# 音画同步

## 误差来源

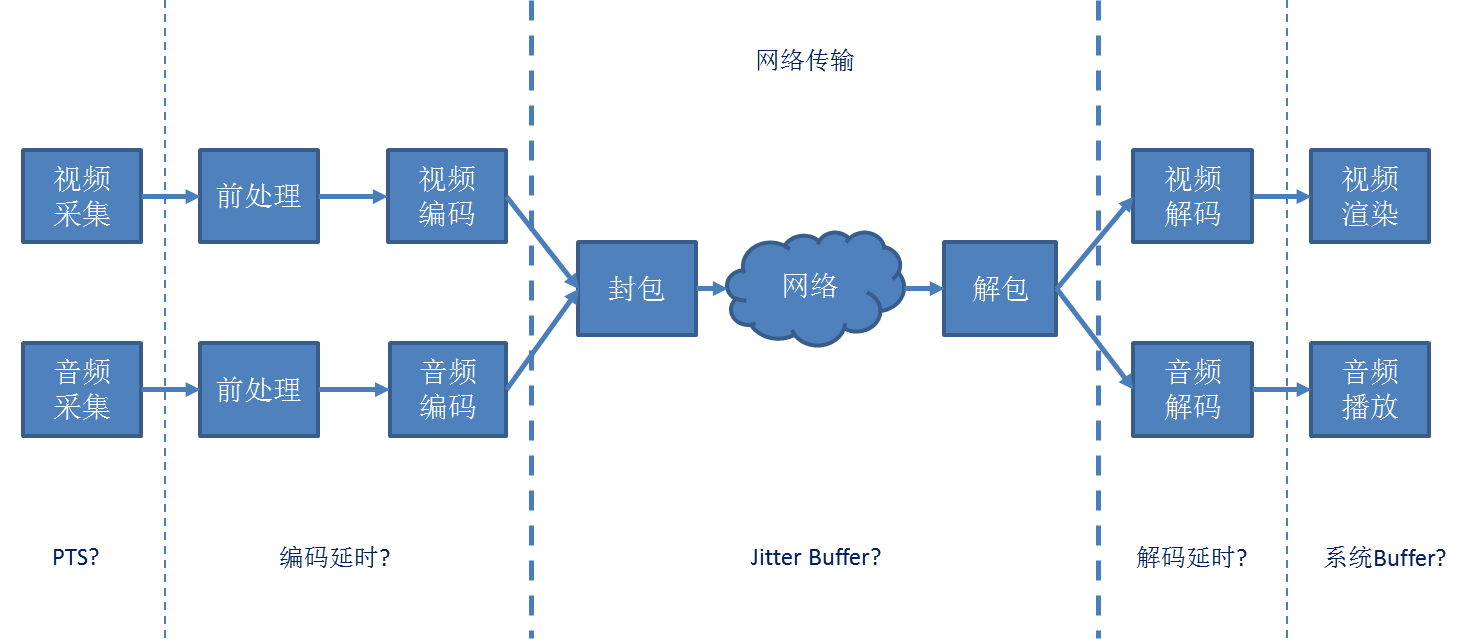


图 1 简单的直播技术

直播方案构架基本如上图所示，不同步的基本原因有硬件固有特性、人为引入。人为引入主要在以下环节：采集、编码、传输、解码、同步算法、渲染。

## 音画同步标准

ITU标准组织已经定义了范围：

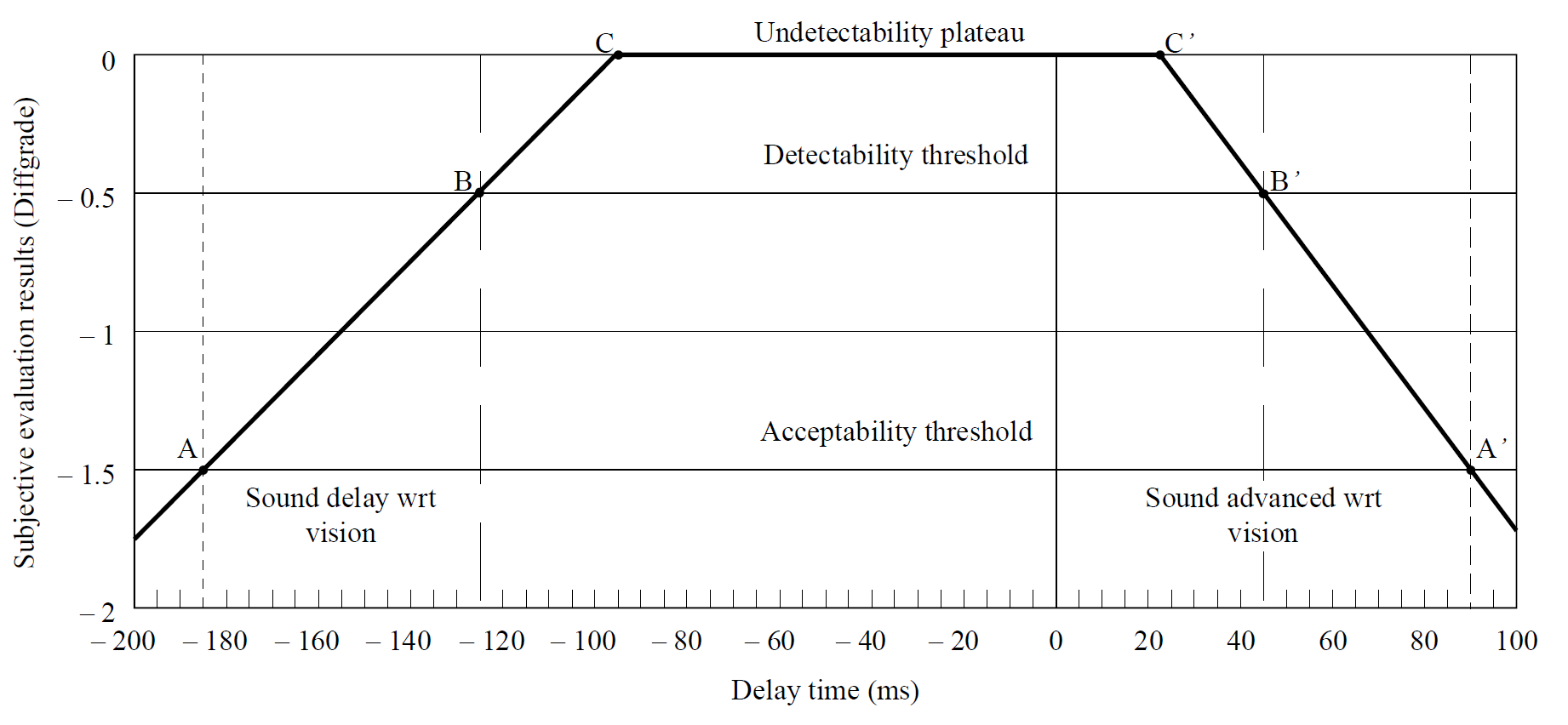


图 2 ITU.BT.1359同步范围

即人耳听到声音和看到视频的时间差最小保持在 -90 ms ～ 180 ms，即声音比视频超前90ms，或声音比视频落后180ms。

## 误差来源检查点

1. 采集端
2. 音视频采集时间戳由系统提供的sample里面提供（Android：Camera2 CaptureResult::SENSOR\_TIMESTAMP，iOS：CMSampleBufferGetPresentationTimeStamp，Windows：IMediaSample.GetMediaTime）
3. 图像前处理需要较长时间，视频时间戳与此无关，不得受到影响
4. 把第1步的时间戳映射到系统时钟中，增加检测预防丢帧/错误导致的时间误差
5. 对于抓屏类应用，时间戳应以抓屏回调第一时间作为时间戳
6. 编码端
7. 音频AAC编码会带来延时，需要校正（SILK约为10ms; FDK AAC约为210ms; AAC Plus约为180ms）
8. 视频需要保证送进去的时间戳和打包的时间戳一致性
9. 传输端
10. FLV的compositiontime是相对时间24 bit有符号数字
11. 保证没有修改时间戳
12. 解码端
13. 解码端保证输出的时间戳和输入的一致
14. 同步算法
15. 同步算法应该使用解码后的音视频时间戳作为同步的依据。如果不能，请增加时间偏移，尽量避免这种情况的发生。
16. 当受到网络状况影响，视频发生滞后时，同步应该丢弃视频帧，尽快追上音频的播放速度。
17. 渲染
18. 尽量与垂直同步，避免撕裂、丢帧等情况。（如60fps的显示器来说，16ms内（一个垂直同步周期内）的所有数据都只渲染一次，丢5帧进去，也就当一帧渲染）。
19. 音频渲染时低层的缓存长度需要告诉同步算法，做为同步依据

# 测试设备

## 工作原理

声音信号是连续的，可以被直接记录的，但视频信号是离散的，不连续，记录时也是离散的，所以直接测量会引入的误差较大，所以需要把这样的信号转换为连续信号 – 光电转换。于是制造了如下设备

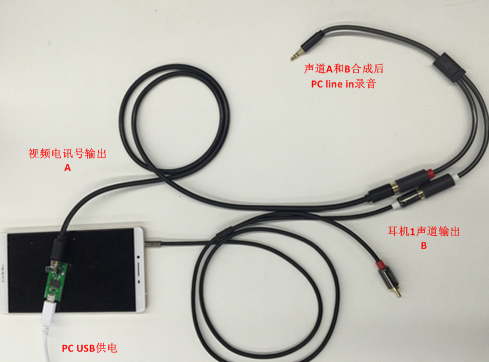


图 3 同步测试硬件

其中声音的两个通道，分别代表视频和音频信号。

开播端使用以下网页提供的测试信号进行开播测试，在观看端进行测量即可得到结果：

<http://www.sync-one2.co.uk/support/test-files/>

这个测试信号的特征如图 4图 5所示：



图 4 普通信号



图 5 同步信号

音频与此类似，同步信号为1 KHz的音频

## 使用说明

测试硬件通过音源输入的方式连接到电脑，左右声道分别用于测试音频、视频同步信号出现的时间。

1. 首先把采集设备插入电脑中，并把接口设置为音源输入，如下图所示（音频需要安装驱动）：

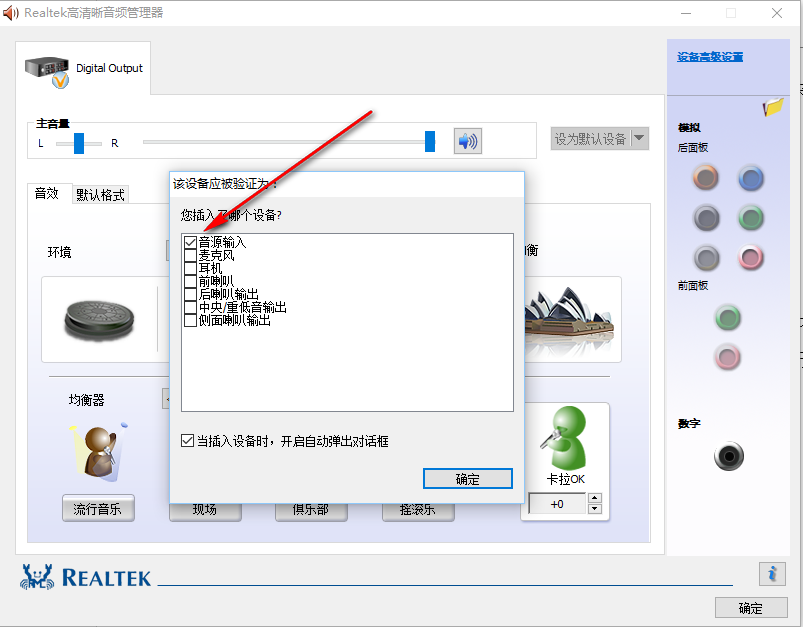


图 6 音源输入设置

1. 打开任意录音软件，把音源输出作为默认录音设备；
2. 把光电转换设备旋转于视频信号上方(把屏幕调节到最高亮度)，把声音接口接到声音（耳机）接口中（音量调节到最大）；
3. 打开观看端，能观察到信号（看到白色闪屏、beep声音）；
4. 录音软件开始录音，即可看到如图 7所示信号，结果如图 8所示。

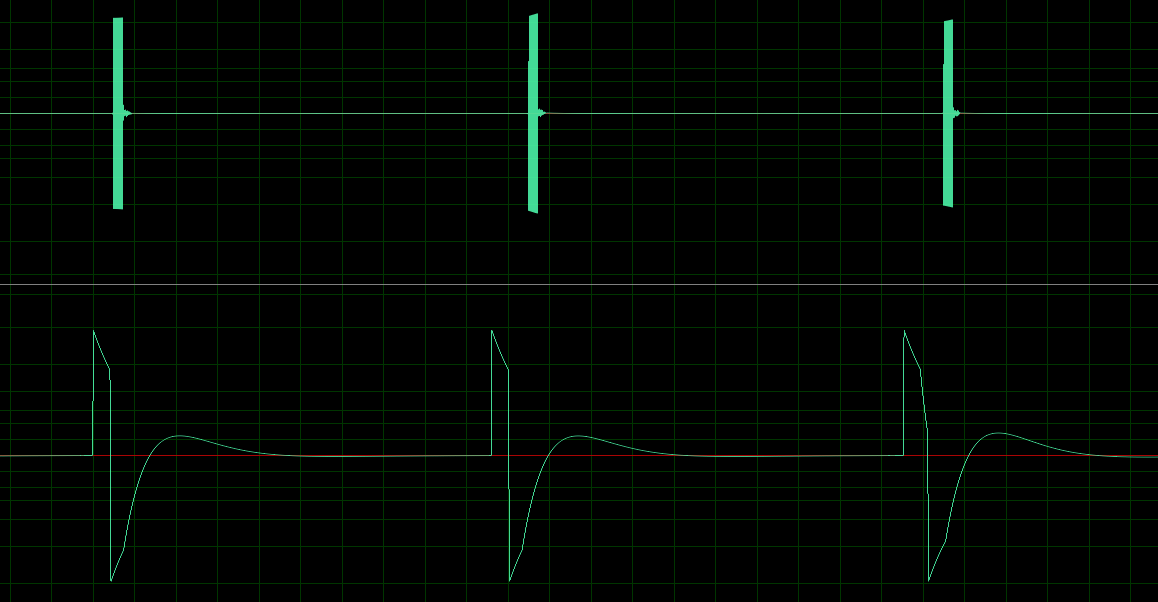


图 7 录音的同步信号

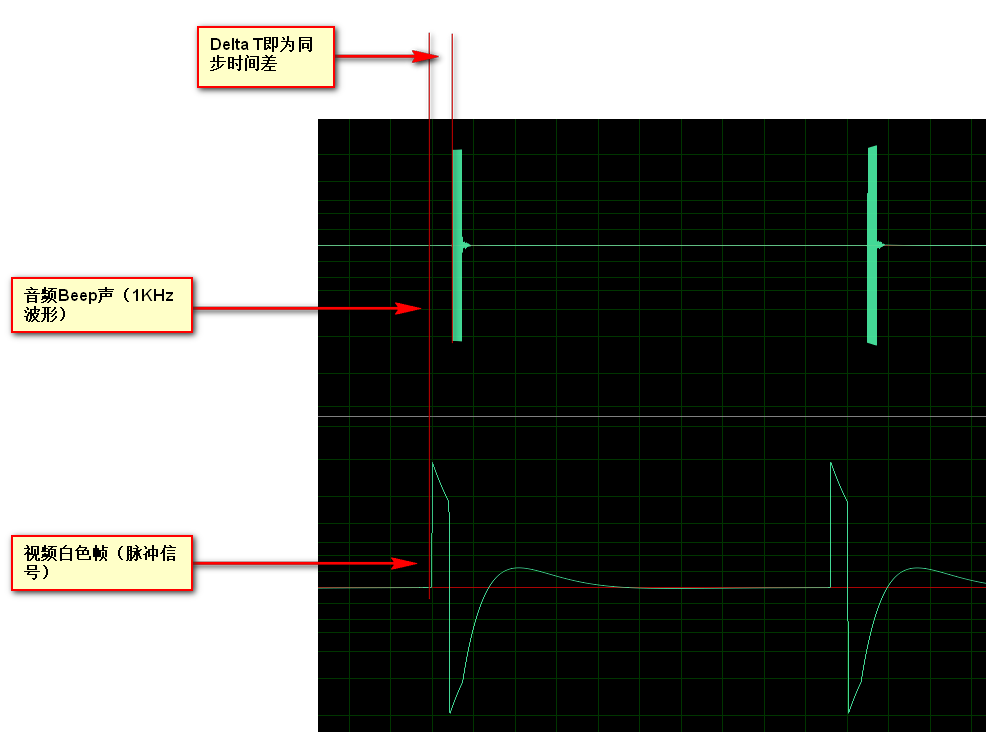


图 8 同步信号解释

## 结果分析

将同步信号放大后，如图 9所示，上方为音频信号，下方为视频信号，如图 10所示，视频同步信号出现的时间为T1，音频同步信号出现的时间为T2，T1 < T2，说明音频比视频落后, T1 – T2即为结果。

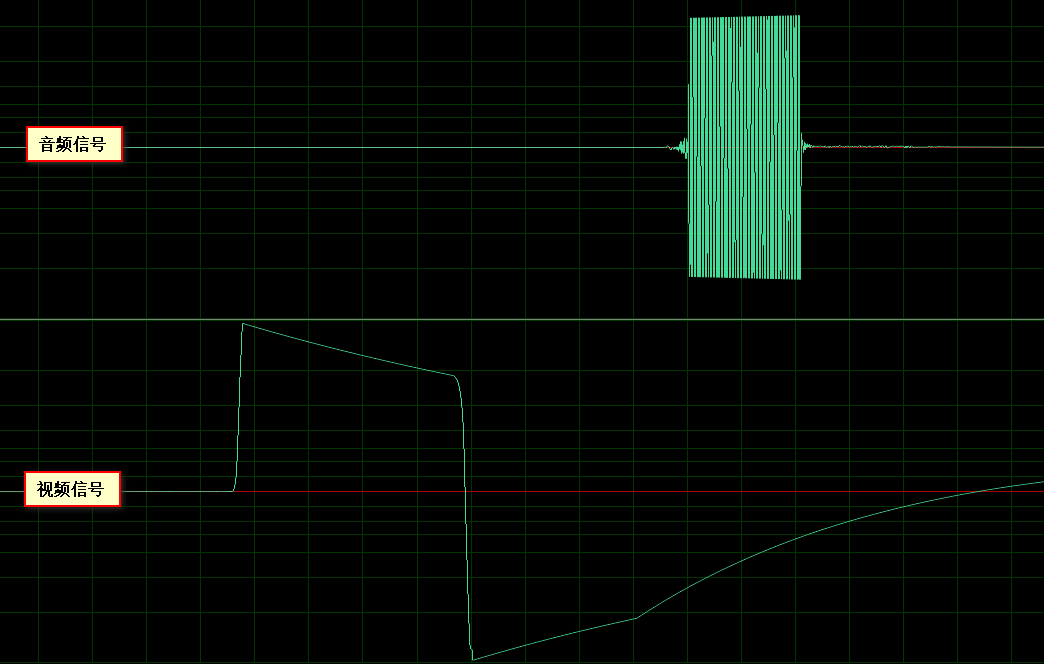


图 9 放大后的同步信号

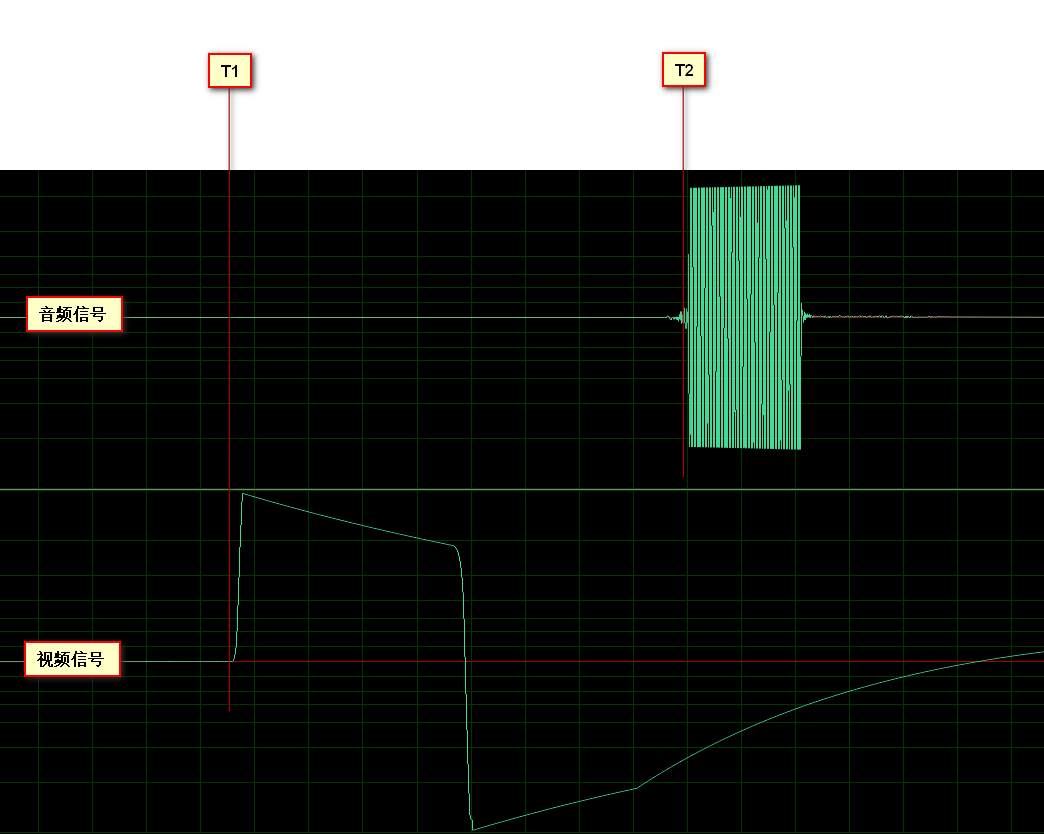


图 10 同步时间差

# 测试方法及步骤

Window 7以上使用media player，暴风影音、mpc-hc等播放器播放测试文件可以保证完全同步(+-10ms以内)。Android手机厂家众多，请挑选有代表性的机器固定下来用于测试。

## OBS

测试OBS及相近软件（如YY侣伴游戏版OBS开播）的音画同步情况

### 测试步骤

1. 自建RTMP服务器
2. 使用OBS播放测试信号
3. 推流到步骤1的服务器地址
4. 使用WEB（flash）进行观看
5. 测试并记录音画同步结果
6. 推流到YY服务器地址
7. 在web端YY频道中观看
8. 测试并记录音画同步结果

### 通过要求

完全同步

### 其它事项

1. 测试通过后，请勿关闭此频道
2. Flash播放地址

<https://www.wowza.com/resources/3.6.0/examples/LiveVideoStreaming/FlashRTMPPlayer/player.html>

## 手Y开播端

### 测试步骤

1. 为Android，iOS手Y制作特殊版本
2. 开播后，通过flash观看测试音画同步情况
3. 使用非特殊版本开播，
4. 测试音画同步情况，记录结果

### 通过要求

步骤2时Web观看音画同步要完全对齐

### 其它事项

1. 步骤1中手Y版本要求：每2秒插入beep声音40ms，第2秒插入2帧白色画面，音视频的时间戳都从1000(毫秒)开始，视频按帧增加，音频按长度增加
2. 测试通过后，请勿关闭此频道

记录结果即是采集端的同步状况，记为Dinput

## 手Y观看端

### 测试步骤

1. OBS端或手Y开播端测试通过后，观看相应YY频道
2. 测试音画同步情况，记录结果

### 通过要求

iOS端要求完全音画同步，Android要在标准范围内

记录结果即是观看端的同步状况，记为Doutput

## PC开播端

### 测试步骤

1. 使用YY伴侣播放测试文件
2. 使用PC模板进行直播
3. 使用flash进行观看

记录结果即是观看端的同步状况，记为DPCinput

## PC观看端

### 测试步骤

1. OBS端或手Y开播端测试通过后，观看相应YY频道
2. 测试音画同步情况，记录结果

### 通过要求

同步在标准范围内

记录结果即是观看端的同步状况，记为DPCoutput

## 回归测试

使用任意平台进行开播，PC、web、手Y进行观看，测试音画同步结果。

# 需求

## 音画同步测试开关

打开开关会生成完全同步的测试信号，用于测试观看端、转码等其它可能引入不同步问题的情况。

当我们有同步相关问题/Bug/方案时，使用该方法开播可以快速验证、定位问题来源。

## 服务器配置

由于硬件有固有延时，不同的厂家、不同机器、不同系统，延时不同，为了尽量减小Dinput、Doutput等值，所以采集端和渲染端都需要有可配置选项，用于指定音画同步误差。

## Dinput、Doutput测试方法

1. 找到测试Doutput标杆

首先测试OBS开播flash观看的同步情况，如果完全同步，使用这个作为测试标杆

如果OBS不同步，尝试使用ffmpeg推流web观看，测试音画同步情况。

1. 测试Doutput

测试步骤1中，观看端的音画同步

1. 测试Dinput

以手Y为例，使用iPhone播放测试信号，手Y采集该信号，在flash端进行观看。测试的音画同步结果即为Dinput