## APP端

**1，发送对讲指令**

N

**2，建立通道成功？**

Y

**3，采集音频数据，编码**

**4，发送音频数据**

N

**5，是否有异常？**

Y

**6，结束**

**PS: 如果不确定设备支持什么格式的音频，APP可以先发IOTYPE\_USER\_IPCAM\_GETAUDIOOUTFORMAT\_REQ ​询问。**

1，APP 发 IOTYPE\_USER\_IPCAM\_SPEAKERSTART 告知开启对讲，以及使用的IOTC Channel ID:

 int nChSpeech = 1;//需是空闲的通道，如果不确定，可以调IOTC\_Session\_Get\_Free\_Channel获取一个空闲通道

​avSendIOCtrl(m\_nAVIndex, IOTYPE\_USER\_IPCAM\_SPEAKERSTART,(char\*)&nChSpeech, 4);

2， APP发送成功后，调用avServStart3()建立通道:

avIndex = avServStart3(...nChSpeech...)

3，采集声音，编码

4，调用avSendAudioData()发送音频数据：

avSendAudioData(avIndex, out\_buff, nLen, frameInfo, sizeof(FRAMEINFO\_t));

5，通过返回值确认是否有异常，以及响应UI层的操作。

6，结束：如果是正常开起了对讲，则APP端先停止采集和发送数据，然后调用avServStop()关闭avIndex，然后再发送IOTYPE\_USER\_IPCAM\_SPEAKERSTOP

。如果没有正常开启对讲，则直接发送IOTYPE\_USER\_IPCAM\_SPEAKERSTOP

## 设备端

**1，接收对讲指令**

N

**2，建立通道成功？**

Y

**3，接收音频数据，送去播放**

N

**4，是否有异常？**

Y

**5，结束**

1. 接收到IOTYPE\_USER\_IPCAM\_SPEAKERSTART 后取出nChSpeech。
2. 调用avClientStart2()获取avIndex：

​avIndex = avClientStart2(...,nChSpeech... )

1. 调用avRecvAudioData()接收音频数据，送去播放。
2. 通过返回值判断是否有异常，如果有异常，则终止
3. 结束：正常对讲中，设备端收到​IOTYPE\_USER\_IPCAM\_SPEAKERSTOP之后，停止接收音频数据，调用avClientStop关闭avIndex。否则，直接结束。