

UVC USB Camera Android JNI Specification

深圳市锐尔威视科技有限公司
RERVISION TECHNOLOGY CO., LTD

2015.7.21 V1.1

目录

开发前准备3

 1 确定 UVC 驱动3

 2 安装 APK 测试3

 3 导入 JNI 库.....3

一、预览拍照录像4

 预览流程4

 录像流程6

 拍照流程8

二、参数设置9

开发前准备

1 确定 UVC 驱动

在Android系统开发USB Camera应用，需要底层Linux系统加载UVC驱动模块，如果是自己开发的系统，请开发人员在编译系统时，把Linux的UVC配置打开，参见《Linux UVC配置说明》。如果不是自己开发系统，请向设备商或方案商要求其加入UVC。

设备一定要先root才能使用，因为应用要操作硬件设备，必须要有最高权限。

2 安装 APK 测试

在Android设备上安装HdmjpegCamera.apk，插入我们的USB Camera，确定可以正常开启预览画面，并正常使用其功能。说明你的Android设备已经具备开发的能力。

3 导入 JNI 库

将“JNI-LIB”目录的“libs”文件夹放到你的JAVA工程中，加载动态库的JAVA类的全名必须是com.camera.hdmjpegcamera.CameraPreview

声明 JNI 函数接口：

```
public native void    nPrepareCamera(byte[] buffers,int size,String path);
public native int     nReleaseCamera();
public native void    nCameraThread();
public native void    nStartMjpgRecord(String flie);
public native void    nStopMjpgRecord();
public native int     nTakePhoto(String file,int size);
public native void    nConctrlPreview(int value);
public native void    nSetRecordSize(int flag);
public native int     getRecordSize();
```

加载动态库：

```
static
{
    try {
        System.loadLibrary("HDMjpegCamera");
    } catch (UnsatisfiedLinkError ule) {
        Log.e(TAG, "can't load HDMjpegCamera library: " + ule);
        System.exit(1);
    } catch (SecurityException se) {
        Log.e(TAG, "can't load HDMjpegCamera library: " + se);
        System.exit(1);
    }
}
```

一、预览拍照录像

预览流程

1) *nPrepareCamera*

接口功能：初始化摄像头

返回值：无

参 数：

buffer --- java层byte[]型变量，用来保存一帧图像数据

size --- preview的分辨率

3: 1920x1080

2: 1280x720

1: 640x480

path --- 存储路径根目录，如/mnt/extsd

```
void Java_com_camera_hdmjpegcamera_CameraPreview_nPrepareCamera(JNIEnv* env,
jobject thiz, jobject buffer, jint size, jstring path);
```

用法：

```
nPrepareCamera(mcp.mBuffers, CameraPreview.SIZE720P, cameraApplication.CAMERA
_DIR);
```

```
public byte[] mBuffers =new byte[MAXLENGHT];
public static final int SIZE480P =1;
public static final int SIZE720P =2;
public static final int SIZE1080P=3;
public static String CAMERA_DIR;
CAMERA_DIR = Environment.getExternalStorageDirectory().getAbsolutePath();
```

存储路径形式：

照片：/mnt/extsd/HDUSBCamera/DCIM

录像：/mnt/extsd/HDUSBCamera/Record

2) *nCameraThread*

接口功能：MJPEG解码预览线程

返回值：无

参 数：无

```
void Java_com_camera_hdmjpegcamera_CameraPreview_nCameraThread(JNIEnv * env,
jobject thiz);
```

用法：

```
private CameraPreview mcp;
```

```
private Thread cameraThread = null;
if(cameraThread == null){
    cameraThread = new Thread(new Runnable() {
        @Override
        public void run() {
            Log.d(TAG, "录像服务开启了");
            mcp.nCameraThread();
        }
    });
    cameraThread.start();
}
```

3) *nConctrlPreview*

接口功能：控制Mjpeg解码预览

返回值：无

参 数：flag ---

1 开启预览

0 关闭预览。

```
void Java_com_camera_hdmjpegcamera_CameraPreview_nConctrlPreview(JNIEnv* env,
jobject thiz, jint flag);
```

用法：

```
@Override
public void surfaceCreated(SurfaceHolder holder)
{
    nConctrlPreview(1);
}
@Override
public void surfaceDestroyed(SurfaceHolder holder)
{
    nConctrlPreview(0);
}
```

4) *nReleaseCamera*

接口功能：关闭摄像头

返回值：无

参 数：无

```
void Java_com_camera_hdmjpegcamera_CameraPreview_nReleaseCamera(JNIEnv* env,
jobject thiz);
```

用法：

```
nReleaseCamera();
```

录像流程

5) *nStartMjpgRecord*

接口功能：开始录像

返回值：无

参 数：

file --- 录像文件名（不用加后缀）

```
void Java_com_camera_hdmjpegcamera_CameraPreview_nStartMjpgRecord(JNIEnv*  
env, jobject thiz, jstring file);
```

用法：

```
private CameraPreview cp;  
.....  
long mRecordingStartTime = System.currentTimeMillis();  
String videoName= SimpleDateFormat.format(new Date(mRecordingStartTime));  
cp.nStartMjpgRecord(videoName);
```

6) *nStopMjpgRecord*

接口功能：停止录像

返回值：无

参 数：无

```
void Java_com_camera_hdmjpegcamera_CameraPreview_nStopMjpgRecord(JNIEnv* env,  
jobject thiz)
```

用法：

```
cp.nStopMjpgRecord();
```

7) *getRecordSize*。

接口功能：查询摄像头支持的录像分辨率

返回值：

3, 支持 1920 x 1080、 1280 x 720、 640 x 480;

2, 支持 1280 x 720、 640 x 480;

1, 支持 640 x 480;

参 数：无

```
jint Java_com_camera_hdmjpegcamera_CameraPreview_getRecordSize(JNIEnv* env, j  
object thiz);
```

用法：

```
public static int recordSize;  
recordSize = getRecordSize();
```

8) *nSetRecordSize*

接口功能：设置录像分辨率，即重新设置预览分辨率

返回值：无

参 数： sizevalue ---

3: 1920x1080

2: 1280x720

1: 640x480

```
void Java_com_camera_hdmjpegcamera_CameraPreview_nSetRecordSize (JNIEnv* env,
jobject thiz, jint sizevalue)
```

用法：

```
private CameraPreview cp;
.....
preview.removeView(cp);
cp.nSetRecordSize(CameraPreview.SIZE720P);
.....
preview.addView(cp);
```

拍照流程

9) *getPictureSize*

接口功能：查询摄像头支持的拍照分辨率

返回值：数组

参 数：无

```
jintArray  
Java_com_camera_hdmjpegcamera_CameraPreview_getPictureSize(JNIEnv* env,  
jobject thiz)
```

用法：

```
public static int[] mPictureSizes;  
mPictureSizes = getPictureSize();
```

拍照分辨率对应表

分辨率	标注	值
320x240	QVGA	0
640x480	VGA	1
800x600	SVGA	2
1280x720	1MP	3
1280x960	1.2MP	4
1280x1024	1.3MP	5
1600x1200	2MP	6
1920x1080	2.1MP	7
2048x1536	3MP	8
2592x1944	5MP	9
3264x2448	8MP	10

10) *nTakePhoto*

接口功能：按指定的分辨率拍照

返回值： 0 --- 完成本次拍照

参 数：

file --- 照片文件名（不用加后缀）

sizevalue --- 拍照分辨率

```
jint Java_com_camera_hdmjpegcamera_CameraPreview_nTakePhoto(JNIEnv* env,  
jobject thiz, jstring file, jint sizevalue)
```

用法：

```
long mRecordingStartTime = System.currentTimeMillis();  
String pictureName = SimpleDateFormat.format(new Date(mRecordingStartTime));  
int ret = cp.nTakePhoto(pictureName, 9);
```


二、参数设置

待补充