**修订记录**

|  |  |  |  |
| --- | --- | --- | --- |
| **日期** | **修订版本** | **修改描述** | **作者** |
| 2018-10-19 | V1.0.0 | 文档开发初稿 | 叶光锦 |
| 2018-10-22 | V1.0.1 | 增加开发环境 | 叶光锦 |
|  | V1.0.2 |  |  |

# 概述

视频通话是一款可同腾讯云小微进行视频或语音通话的APP。目前支持视频通话与语音通话两大功能。配置了此APP的视频机器人提供授权二维码，移动端通过QQ扫码获得授权后，即可进行通讯。视频机器人提供了语音呼叫、首页人口、联系人界面、消息界面、实体按键等五个入口触发视频通话

# 页面介绍

视频通话完整页面主要包括大小两个窗口以及下方6个按键，如下图，图2.1是为手Q端呼入，为接听状态；图2.2是接通后的状态



* 1. 接听



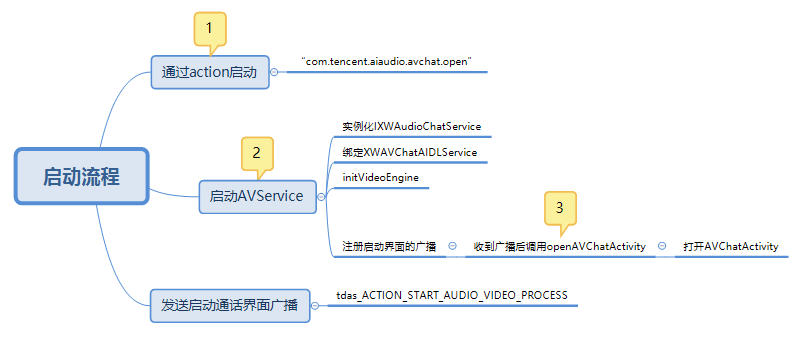
* 1. 接通

大小窗口分别是本地摄像头和手Q端摄像头的预览界面，可互相切换。下面简单介绍下各个按键的作用

1. 移动摄像头，通过此按键可进入二级导航界面，按上下左右即可转动摄像头设备
2. 关闭摄像头，用于控制本地摄像头的开关
3. 接听和挂断，用于接听或挂断音视频通话
4. 静音，控制本地音量的打开或者关闭
5. 窗口切换，用于本地摄像头与远端摄像头的预览界面的切换

这里主要介绍视频通话，关于语音通话，主要是显示上的差异，如移动摄像头、关闭摄像头和窗口切换将被隐藏，自行对比即可

# 启动流程



* 1. 接通

1. 外部通过Action：“com.tencent.aiaudio.avchat.open” 启动通话界面，携带的参数方面需要注意，如需设置“sub\_action”指明是视频通话还是语音通话，详见开发相关
2. 接着启动AVService，负责的工作如下：
3. 实例化IXWAudioChatService，外部可以通过AIDL访问，如AiSupport即是在语音解析后访问这里提供的接听、挂断等方法，以达到语音控制视频通话的效果。可以看出，这个服务是暴露接口给第三方调用
4. 绑定XWAVChatAIDLService，该服务位于AiSupport，实例化IXWAVChatAIDLService，实现了获取Din，获取音频数据等接口，可以看出这个服务是用于从AiSupport中获取资源的
5. 通过VideoController类的“initVcController”初始化音视频引擎
6. 准备完毕后，根据广播“tdas\_ACTION\_START\_AUDIO\_VIDEO\_PROCESS”开启音视频界面，除了必要的View初始化外，还需要关注以下三点：
7. 除了注册Button点击事件，还需要通过AVService的“setIAVOperateListener”提供接听、挂断接口给语音事件
8. 通过HiCameraControllerManager类的“setMotorBySteps”等实现移动摄像头逻辑
9. 注册BroadcastHandler广播，接收手Q端操作事件，包括接听、挂断、开关摄像头等，还有从AiSupport发来的TTS播报结束广播，以及从系统发来的声音通道改变广播和长按返回键事件广播

# 开发相关

## 如何启动APP

通过Action：“com.tencent.aiaudio.avchat.open”启动，需要包含以下参数：

1. Bundle对象，包含“peerid”，Long型参数，表示对方id
2. “sub\_action”，指明是视频通话请求，还是语音通话请求
3. “receive”，True表示呼入；False表示呼出
4. “main\_package”，调用者包名
5. “id”，当此次启动时呼入时，需要带此参数，取值为当前时间即可，用于和呼入时语音提示“接听还是拒绝”TTS的id同步

示例：

Bundle bundle = new Bundle();

bundle.putLong("peerid", peerId);

id = String.valueOf(System.currentTimeMillis());

Intent intent = new Intent();

intent.setAction(mAVPackageName + ".open");

intent.putExtra("sub\_action", action);

intent.putExtra("receive", true);

intent.putExtra("main\_package", mContext.getPackageName());

intent.putExtra("id",id);

intent.putExtras(bundle);

intent.setFlags(Intent.FLAG\_ACTIVITY\_NEW\_TASK);

intent.setPackage(mAVPackageName);

mContext.startActivity(intent);

## 语音相关

调用者通过绑定Action：“com.tencent.aiaudio.RemoteTXAudioChatService”绑定IXWAudioChatService，接着在解析出“挂断”、“接听”等方法内调用IXWAudioChatService提供的方法即可

## 广播相关

### 4.3.1 广播接收

首先，手Q端一些动作，如开关摄像头、开关麦克风，又如网络状态等对应的广播是要注册的，这部分由云小微的SDK提供，用于同手Q端状态同步，具体请参考AudioVideoChatEventController

此外，还需注册以下三个广播：

1. “com.himedia.aisupport.tts\_end”，TTS播报结束广播，在被呼叫时，需要播报一段TTS，类似“收到xxx的电话，接听还是拒绝”，需要获取到播报结束状态，用于继续播放铃声
2. “com.hitv.sound.AUDIO\_PORT\_CHANGED”，声音通道改变广播，用户通过设置改变声音通道后，发出此广播，这里则在接收后，执行对应的切换动作
3. “com.himedia.KEY\_LONG\_BACK”，长按返回键广播，由系统发出，用于通话界面同首页小窗口互相切换动作

### 4.3.2 广播发送

包括申请TTS、唤醒、取消对话悬浮窗等三个对外广播

1. “com.himedia.aisupport.request\_qqcall\_tts”，申请TTS广播，被呼叫时通过此广播申请TTS语音提示
2. “com.himedia.aisupport.wake\_up”，唤醒广播，TTS播报结束后，通过此广播唤醒小微
3. “com.himedia.aisupport.sleep”，在语音交互应该结束的地方，通过此广播关闭对话悬浮窗

## 硬件设备控制

该APP涉及到的硬件设备有两种：LED灯控制，摄像头转动控制：

1. LED灯控制，操作系统变量“persist.sys.led.mode”，写入对应的值
2. 摄像头转动控制，初始化阶段在类HiCameraControllerManager绑定服务“HiCameraControllerService”，通过方法“setMotorBySteps”实现单步转动，“setMotorStart”实现连续转动

## 其他

Sourcecode：

<http://172.20.4.122/svn/hi3716/Apps/AiProj/trunk/src/AiSupportNew/qqvideo>

编译环境：

Android Studio 3.0

签名：

test.jks