# Demo目的

1. 验证H264编解码功能在目标机型上是否能正常工作：能看到远端画面
2. 验证目标机器摄像头采集是否能正常工作：能看到本地预览
3. 获取目标机器运行Demo时CPU消耗情况：adb shell top命令查看
4. 获取目标机器运行Demo时视频的延时情况：通过录屏及log分析
5. 获取目标机器运行Demo时视频帧率：画面右上角显示数据

# Demo运行说明

1. Demo需运行在两台电视，两台电视处于同一局域网，推荐使用有线连接
2. 两台电视都需要插上支持采集1080分辨率的摄像头
3. Demo使用UDP固定3000端口进行通信
4. Demo有音频处理，如遇啸叫严重，请静音电视

**连接界面：**

Demo打开后进入的第一个页面，如图：

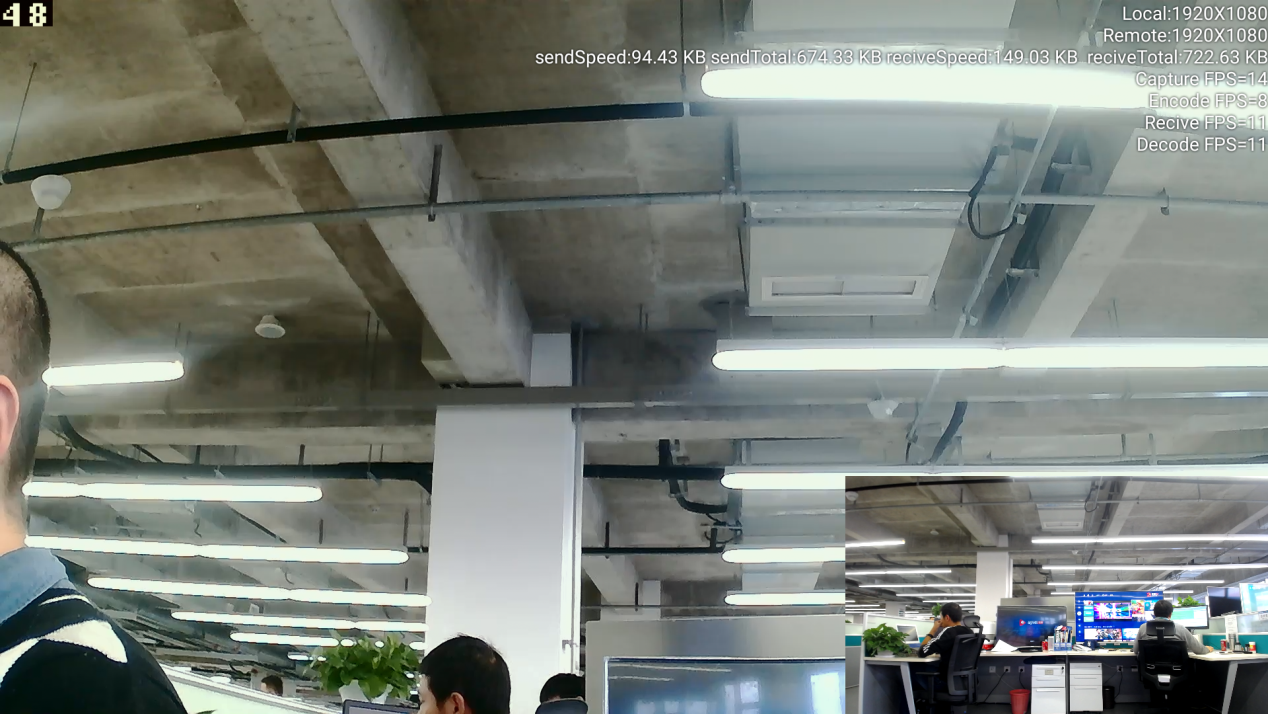


第一栏：IP地址填写对方IP，只填主动连接侧即可

第二栏：分辨率，默认选择摄像头能采集的最高分辨率，可手动切换到低分辨率

第三栏：摄像头采集模式，MStar平台使用YUV模式

点击“连接”按钮后，如果一切正常，两台电视都会切换到视频界面，如图：



**视频显示界面：**

1. 大画面为远端视频
2. 右下角小画面为本地摄像头预览
3. 左上角数字是当前显示的远端画面帧号，从0开始递增
4. 右上角显示部分信息，包括本地采集分辨率、远端采集分辨率、网络收发带宽、本地采集帧率、本地编码帧率、接收远端帧率、解码远端帧率

**异常情况处理：**

1. 无法识别摄像头，界面展示如下：



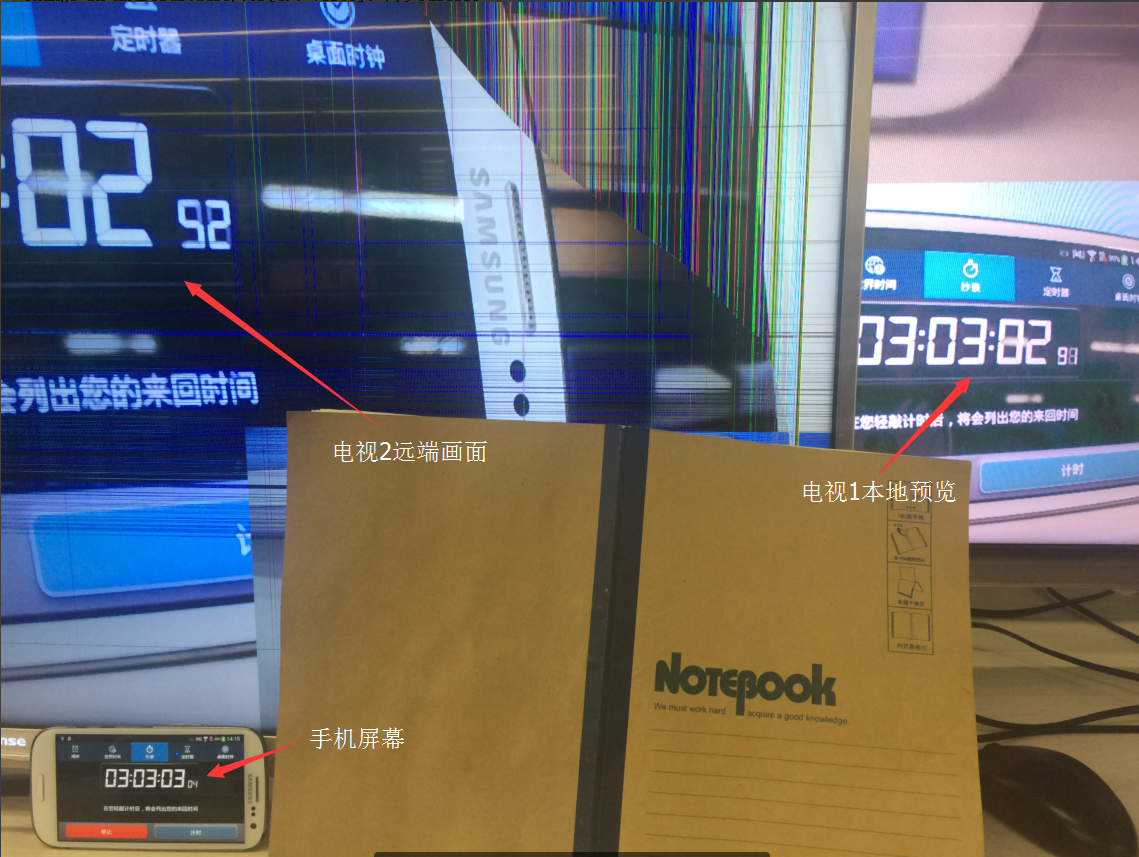
一般通过拔插摄像头、更换USB插口（N8800U只有USB2.0可以识别摄像头）或者重启电视可以修复

1. 其他异常

通过返回键退出所有Demo界面后，Demo会杀掉自己的进程，再次打开Demo一般可修复

# Demo 延时分析说明

准确的延时只能通过录屏来获得，摄像头插在电视1上并对准手机屏幕，另一手机录制手机屏幕+电视1本地预览屏幕+电视2远端画面，如图：



手机屏幕显示时间3.04

电视1本地预览显示时间2.91

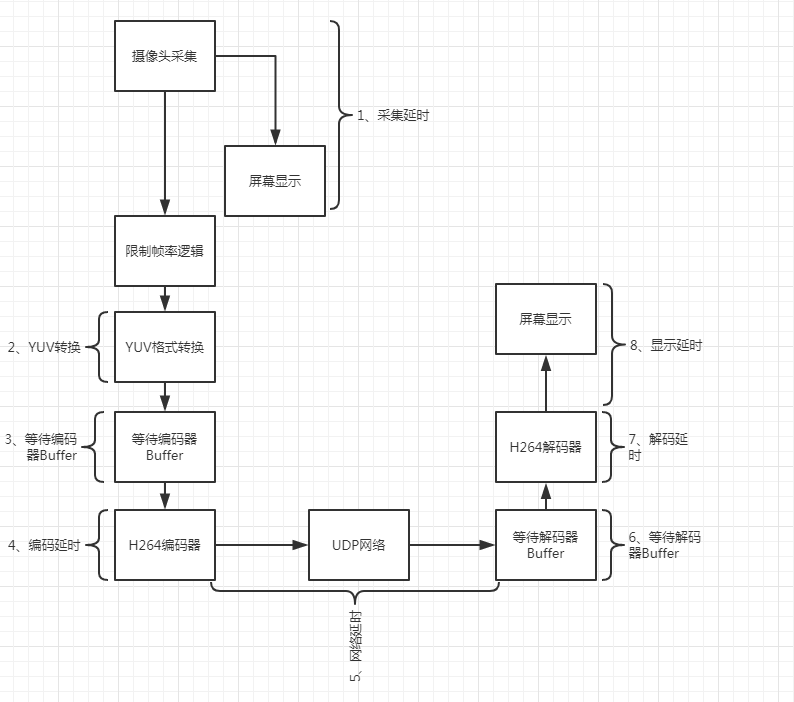
电视2远端画面显示时间2.52

**采集延时** = 手机屏幕显示时间 - 电视1本地预览，3.04-2.91=0.13秒

**总延时** = 手机屏幕显示时间 - 电视2远端画面，3.04-2.52=0.52秒

总延时的分布情况如下图所示：

1. 采集延时
2. YUV转换延时
3. 等待编码器buffer
4. 编码延时
5. 网络延时
6. 等待解码器buffer延时
7. 解码延时
8. 显示延时



上图中2~7步可以通过程序添加log分析得到，具体log如下：

2、YUV转换延时log

D/DelayCalculate: ConvertYUV time=51 avg=23

3、等待编码器buffer

D/DelayCalculate: EncodeQueueInputBuffer time=12 avg=13

4、编码延时

D/DelayCalculate: encode No=1230 time=115 avg=73

5、网络延时

D/DelayCalculate: NetworkDelay time=9 avg=17

6、等待解码器buffer延时

D/DelayCalculate: DecodeQueneInputBuffer time=4 avg=11

7、解码延时

D/DelayCalculate: decode No=2388 time=13 avg=21

上述log每帧打印一次， time 表示当前帧耗时，avg表示平均耗时。

限制帧率逻辑和8屏幕显示延时所造成的延时目前没有很好的办法单独测量。