *)codecLongName{
    //Listen to the played video information
}

- (void)audio_info:(NSString *)codecName codecLongName:(NSString *)codecLongName
    sampleRate:(int)sampleRate channels:(int)channels{
    //Listen to the played audio information
}
```

5. Get the video YUV date after decoding

```
- (void)startGetYUVData:(BOOL)start;//Enable get the video YUV data after decoding.
- (void)GetYUVData:(int)width :(int)height
    :(Byte*)yData :(Byte*)uData :(Byte*)vData
    :(int)ySize :(int)uSize :(int)vSize;//Listen to the video YUV data after decoding
{// Get the video YUV data after decoding.
```




深圳市瑞科慧联科技有限公司
Shenzhen Rakwireless Technology Co., Ltd

WisView Video SDK Porting User Guide

}

2.4 Transplant Video parameter config

Video parameter configure part is to get and configure reference parameters of the video, it is depends on ParametersConfig.m, ParametersConfig.h.

Config Interface	Function Description	Incoming parameters	Back value	
updateUsernameAndPassword	Update user name and password of the module	User name	Succeed	{"value": "0"}
		Password	Failed	Other
getUsernameAndPassword	Get user name and password of the module	Nun	user name and password of the module	
getSsidList	Get wireless network list	Nun	Wireless network list	
joinWifi	Configure the module connect to router	Router name	Succeed	{"value": "0"}
		Router password	Failed	Other
getVersion	Get the version of the module	Nun	Version of the module	
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 frame rate of the module	Type: 0:local video 1:remote video	Succeed	{"value": "0"}
		Frame rate(1~30)	Failed	Other
getFps	Get frame rate of the module	Type: 0:local video 1: remote video	Frame rate of the modules	

setQuality	Set quality of the video	Type: 0: local video 1: remote video	Succeed	{"value": "0"}
		Quality (0~139)	Failed	Other
getQuality	Get the video quality	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	Nu	GOP of the module	
startSdRecord	Start recording video with SD card	Type: 0: local video 1: remote video	Succeed	{"value": "0"}
			Busy	{"value": "-4"}
			Not enough space	{"value": "-22"}
			Failed	Other
stopSdRecord	Stop recording video with SD card	Type: 0: local video 1: remote video	Succeed	{"value": "0"}
			Failed	other
getSdRecordStatus	Get SD card recording status	Type: 0: local video 1: remote video	idle	{"value": "0"}
			Busy	{"value": "1"}
setModuleRtcTime	Set module RTC time	Data, hour, minute, seconds, timezone	Succeed	{"value": "0"}
			Failed	other
getVideoFolderList	Get SD card record video folder list	Nun	SD card record video file list	
getVideoList	Get the video list of the SD card recording video folder	SD card video folder path	SD card video folder path	
getSignal	Get the video module signal value	Nun	The router name which the module connected to	
			Signal value of the module	

NOTE:

Since there are too many type of module and the interface, above is only parts of the commonly used module configuration interface. So we complete open source for this part, you can add the video parameters according to your application.



深圳市瑞科慧联科技有限公司
Shenzhen Rakwireless Technology Co., Ltd

WisView Video SDK Porting User Guide

If there is any question please contact: steven.tang@rakwireless.com

2.5 Transplant remote nabto

Remote nabto part is used to open the remote channel to realize playing video remotely, it depends on common package and 3rdParty package nabtopackage.

1. NabtoLibraryInit();//Initialize nabto

2. Async_ConnectDeviceWithTunnel(&videoTunnel,deviceId,554,5555);//videoTunnel: video channel
5555:mapped play port number; 554:video default port; _deviceId:device id

3. Async_ConnectDeviceWithTunnel(&httpTunnel,deviceId,554,3333);//httpTunnel: control(transparent
transmission) channel, 3333:mapped control port; 80:control default port ; _deviceId:device id

4. int status = CheckConnectStatus(&videoTunnel);//status is 0 means remote connection succeed, mapped IP
is “127.0.0.1”, port number is “5555”
int status = CheckConnectStatus(&httpTunnel);//status is 0 means remote connections succeed, mapped IP
is “127.0.0.1”, port number is “3333”

5. CloseTunnel(&videoTunnel);//Close video channel

CloseTunnel(&httpTunnel);//Close control(transparent transmission) channel

Note:

Locally : target ip is the ip of the module , video play port is 554 , control port is 80.

Remotely:target ip is “127.0.0.1” , video play port is the port 554 after mapping,control port
is the port 80 after mapping.

2.6 Transplant Voice Intercom

Voice intercom part realize intercom function, it depends on AudioRecord.m, AudioRecord.h, sendAudio.m, sendAudio.h.

1. Collect PCM format voice data

This part use IOS voice collection interface.

2. Initialize voice intercom interface.

```
AudioRecord* audioRecord = [[AudioRecord alloc] init];
```

3. Record PCMU format voice data.

```
[audioRecord StartRecord]; //Start recording PCMU format voice data
```

```
NSData* PCMUData = [audioRecord StopRecord]; //Stop recording PCMU voice data
```

4. Sent intercom voice data.

```
[sendAudio sendWithIp: _deviceIp port: _voicePort data:PCMUData];
```

Parameter Description:

_deviceIp : IP address of the module

_voicePort: voice intercom interface of the module

PCMUData: PCMU voice data content

Note:

Locally: _deviceIp is the module ip, _voicePort is 80.

Remotely: _deviceIp is "127.0.0.1", _voicePort is the port 80 after mapping.

2.7 Transplant Video Playback

Video playback is to download and play the video folder recording in the TF card .

Use method as below

1. Initiaze module parameter config interface

```
#import "Scanner.h" //Cite the header file

<ParametersConfigDelegate> //Add a proxy, used for receive configure return data

//_ip: ip address of the module _psk: module password initialize is used for call ParametersConfig
reference interface
```

```
ParametersConfig *_parametersConfig=[[ParametersConfig alloc]init:self ip:_ip password:_psk];
```

2. Set the parameters to configura interface to listen

```
- (void)setOnResultListener:(int)statusCode :(NSString*)body :(int)type{
    if (type==GET_VIDEO_FOLDER_LIST) {
        if(statusCode==200){ //Get video folder list from TF card
        }
    }
    else if (type==GET_VIDEO_LIST) {
        if(statusCode==200){ //Get video list of one of the folder in TF card
        }
    }
}
```

3. [_parametersConfig getVideoFolderList]; //Get the video folder list from TF card

4. [_parametersConfig getVideoList:folder]; //Get video list of one of the folder in the TF card

5. Play the video according to the obtained video folder and video path.

```
MPMoviePlayerController *_moviePlayer=[[MPMoviePlayerController alloc]initWithContentURL:url];
_moviePlayer.view.frame=self.view.bounds;
_moviePlayer.view.autoresizingMask=UIViewAutoresizingFlexibleWidth|UIViewAutoresizingFlexibleHeight;
[self.view addSubview:_moviePlayer.view];
```

Parameter Description:

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

Note:

_psk is the module password, default is admin。

Locally: _ip is the ip of the module, control Port is 80.

Remotely: _ip is “127.0.0.1” , control Port is the port 80 after mapping.

2.8 Transplant Transparent Transmission

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

Some of the products realize transparent transmission by establish TCP connection, target port is 80;some by UDP connection, target port number is1008, for details please refer to the specifications of the products.

1.TCP transparent transmission

(1) Creat TCP connection

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

(2) TCP send data

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

(3) TCP recieve data

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

(4) Close TCP connection

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

2.UDP transparent transmission

(1) Creat UDP connection

```
GCDUdpSocket = [[GCDAsyncUdpSocket alloc] initWithDelegate:self  
delegateQueue:dispatch_get_main_queue()];
```



```
[GCDUdpSocket bindToPort :25000 error:nil];
```

(2) UDP send data

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

(3) UDP receive data

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

(4) Close UDP connection

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

Note:

Send data all begins with 01 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](#)

Locally: _deviceIP is the ip of the module , _sendPort is 80.

Remotely: _deviceIp is “127.0.0.1” , _sendPort is the port 80 after mapping.

3. Reference Frameworks

WisView SDK need to be used 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. Revision History

Version	Autor	Date	Modification
V1.0	Jin Qu	2016/03/05	Create the document
V1.1	Jin Qu	2016/07/07	1.Add the split-screen display. 2.Add record video to the specified path. 3.Add get the YUV data after decoding.
V1.2	Jin Qu	2016/12/02	1.Save ffmpeg and mp4v2 two record format. 2.Avoid playing 5275 flash broken problems. 3.Add video playback function. 4. Add transparent transmission function.
V1.3	Jin Qu	2017/02/24	1.Optimized record video
V2.0	Jin Qu	2017/04/17	1. Sort out and open the interface of setting parameters. 2. Sort out and open interface of intercom. 3.Sort out SDK。