Miracast接收端SDK接口文档

文件名称：Miracast接收端SDK接口文档

Android平台
当前版本：V0.9

苏州必捷网络有限公司

修订记录

版本号	拟制/修改人	拟制/修改日期	评审人	修改内容要点
0.9	袁依吉	2019.11.5		初稿

概述

1. 目的

用于指导使用必捷Miracast接收端SDK的开发人员进行开发及测试。

2. 读者对象

本文档适用于开发Android平台Miracast接收端的开发人员。

3. 缩略语定义

缩写名称	英文	中文
Miracast		由Wi-Fi联盟于2012年所制定，以Wi-Fi直连（Wi-Fi Direct）为基础的无线显示标准。

范围

1. 功能

本SDK可以接受来自android，widi使用Miracast方式的投屏处理，并提供接口给应用层调用处理。

2. SDK框架

BJMiracast Receiver SDK总体分为两层

1. cast_base_lib-1.0.20-release.aar：它是一个Android Module，它定义了MediaChannel接口。
2. bj_miracast_lib-1.0.19-release.aar：它是一个Android Module，它定义了MiraProxyBaseImp，MiraProxyModule以及相关JNI接口。

应用程序原则上不应修改cast_base_lib-1.0.20-release.aar，bj_miracast_lib-1.0.19-release.aar中的内容。

3. SDK的DEMO实现

castdemo是接收端的一个参考实现，它基于bj_miracast_lib-1.0.19-release.aar和cast_base_lib-1.0.20-release.aar实现了Miracast接收端功能。
其中MiraProxyModuleImp继承了MiraProxyBaseImp类，实现了对Miracast音频会话的控制。
MiraRenderChannel继承了MediaChannel类，实现了对Miracast音频会话的处理。

4. SDK交付物

- SDK库（两个AAR文件）
- DEMO源代码
- SDK接口文档

接口

接口主要在MiraProxyModule，MiraProxyBaseImp和MediaChannel类中进行定义;
其中MiraProxyModule类提供了初始化SDK，去初始化SDK，强制结束某会话等接口。
MiraProxyBaseImp是一个接口类，需要由用户继承相关接口，SDK通过该接口类通知用户程序Miracast投屏会话的开始与结束，用户程序需要根据不同的会话类型创建对应的MediaChannel实现类和对应的播放器应用。
MediaChannel是一个接口类，需由用户程序实现，SDK通过此接口类将音视频数据，音量控制，视频旋转等控制事件通知应用程序，应用程序在对应播放器中进行处理。
用户需实现MediaChannel，MiraProxyBaseImp相关接口, 也可参考我司提供的DEMO源代码实现。

1. MiraProxyModule类中的接口说明

1. 设置客户定制的模块实现类接口

public void setProxyImp(MiraProxyBaseImp proxyImp);
输入：用户实现的MiraProxyModuleImp接口实例。

2. 初始化接口

public native boolean init(Properties var1);
MiraProxyModule类的Init方法初始化Miracast接收端模块。App在启动做初始化时调用，MiraProxyModule对象应设计为全局只有一个。
返回值为1，表示初始化成功，返回值为0，表示初始化失败。

3. 去初始化接口

public native void fini();
MiraProxyModule类的fini方法去初始化Miracast模块。App销毁Miracast接收端服务时调用。

4. 强制结束某路Miracast会话接口

public void hangup(int channelid);
channelid为MiraProxyBaseImp reqMediaChannel创建的对应会话MediaChannel的id。

5. 关键帧请求

public void reqKeyFrame(int channelid);

6. 延时关键帧请求

public void advReqKeyFrame(int channelId);

2. MiraProxyBaseImp接口说明

1. 会话接入接口

public abstract MediaChannel reqMediaChannel(MediaChannelInfo var1, UserInfo var2, int var3);
输入：会话信息，描述其业务类型。
MediaChannelInfo描述了当前会话的类型信息。
UserInfo描述了发射端的IP，设备型号，设备名称等信息。
int sessionId描述了当前会话的sessionId。
输出：创建的 MediaChannel对象。
这是一个回调接口，当Miracast协议栈发现有会话接入时，JNI层会主动调用此接口，应用层需要实现相关逻辑，需要在客户自身的MiraProxyModuleImp实现类中去实现该逻辑。
具体实现可以参考DEMO源代码。

2. 会话结束接口

public abstract void hangup(int var1);
输入：channelid为MiraProxyBaseImp reqMediaChannel创建的对应会话MediaChannel的id。

3. **没有数据接口**

   public abstract void noMediaNotify(int var1);
   输入：channelid为MiraProxyBaseImp reqMediaChannel创建的对应会话MediaChannel的id。

4. **关键帧请求接口**

   public abstract void reqKeyFrame(int var1);
   输入：channelid为MiraProxyBaseImp reqMediaChannel创建的对应会话MediaChannel的id。

3. **MediaChannel会话处理接口**

当会话建立成功后，协议栈会调用MediaChannel类中的相应接口回吐数据或者获取状态。以下接口需要在MediaChannel子类中实现，客户需要根据自身实际情况实现下列接口。
MediaChannel子类中重要的接口如下，Miracast接收端需要重点关注以下接口。

1. **设置窗口句柄**

   public void setSurface(Surface surface);
   配置窗口句柄,用于MiraRenderChannel输出。Surface由用户创建的播放器视图获取。参考Demo中的实现。

2. **音频数据回吐接口**

   public void onAudioFrame(byte[] buffer, int len, long ts);
   JNI回吐音频数据,音频是PCM格式的数据,用户需要实现对音频数据的播放处理功能. 具体实现可以参考SDK中DEMO源代码。
   其中buffer为音频数据数组，len为数据长度，ts为时间戳。

3. **视频数据回吐接口**

   public void onVideoFrame(byte[] buffer, int len, long ts);
   镜像视频数据接口，该接口中吐出的是H264视频数据，用户需要进行解码并播放。
   其中buffer为视频数据数组，len为数据长度，ts为时间戳。
   具体实现可以参考SDK中DEMO源代码。

4. **绑定buffer池大小**

   public ComBuffer reqBuffer(int size);

5. **打开MediaChannel**

   public abstract boolean open();
   需要用户来实现具体打开MediaChannel功能。参考Demo中的实现。

6. **关闭MediaChannel**

   public abstract void close();
   需要用户来实现具体关闭MediaChannel功能。参考Demo中的实现。

7. **设置音频信息**

   public void notifyTrackCodecInfo(int trackId, AudioTrackInfo audioTrackInfo);
   通过AudioTrackInfo设置音频信息。

Demo说明

Demo中MiraProxyModuleImp实现了MiraProxyBaseImp接口，实现了reqMediaChannel方法。reqMediaChannel中创建了与会话类型对应的MediaChannel实现，并启动播放相关的View，创建用于播放的Surface。MiraRenderChannel实现了MediaChannel中相关功能接口。

客户如何使用SDK

1. 导入aar，在app目录下新建libs目录并将aar文件放在该目录下，然后在build.gradle（app）中dependencies上方及内部分别添加如下代码：repositories{ flatDir { dirs 'libs '} }，compile (name: 'bj_miracast_lib-1.0.19-release.aar', ext: 'aar') compile (name: 'cast_base_lib-1.0.20-release.aar', ext: 'aar')，可参考Demo。
2. 实现MiraProxyBaseImp接口，参考MiraProxyModuleImp。reqMediaChannel接口返回自定义的MediaChannel实现类的实例，并启动播放界面。
3. 实现MediaChannel相关功能接口，参考MiraRenderChannel。
4. 实现自身的播放界面，可参考view这个package中的view的实现，将播放界面的Surface设置到MediaChannel中。