Transplant description of RakVideo Tool V1.4

Shenzhen Rakwireless Technology Co., Ltd

www.rakwireless.com
info@rakwireless.com

1. Brief introduction

1.1 Summary

RakVideo achieve the following functions:

1) Local connect to device's AP to watch video.

2) Configure device to connect network by EasyConfig and ApConfig.

3) Save、Modify、Delete device have added.

4) Mobile local watch video or mobile phone to open data traffic remote watch video.

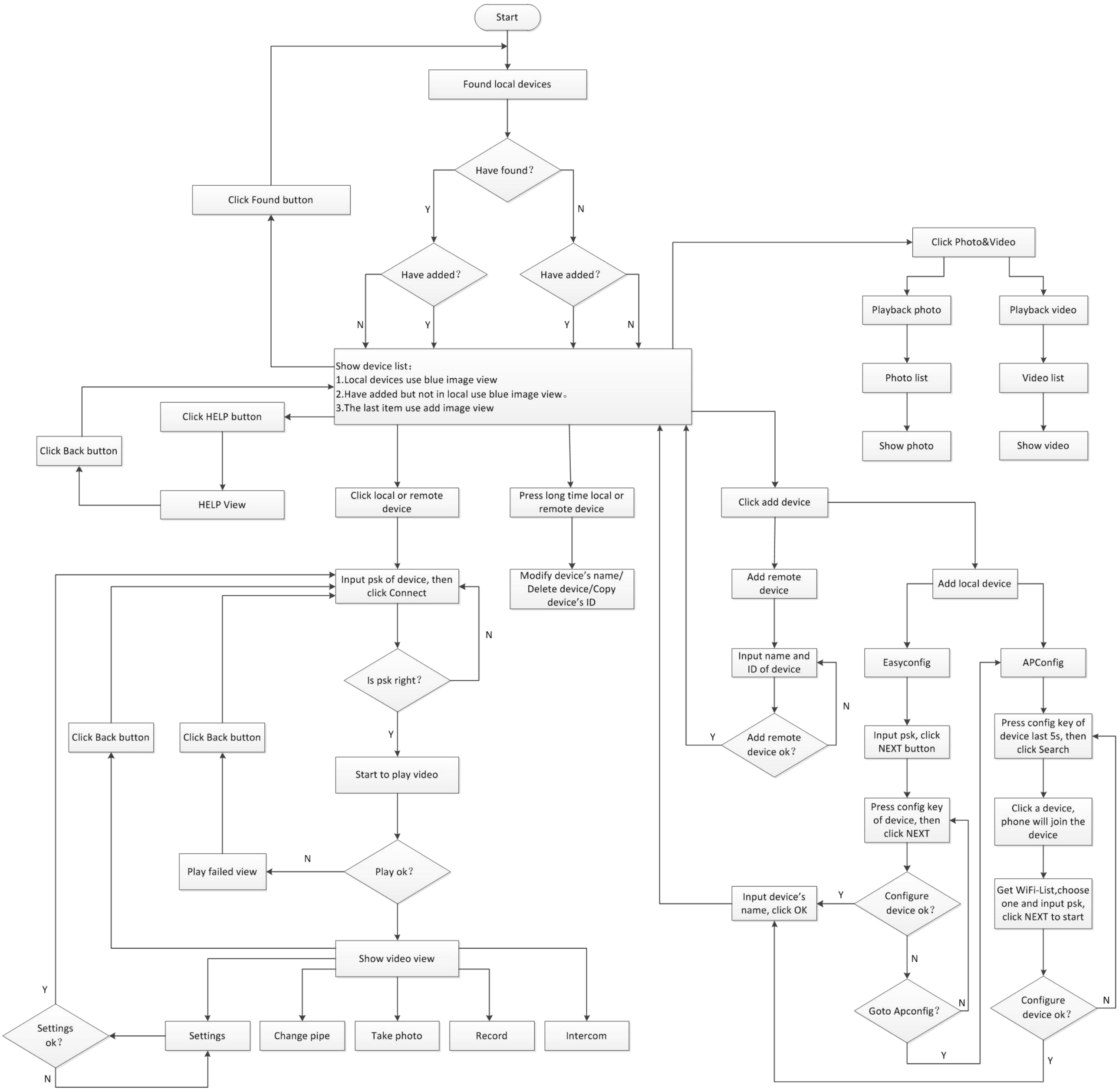
5) Achieve the functions as take-photo/record/change pipe/intercom and so on.

6) Look over the photo and video have saved.

7) Automatic switching language in Chinese and English (the mobile phone system is Chinese, it is displayed in Chinese, otherwise it is displayed in English).

1.2 Design flow chart

Design flow chart of RakVideo tool as following:


Start
Found local devices
Have found？
Click Found button
Y
N
Have added？
Have added？
N
Y
Y
N
Show device list：
1.Local devices use blue image view
2.Have added but not in local use blue image view。
3.The last item use add image view
Click HELP button
Click Back button
HELP View
Click local or remote device
Press long time local or remote device
Modify device's name/ Delete device/Copy device's ID
Input psk of device, then click Connect
N
Is psk right？
Y
Click Back button
Click Back button
Start to play video
Play failed view
N
Play ok？
Show video view
Y
Settings ok？
N
Settings
Change pipe
Take photo
Record
Intercom
Click Photo&Video
Playback photo
Playback video
Photo list
Video list
Show photo
Show video
Click add device
Add remote device
Add local device
Input name and ID of device
N
Y
Add remote device ok？
Easyconfig
APConfig
Input psk, click NEXT button
Press config key of device last 5s, then click Search
Press config key of device, then click NEXT
Click a device, phone will join the device
Get WiFi-List,choose one and input psk, click NEXT to start
Input device's name, click OK
Y
Configure device ok？
N
Goto Apconfig？
N
Y
Configure device ok？
N
Y

2. Transplant introduction

2.1 Summary

Procedures for transplantation mainly include Easyconfig, ApConfig, local watch video and remote watch video, etc..

If transplant Android Studio project,it is dependent on wiseview.sdk.aar, jason-all.jar, libnabto_client_api_jni.so several libraries and nabto resource files.

If transplant Eclipse project,it is dependent on wiseview.sdk.jar, libdemo.sdk.so, jason-all.jar, libnabto_client_api_jni.so several libraries and nabto resource files.

In fact,wiseview.sdk.aar in Android Studio project is equal to wiseview.sdk.jar and libdemo.sdk.so in the Eclipse project.

The following introduction show how to transplant Android Studio project as an example.

2.2 Transplant Easyconfig

Easyconfig is a way to configure the device to quickly add to the router, it is dependent on wiseview.sdk.aar。The method to use is as following:

```
1. EasyConfig _easyConfig = new EasyConfig();    //Defines a class object, which is used to call the EasyConfig related interface.

2. _easyConfig.setOnStopListener(new EasyConfig.OnStopListener() {
       @Override
       public void onStop() {
                  ......    //Listen stop configuration event.
           }
       });

3. _easyConfig.setOnProgressListener(new EasyConfig.OnProgressListener() {
       @Override
       public void onData(int progress, String ip, String mac,String id) {
                       ......    //Listener configuration data return in the process, if the configuration success will return IP, MAC, ID information of the device.
           }
       });

4. _easyConfig.start(_password);//Start configuration,Incoming parameters for the router's password.

       5. _easyConfig.stop();//Stop configuration.
```

Specific please refer to AddDeviceStep3.java in the project.

2.3 Transplant 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 is dependent on wiseview.sdk.aar、WLANAPI.java and LX520.java。

The method to use is as following:

1. lx520.Get_Ssid_List();//Get the network list of devices.

2. Scanner _scanner = new Scanner(AddDeviceStep3.this); //Defines a class object, which is used to call the Scanner related interface.

```
_scanner.setOnScanOverListener(new Scanner.OnScanOverListener() {
            @Override
            public void onResult(Map<InetAddress, String> data, InetAddress gatewayAddress) {
            ......    //Listen the completion event of scan, if the device can be found,you will obtain the device's
            information, here is mainly to record the ID of this configuration device.
        }
    });
    _scanner.scanAll(); //Start to scan.
```

3. lx520.joinWifi(_ssid, _password); //Configure the device to connect to a router,Incoming parameters for the router's name and password.

4. mWifiAdmin.addNetWork(mWifiAdmin.CreateWifiInfo(_ssid, _password, 1));//Mobile phone automatically switches to the router have configured, Incoming parameters for the router's name ,password and encryption.

5. Call the Scanner again, to find the device have configured.

Specific please refer to AddDeviceStep2AP.java and AddDeviceStep3.java in the project.

2.4 Transplant paly video

Video play part is mainly the process of decoding the audio and video data stream, it is dependent on wiseview.sdk.aar.

The method to use is as following:

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 user name
_module.setPassword("admin");//Set password
_module.setPlayerPort(554);//Set target port of playing video
_module.setModuleIp(_moduleIp);//Set target IP of playing video
```

2.Set play video related parameters

```
Player _player = _module.getPlayer();
_player.setTimeout(10000);//Set the timeout, unit: ms
_player.setRecordFrameRate(10);//Set the video frame rate
_player.setAudioOutput(false);//Set the sound on or off
_player.setDisplayView(context, _displayView, _displayView2, _viewType);//Set the playing canvas
```

Parameter description:

(1) _displayView、_displayView2//canvas

(2) _viewType//0: SurfaceView(Software decoder) 1: TextureView(Software decoder) 2: TextureView (Hardware decoder)

Instructions：

(1) If only use a single view to display,set _displayView2 or _displayView to null. Such as:

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

(2) If you use TextureView, you can get the TextureView by the following method, and then do the corresponding transformation.

```
//Get TextureView of displayView.
    TextureView _textureView=_displayView.getGLTextureView();
    if(_textureView!=null){
        _textureView.setRotation(45.0f);//Rotate clockwise 45°
    }
//Get TextureView of displayView2.
    TextureView _textureView2=_displayView2.getGLTextureView2();
    if(_textureView2!=null){
        _textureView2.setRotation(-45.0f);//Counter clockwise rotation 45°
    }

_player.getState();//Get the video playback status
```

Status description：

(1) Enums.State.IDLE//Free state

```
        (2) Enums.State.PLAYING//Playing
        (3) Enums.State.PREPARING//Ready to play
        (4) Enums.State.STOPPED//Have stopped playing
    _player.play(_pipe, Enums.Transport.UDP);//Get video and play through UDP
    _player.play(_pipe, Enums.Transport.TCP);//Get video and play through TCP
    _player.stop();//Stop playing video
    boolean _recording = _player.isRecording();//Whether is recording video
    _player.beginRecord(String path, String name);//Start recording,use mp4v2
    _player.beginRecord0(String path, String name);//Start recording,use ffmpeg
    _player.endRecord();//End of recording
    Bitmap photo = _player.takePhoto();//Take photos

    _player.setOnStateChangedListener();//Monitor the status of the video when the state have changed
    _player.setOnRecordStateChangedListener();//Monitor the status of the recording video
    _player.setOnTimeoutListener();//Monitor play video timeout
```

3.Set default video resolution

```
    Controller _controller = _module.getController();
    _player.changePipe(_pipe);//Set the resolution of the video that the phone to get
    _pipe parameter description：
    (1) _pipe =Enums.Pipe.H264_PRIMARY//Set the phone to get the video for the H264 format，
          resolution 1280X720
    (2) _pipe =Enums.Pipe.H264_SECONDARY//Set the phone to get the video for the H264 format， resolution
          320X240
    (3) _pipe =Enums.Pipe.MJPEG_PRIMARY//Set the phone to get the video for the MJPEG format， resolution
          1280X720
    (4) _pipe =Enums.Pipe.MJPEG_SECONDARYSet the phone to get the video for the MJPEG format，
          resolution 320X240
```

4.The set video on canvas

```
DisplayView _displayView;//Set play video canvas
    _displayView = (DisplayView)findViewById(R.id.sview);
    _displayView.setFullScreen(true);//Set video full canvas
    Layout file as follows：
<com.demo.sdk.DisplayView
    android:id="@+id/video_view"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    />
```

5.Get YUV data of the video

```
  _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 get YUV data of the video
        }
    });
```

Specific please refer to VideoPlay.java in the project.

2.5 Transplant nabto

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

The method to use is as following:

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

2. _remoteTunnel.openTunnel(0,getApplicationContext(), 5555, 554, _deviceId);//5555:Mapped play port number; 554:Play port ; _deviceId:Decice 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 is not online
            result.equals("NTCS_UNKNOWN")||          //Unknown error in remote connection
            result.equals("FAILED"))                 //Remote connection failure

            {
            }
            else
            {
            //Remote connection success, mapping IP “127.0.0.1”, Port number “5555”
            }
    });
```

4. _remoteTunnel.openTunnel(0,getApplicationContext(), 3333,80, _deviceId);//3333:Mapped control port number; 80:Control port ; _deviceId:Decice id

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

6. Notice:

In local:Target IP is the module of the IP, the video play port is 554, the control port is 80.

In remote:Target IP is "127.0.0.1",the video play port is the port mapped for 554,the control port is the port mapped for 80.

Specific please refer to VideoPlay.java in the project.

2.6 Video playback

Download and play the recorded video in module's TF Card.

1.Get the folder list from module's TF Card

```
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){
                //The folder list from module's TF Card
            }
        }
    }
});
lx520.Get_Video_Folder_List();
```

2.Get the video list from a folder

```
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){
                    //The video list from a folder
                }
            }
        }
    }
});
lx520.Get_Video_List(path);//path is a folder path of module's TF Card
```

3.Download and Play video

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

Parameter description：

url：The video path ,e.g.：http://admin:admin@192.168.100.1/link//mnt/rec_folder/video/pipe0/1970Y01M04D15H/NVTDV19700104_150156.mp4

savePath：save the video to specify path of the phone.

videoHandler：Relevant state returned.

4.NOTE:

_psk is module's password,the default is admin.

In local:_ip is module's ip, controlPort is 80.

In remote:_ip is "127.0.0.1″, controlPort is the port mapped for 80.

Specific please refer to PlayBackVideoActivity.java, PlayBackFolderListActivity.java,PlayBackVideoListActivity.java in the project.

2.7 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 TCP, the destination port number is 80, such as the LX520 module; Some of the products realize transparent transmission by UDP, the destination port number is 1008, such as RAK566, please see the specific corresponding product specifications and other documents.

1.TCP transparent transmission

（1）Create TCP Socket

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

（2）Send data by TCP Socket

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

（3）Recieve data by TCP Socket

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

（4）Close TCP Socket

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

2.UDP transparent transmission

（1）Create UDP Socket

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

（2）Send data by UDP Socket

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

（3）Recieve data by UDP Socket

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

（4）Close UDP Socket

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

3.NOTE:

Send data begin with 0x01 0x55, recieved data will add 0x01 0x55 in the begin, such as:

Send data: 0x01 0x55 real send data

Recieve data: 0x01 0x55 real recieve data

In local:_ip is module's ip, controlPort is 80.

In remote:_ip is "127.0.0.1″, controlPort is the port mapped for 80.

Specific please refer to VideoPlay.java，DeviceUart.java in the project.

3. Relevant permissions

RakVideo tool need permissions to be used as following：

```
 <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. Revision History

Version	Author	Date	Modification
V1.0	Jean	2016/03/05	Create the document
V1.1	Jean	2016/05/17	1.Update interface of set canvas ,you can set single or dual view to display, and you can choose surfaceview or TextureView to display. 2.Add interface to get YUV data of the video.
V1.2	Jean	2016/08/05	1.Add Hardware Decoder.
V1.3	Jean	2016/12/01	1.Optimized local scan 2.Retain two recording methods,ffmpeg and mp4v2. 3.Add image processing function. 4.Avoid playing 5275 flash broken problems. 5.Add video playback function. 6.Add transparent transmission function.
V1.4	Jean	2017/02/24	1.Optimized record video.