

PISOFTECH

圆周率全景相机 Android SDK 使用指南

版本: V0. 9. 0. DOUBLE

Louise

本文主要指导用户如何将本公司的 Android SDK集成到自己的 Android 项目中，及提供 API 说明

目录

一、	功能介绍.....	3
1.	功能列表.....	3
2.	使用限制.....	3
二、	Android studio 如何导入 SDK.....	3
1.	SDK 文件结构.....	3
2.	导入必要文件.....	4
3.	工程配置.....	6
三、	调用 SDK.....	9
1.	初始化.....	9
2.	如何开始预览.....	10
3.	如何查看本地照片.....	10
4.	如何播放视频.....	10
四、	API 说明.....	11
	初始化.....	11
	<i>int init(Activity actvty)</i>	11
	<i>View getPlayerView()</i>	11
	<i>void setSDKIsReadyListener(OnSDKIsReadyListener listener)</i>	11
	实时预览.....	12
	<i>void setPreviewIsReadyListener(OnPreviewIsReadyListener listener)</i>	12
	<i>int startPreview()</i>	12
	<i>int stopPreview()</i>	12
	<i>void setPreviewTextureSize(int width, int height)</i>	12
	<i>SurfaceTexture getPreviewSurfaceTexture()</i>	13
	视频文件播放.....	13
	<i>int openVideo(String url)</i>	13
	<i>void pause()</i>	13
	<i>void Resume()</i>	13
	<i>void stop()</i>	14
	<i>double getVideoProgress()</i>	14
	<i>void seek(double offset)</i>	14
	查看本地照片.....	14
	<i>int openPhoto(String path)</i>	14
	显示效果.....	15
	<i>int setShowMode(int showMode)</i>	15
	其他.....	15
	<i>String getVersion()</i>	15

一、功能介绍

1. 功能列表

功能	描述
实时预览	以多种全景预览模式，观看全景相机拍摄的实时图像
浏览图片	以多种全景观看模式，查看本地保存的全景照片
播放全景视频	以多种全景观看模式，播放本 SDK 录制的全景视频文件

2. 使用限制

软件要求：

只支持 android 5.0 及以上 armeabi-v7a 版本的 android 应用开发。

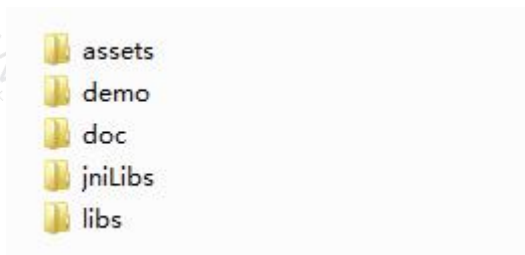
硬件要求：

要求 CPU 四核 2.7GHz ，运行时内存 2GB 同等及以上性能的手机。

二、 Android studio 如何导入 SDK

1. SDK 文件结构

pi_camera_sdk.zip 解压后会得到 libs、assets、jniLibs、demo、doc 三个文件夹，如下图所示：



其中：

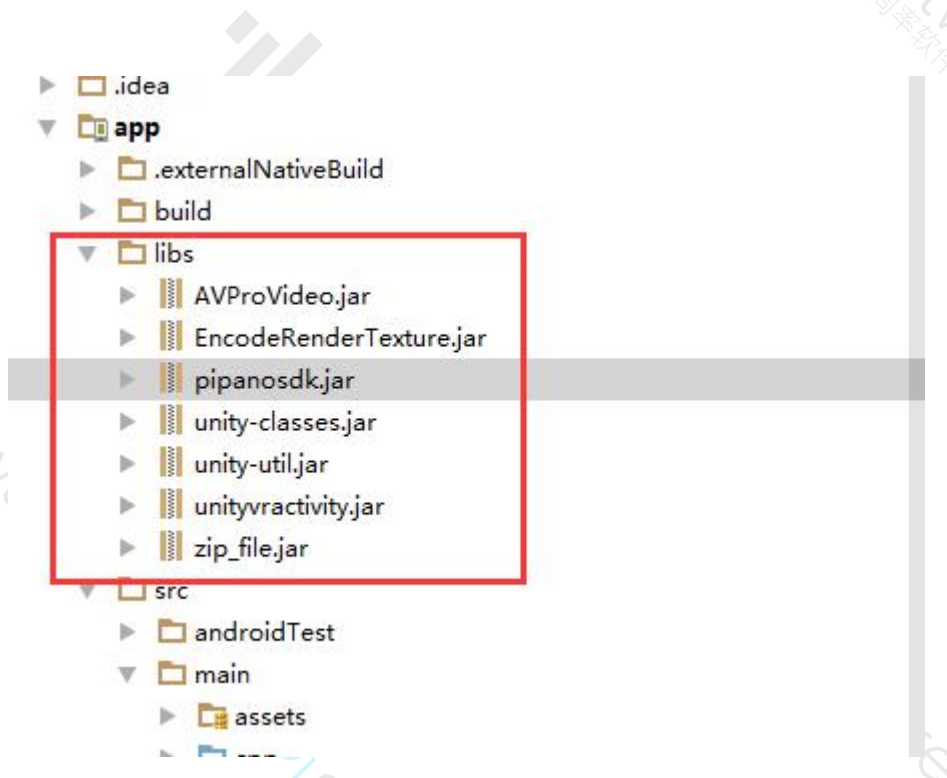
libs、assets 及 jniLibs 文件夹包含的是 android 工程集成 SDK 的必须文件；

demo 包含的是 android studio 调用 SDK 的实例工程；

doc 包含的是 SDK 的帮助文档。

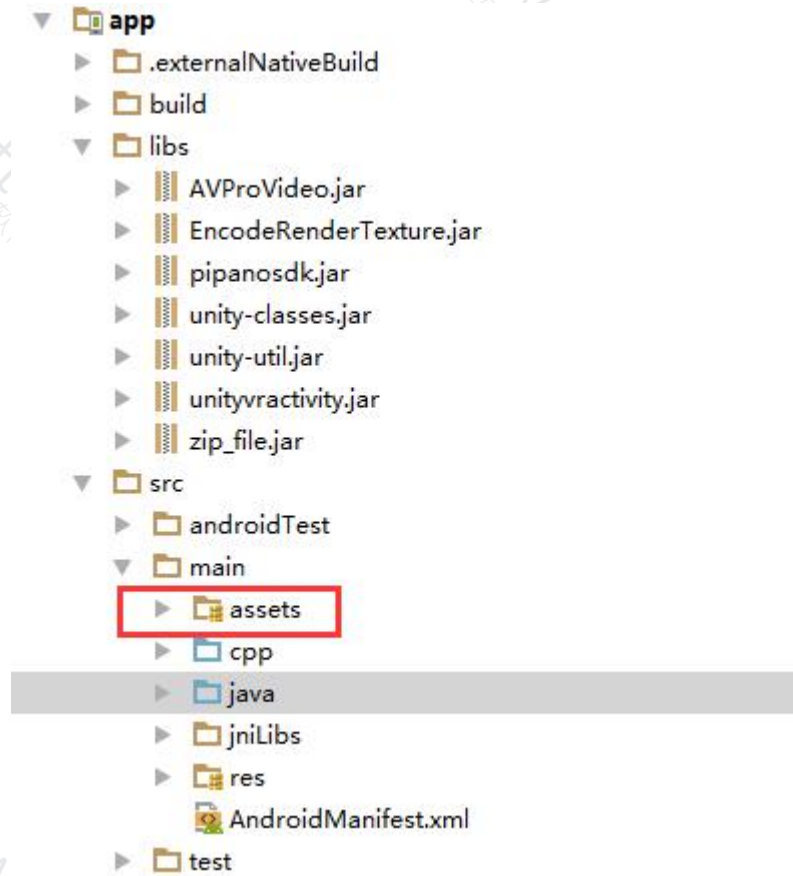
2. 导入必要文件

1) 将 libs 目录下的所有文件复制到 android studio 工程的 libs 目录中，如下图：

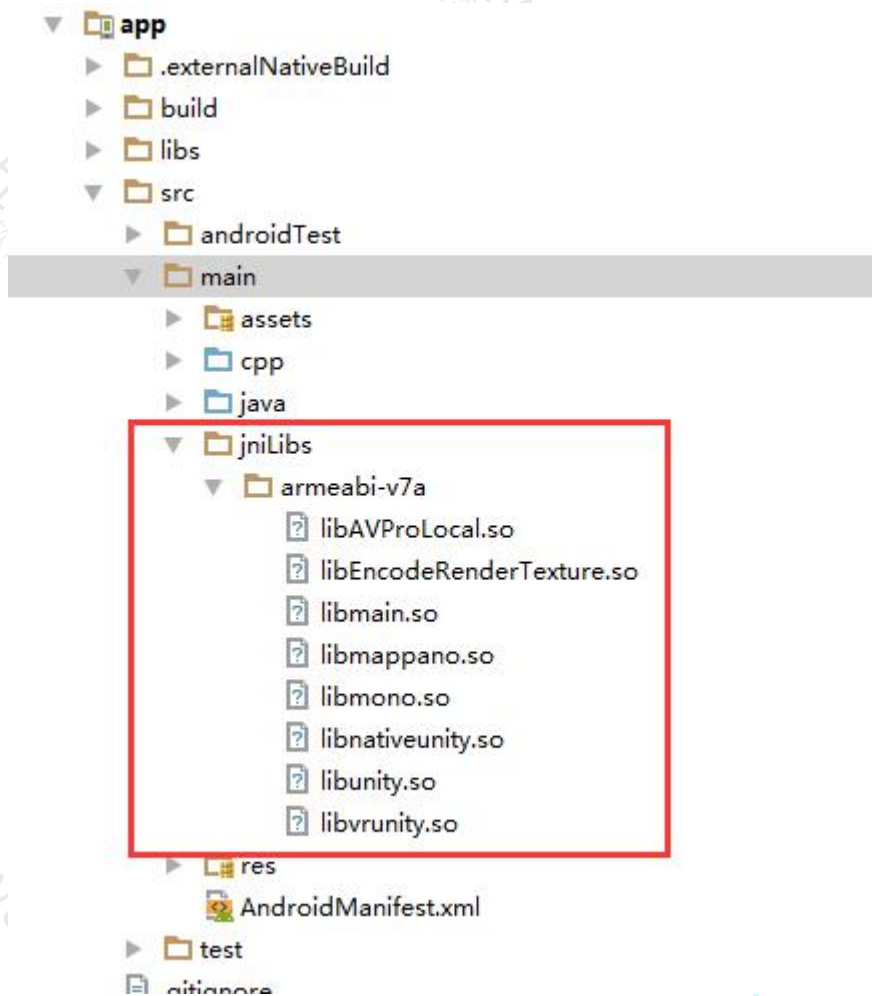


图中文件仅供参考，实际文件会根据使用前版本有所差异。

2) 将 assets 文件夹复制到 android studio 工程的 src/main 目录中，如下图：



- 3) 将 jniLibs 目录下的文件夹复制到 android studio 工程的 src/main/jniLibs 目录(如果没有 jniLibs 目录, 则新建一个)中, 如下图:



图中文件仅供参考，实际文件会根据使用前版本有所差异。

3. 工程配置

用户须要根据实际需求，配置相应的权限。以 demo 中的实例工程为例，拍照、录像须要 SD 卡的读写及音频采集权限，如下图：


```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.pi.demo">
```

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.CAMERA" />
<uses-permission android:name="android.permission.INTERNET" />
```

```
<application
    android:allowBackup="true"
    android:icon="@mipmap/ic_launcher"
    android:label="Demo"
```

在需要嵌入预览或播放器的 layout 文中加入一个播放器的父控件，如 LinearLayout，参考下图：

```
android:layout_weight= 1
android:text="展开"/>
```

```
</RadioGroup>
```

```
<LinearLayout
    android:id="@+id/videoview"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:layout_alignParentBottom="true"
    android:layout_below="@+id/videocontrol"
    android:orientation="horizontal"
    tools:layout_editor_absoluteX="8dp"
    tools:layout_editor_absoluteY="8dp">
```

```
</LinearLayout>
```

```
<LinearLayout
    android:id="@+id/videocontrol"
    android:layout_width="match_parent"
    android:layout_height="wrap_content">
```

在 Activity 的 onCreate() 中加入

```
LinearLayout layout = (LinearLayout) findViewById(R.id. videoview);
layout.addView(panoPlayerView);
```

layout 变量对应上面 layout 中添加的控件，panoPlayerView 变量由 SDK 的 API 中获得，详见[下文](#)。

重载 Activity 中以下函数：

```
@Override protected void onDestroy ()
{
    mPiPanoSDK.onDestroy();

    super.onDestroy();
}

// Pause Unity
@Override protected void onPause ()
{
    super.onPause();
    mPiPanoSDK.onPause();
}

// Resume Unity
@Override protected void onResume ()
{
    super.onResume();
    mPiPanoSDK.onResume();
}

// This ensures the layout will be correct.
@Override public void onConfigurationChanged(Configuration newConfig)
{
    super.onConfigurationChanged(newConfig);
    mPiPanoSDK.onConfigurationChanged(newConfig);
}

// Notify Unity of the focus change.
@Override public void onWindowFocusChanged(boolean hasFocus)
{
    super.onWindowFocusChanged(hasFocus);
    mPiPanoSDK.onWindowFocusChanged(hasFocus);
}
```

mPiPanoSDK 变量由 SDK 的 API 中获得，详见[下文](#)。

三、调用 SDK

1. 初始化

1) 先要在要调用 SDK 的 java 文件中加入

```
import com.pi.RenderTextureToEncoder.TextureListener;
import com.pi.pipanosdk.ErrCode;
import com.pi.pipanosdk.PiPanoSDK;
```

2) 在 Activity 的 onCreate() 中加入创建 SDK 对象

```
mPiPanoSDK = new PiPanoSDK();
```

3) 设置监听 SDK 初始化是否完成

```
mPiPanoSDK.setSDKIsReadyListener(this); // 设置监听 SDK 初始化是否完成
```

4) 加入以下代码初始化 SDK

```
int ret = 0;
ret = mPiPanoSDK.init(this);
if (ret != ErrCode.SUCCESS)
{
    Log.e(TAG, "mPI_PanoPlayer.init() failed!, ret = " + ret);
    return;
}
```

5) 获取 SDK 图像渲染窗口

```
View panoPlayerView = mPiPanoSDK.getPlayerView();
if (null == panoPlayerView)
{
    Log.e(TAG, "null == panoPlayerView");
    return;
}
```

6) 实现接口 PiPanoSDK.OnSDKIsReadyListener 中的 onSDKIsReady(), 监听 SDK 初始化完成

```
public void onSDKIsReady()
{
    mSDKIsOK = false;
    Log.d(TAG, "SDK is ok");
}
```

2. 如何开始预览

- 1) 使用 [setPreviewIsReadyListener\(\)](#) 设置监听预览是否准备就绪；
- 2) 实现 `OnPreviewIsReadyListener()` 方法；
- 3) 调用 [setPreviewTextureSize\(\)](#) 设置 SDK 预览的 texture size；
- 4) 调用 [setShowMode\(\)](#) 设置显示模式；
- 5) 调用 [startPreview\(\)](#) 开始预览；
- 6) `OnPreviewIsReadyListener()` 响应后，调用 [getPreviewSurfaceTexture\(\)](#) 获取 SDK 的 `SurfaceTexture`，然后将图像渲染到此 `SurfaceTexture`，即可预览到图像；
- 7) 在预览过程中可调用 [setShowMode\(\)](#) 切换预览模式，详见 [API 说明](#)；
- 8) 调用 [stopPreview\(\)](#) 停止预览。

3. 如何查看本地照片

- 1) 查看本地照片
调用 [openPhoto\(\)](#) 打开指定的本地照片，并显示在之前关联的控件中。
注意：
 - A. 调用 `openPhoto()` 前必须已调用 `setShowMode()` 设置了显示模式。
- 2) 在照片浏览时，可以调用 [setShowMode\(\)](#) 切换浏览模式。

4. 如何播放视频

- 1) 播放录像
 - A. 调用 [openVideo\(\)](#) 打开视频文件，可以是本地文件也可以是网络 url。
注意：
 - a) 在调用 `openVideo()` 前必须已调用 `setShowMode()` 设置了显示模式。
 - B. 在播放过程中，可以调用 `setShowMode()` 切换浏览模式。
 - C. 调用 [pause\(\)](#) 暂停播放。
 - D. 调用 [resume\(\)](#) 继续播放。
 - E. 调用 [getVideoProgress\(\)](#) 获取当前播放进度百分比(1.0 表示 100%)。

- F. 调用 [seek\(\)](#) 快进到指定的进度百分比开始播放。
- G. 调用 [stop\(\)](#) 停止视频播放。

四、API 说明

初始化

int **init(Activity actvty)**

初始化 SDK

Parameters

actvty 使用 SDK 的 actvty

Returns

返回 0 表示成功就，返回负值表示失败

View **getPlayerView()**

获取 SDK 用于图像渲染的 view，以便部署到 android 工程的 UI 中

Returns

返回 null 表示失败

void **setSDKIsReadyListener(OnSDKIsReadyListener listener)**

给 SDK 绑定 OnSDKIsReadyListener，以便监听 SDK 初始化是否完成

Parameters

listener

Returns

无

实时预览

void setPreviewsReadyListener(OnPreviewsReadyListener listener)

给 SDK 绑定 OnPreviewsReadyListener，以便监听预览是否准备就绪

Parameters

listener

Returns

无

int startPreview()

开始显示预览图像

Returns

返回 0 表示成功就，返回负值表示失败

int stopPreview()

停止显示预览图像

Returns

返回 0 表示成功就，返回负值表示失败

void setPreviewTextureSize(int width, int height)

设置预览 Texture 的 size

Returns

无

SurfaceTexture getPreviewSurfaceTexture()

获取用于渲染预览图像的 SurfaceTexture，用户可以对此 SurfaceTexture 进行操作

Returns

用于渲染预览图像的 SurfaceTexture

视频文件播放

int openVideo(String url)

打开并播放视频文件(支持本地和网络)

Parameters

url 本地视频文件路径或者网络链接

Returns

返回 0 表示成功就，返回负值表示失败

void pause()

暂停播放

Returns

无

void Resume()

继续播放

Returns

无

void stop()

停止播放

Returns

无

double getVideoProgress()

获取视频当前播放进度的百分比

Returns

视频当前播放进度的百分比(范围: 0.0~1.0)

void seek(double offset)

进度跳转

Parameters

offset 表示跳转到视频时长的百分之几的位置(范围: 0.0~1.0)

Returns

无

查看本地照片

int openPhoto(String path)

打开并显示照片

Parameters

path 本地视频文件路径或者网络链接

Returns

返回 0 表示成功就，返回负值表示失败

显示效果

int setShowMode(int showMode)

设置显示模式

Parameters

showMode 对应各种全景显示模式的枚举值(鱼眼、VR、小行星等)
取值范围:

全景模式	值
沉浸式	9
鱼眼	10
小行星	11
VR 分屏	12

Returns

无

其他

String getVersion()

获取当前 SDK 版本号

Returns

当前 SDK 版本号
