

H6 多媒体模块

使用说明书

0.8

2017.04.10

文档履历

版本号	日期	制/修订人	内容描述
0.8	2017.04.10		

confidential

目录

1. 前言	1
1.1 编写目的	1
1.2 适用范围	1
1.3 相关人员	1
1.4 相关术语	1
2. 多媒体模块框架和支持列表	3
2.1 Android 多媒体框架	3
2.2 代码结构	4
3. 多媒体支持说明	6
3.1 视频容器格式支持说明	6
3.2 解码格式支持说明	6
3.3 流媒体协议支持说明	6
3.4 多屏互动支持说明	6
3.4.1 airplay 功能支持说明	6
3.4.2 dlina 功能支持说明	7
3.4.3 miracast receiver 投屏支持说明	7
3.5 支持第三方使用 openMAX 的播放器	7
3.6 支持蓝光播放	7
3.7 支持 3D 播放	8
3.8 支持 Deinterlace 处理	8

3.9 支持多音轨切换	8
3.10 支持的其他播放特性	8
4. 模块配置使用说明	9
4.1 最后一帧显示黑屏配置说明	9
4.2 字幕 Native 层显示配置说明	9
4.3 开机视频、动画使用说明	10
4.3.1 开机视频使用说明	10
4.3.2 开机动画使用说明	10
4.3.3 客户开机动画视频定制使用说明	13
4.4 第三方应用播放蓝光使用说明	13
4.4.1 andriod44 平台蓝光使用	13
4.4.2 andriod N 平台蓝光使用	13
5. 调试说明	15
5.1 多媒体中间件如何修改打印等级	15
5.2 使用 MediaPlayer 时，如何简单定位播放问题	15
5.2.1 不能播放定位	15
5.2.2 播放异常定位	17
5.3 使用 MediaPlayer 时，如何保存解码前的码流和解码后的图片	20
5.4 使用 MediaCodec 时，omx_vdec 如何保存解码后的图片	21
5.5 使用 MediaCodec 时，omx_venc 如何保存编码好的数据	22
5.6 使用 MediaCodec 时，如何统计 OMX 解码效率	22
5.7 遇到解码器内部问题，怎么处理	23

6. FAQ	24
6.1 为什么有些音视频不能播放?	24
7. Declaration	25

confidential

1. 前言

1.1 编写目的

为了让多媒体开发人员熟悉 H6 盒子产品的多媒体框架，实现多媒体功能定制和简单调试。

1.2 适用范围

本模块说明适用于全志科技 H6 平台 Android 系统盒子产品。

1.3 相关人员

多媒体开发人员。

1.4 相关术语

- **OMX**: 开放多媒体加速层 Open Media Acceleration，也称 OpenMAX。
- **CedarX**: 全志多媒体中间件，通过 AwPlayer 对接到 android 系统中，架构详见 "2.1 Android 多媒体框架" 章。
- **CedarC**: 全志多媒体音视频编解码库，和 openMAX HAL 层实现。
- **Stream**: CedarX 对多媒体协议类型访问的统一接口，支持的媒体协议类型包括：本地文件、文件描述符、RTSP、UDP、RTP、HTTP、SSL、TCP、RTMP、MMS、MMSH、MMST、MMSHTTP、AES、BDMV 等。
- **Parser**: CedarX 对封装格式的解析的统一接口，支持的媒体封装类型包括：ASF、TS、AVI、FLV、MKV、MOV、DASH、RTSP、HLS、BD、PMP、OGG、MPG、MMS、

MMSHTTP、M3U9、PLAYLIST、MP3、APE、FLAC、AMR、ATRAC、AWTS、REMUX 等。

- Demuxer: CedarX 对媒体的 Stream 和 Parser 解析的统一接口。
- Decoder: 音频, 视频, 字幕的解码器。
- Render: 音频, 视频, 字幕渲染。

confidential

2.2 代码结构

1、Android 多媒体模块，java 层和 jni 层代码目录：

android/frameworks/base/media

- |— java
- |— jni
- |— lib
- |— mca
- |— tests

2、Android 多媒体模块，Native 层代码目录：

android/frameworks/av/media

- |— common_time
- |— libcedarc
- |— libcedarx
- |— libcpustats
- |— libeffects
- |— libmedia
- |— libmediaplayerservice
- |— libnbaio
- |— libstagefright
- |— mediaserver
- |— mtp

3、CedarX 多媒体中间件目录：

android/frameworks/av/media/libcedarx

- |— android_adapter
 - |— awplayer
 - |— iptv
 - |— metadataretriever

```
|   └── output
|   ├── awrecorder
|   ├── config
|   ├── demo
|   ├── document
|   ├── external
|   ├── libcore
|   │   ├── playback
|   │   └── stream
|   ├── xmetadataretriever
|   └── xplayer
```

4、CedarC 多媒体编解码库目录:

```
android/frameworks/av/media/libcedarc
|   ├── base
|   ├── config
|   ├── include
|   ├── library
|   ├── memory
|   ├── openmax
|   │   ├── adec
|   │   ├── libstagefrighthw
|   │   ├── omxcore
|   │   ├── vdec
|   │   └── venc
|   └── vdecoder
```

3. 多媒体支持说明

3.1 视频容器格式支持说明

默认支持如下封装格式：asf, avi, flv, f4v, mkv, mov, mp4/m4v, vob, mpg, pmp, ts/tp, m2ts, mts, wmv, webm, 3GP，此部分内容详见《多媒体模块说明书》。

3.2 解码格式支持说明

默认支持如下视频格式：MPEG1/MPEG2/MPEG4、VC-1/WMV9 SP/MP/AP、MJPEG、H.263 Baseline、H.264 Baseline/HP/MP、H.265、VP6、VP8、VP9、WMV7/8、AVS/AVS+、DIVX，此部分内容详见《多媒体模块说明书》。

3.3 流媒体协议支持说明

默认支持如下流媒体协议：rtsp、http、https、mmsh、mmst、rtmp、mms、udp、hls、rtp，此部分内容详见《多媒体模块说明书》。

3.4 多屏互动支持说明

默认支持 AirPlay、DLNA、Miracast 三大多屏互动协议。

3.4.1 airplay 功能支持说明

AirPlay 是苹果开发的一种无线技术，可以通过 WiFi 将 iPhone、iPad、iPod touch 等 iOS 设备上的包括图片、音频、视频通过无线的方式传输到支持 AirPlay 的 H6 盒子设备上，实时在大屏幕（盒子连接的电视）显示，支持 apk 为乐播投屏。

AirPlay 还有一个功能叫 AirPlay 镜像，该功能可以将 iPhone 或者 iPad 上的画面无线传输到电视上（H6 盒子设备），也就是说你的 IOS 设备显示的是什么电视屏幕显示就就是什么，而不仅限于图片和视频。

3.4.2 dlina 功能支持说明

DNLA, Digital Living Network Alliance, 是索尼、英特尔、微软等发起的一套 PC、移动设备、消费电器之间互联互通的协议。它们的宗旨是“随时随地享受音乐、照片和视频”。DLNA 与苹果的 AirPlay 功能比较类似，协议也大体相同，他们都可以让你手机中的媒体内容投放到电视屏幕里。不同的是手机上的 DLNA 并没有类似 Apple TV 的 AirPlay 的镜像功能，支持 apk 为乐播投屏。

3.4.3 miracast receiver 投屏支持说明

Miracast 是由 Wi-Fi 联盟于 2012 年所制定，以 Wi-Fi 直连为基础的无线显示标准。支持此标准的设备可通过无线方式分享视频画面，例如手机可通过 Miracast 将影片或照片直接在电视或其他装置播放而无需受到连接线缆长度的影响，H6 盒子中支持该功能的 apk 为 MiracastReceiver。

与 DLNA 不同的是，Miracast 也有类似于 AirPlay 的镜像功能，可以将手机中屏幕内容直接投放到高清电视屏幕里，这样你也可以通过电视屏幕来玩游戏了。

3.5 支持第三方使用 openMAX 的播放器

默认支持使用 openMAX IL 层和 openMAX AL 层 Android 原生标准接口的视频播放 apk，比如 Kodi 等。

3.6 支持蓝光播放

默认支持蓝光原盘和蓝光 iso 格式播放，播放双流蓝光时，支持默认 3D 模式切换。

3.7 支持 3D 播放

- 支持上下 3D 播放 1080p@30fps@20Mbps
- 支持左右 3D 播放 1080p@30fps@20Mbps
- 支持双流 3D 播放 1080p@30fps@20Mbps

默认支持。

3.8 支持 Deinterlace 处理

默认支持 de-interlace, 1080i@30fps。

3.9 支持多音轨切换

默认支持多音轨切换。

3.10 支持的其他播放特性

- H.265 4K@60fps@10bit
- H.265 6144*3160@30fps
- H.264 4K@30fps@8bit
- H.264 4K@60fps@8bit
- VP9 4K@60fps@8bit
- VP9 4K@60fps@10bit
- 支持 HDR-10
- 互联网视频最大码率：80Mbps
- 本地视频最大码率：100Mbps

4. 模块配置使用说明

4.1 最后一帧显示黑屏配置说明

最后一帧显示黑屏是指在切台场景，切换中屏幕保持黑屏。修改方案的 CedarX 配置文件：

H6 的配置文件为 `android/frameworks/av/media/libcedarx/config/petrel_config.mk`

```
102 ##---: if need send black frame to gpu for avoid GPU bug
103 LOCAL_CFLAGS += -DCONF_SEND_BLACK_FRAME_TO_GPU
```

此功能配置默认打开。

4.2 字幕 Native 层显示配置说明

字幕 Native 层显示是指字幕直接通过底层显示，而非应用处理显示。该功能在 CedarX 中实现，目前是通过指定播放器应用包名，给指定应用支持。

修改文件 `android/frameworks/av/media/libcedarx/android_adapter/awplayer/awplayer.cpp`

```
diff --git a/media/libcedarx/android_adapter/awplayer/awplayer.cpp b/media/libcedarx/
android_adapter/awplayer/awplayer.cpp
index 683a036..19111f2 100755
--- a/media/libcedarx/android_adapter/awplayer/awplayer.cpp
+++ b/media/libcedarx/android_adapter/awplayer/awplayer.cpp
@@ -58,6 +58,7 @@
static const char *WASU_APP_NAME = "net.sunniwell.app.ott.chinamobile";
static const char *CMCC_LOCAL_APP_NAME = "net.sunniwell.app.swplayer";
// static const char *MANGGUO_APP_NAME = "com.starcor.hunan";
+static const char *CUSTOMER_APP_NAME = "xxx.customer.app.xxxplayer"; /* 增
```

加播放器应用apk包名

```
static int XPlayerCallbackProcess(void* pUser, int eMessageId, int ext1, void* ext2);
static int GetCallingApkName(char* strApkName, int nMaxNameSize);
@@@ -181,7 +182,7 @@ AwPlayer::AwPlayer()

    SubCtrl* pSubCtrl = NULL;

-    if (!strcmp(mPriData->strApkName, CMCC_LOCAL_APP_NAME))
+    if (!strcmp(mPriData->strApkName, CUSTOMER_APP_NAME)) /** 此处判断
走Native层显示
        pSubCtrl = SubNativeRenderCreate();
    else
        pSubCtrl = SubtitleCreate(SubCallbackProcess, this);
```

此功能需要修改代码。

4.3 开机视频、动画使用说明

4.3.1 开机视频使用说明

将视频命名为 **boot.mp4**，放到 **/system/media/** 下或者 **/data/local/** 下。系统启动优先从 **/data/local/** 检测视频文件，如果没有则从 **/system/media/** 下获取，如果两个路径件都没有，则默认使用启动动画。

4.3.2 开机动画使用说明

1. 文件存放位置

开机画面都是以附件 **bootanimation.zip** 的形式存放在机顶盒中。存放位置有两个：

- **/system/media/bootanimation.zip**

- /data/local/bootanimation.zip

读取顺序：机顶盒开机画面或视频，优先读取/data/local/bootanimation.zip，如果发现这个 zip 包没有或者解压有问题或者包中的文件有异常或损坏，则使用/system/media/bootanimation.zip 作为开机画面或开机视频展现。

2. 开机画面的内部结构

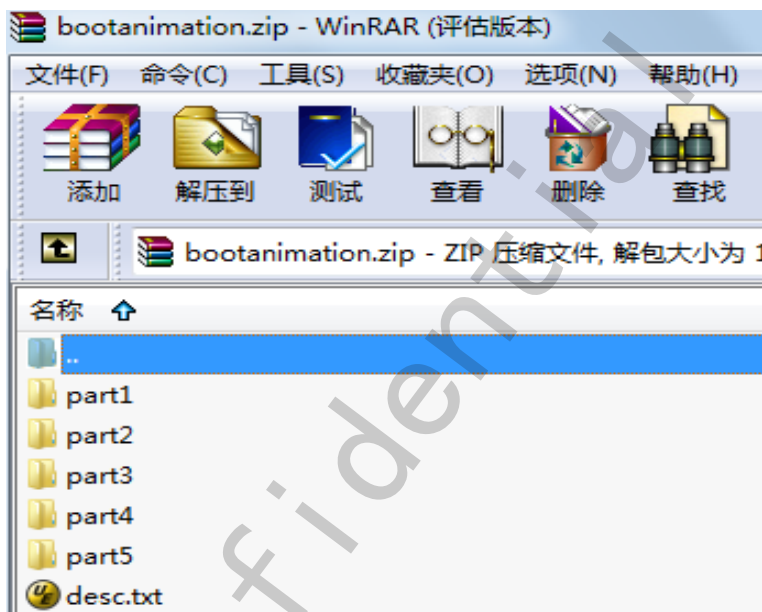


图 2: bootanim01

文件夹中存放图片，/part1 文件夹内部如下：

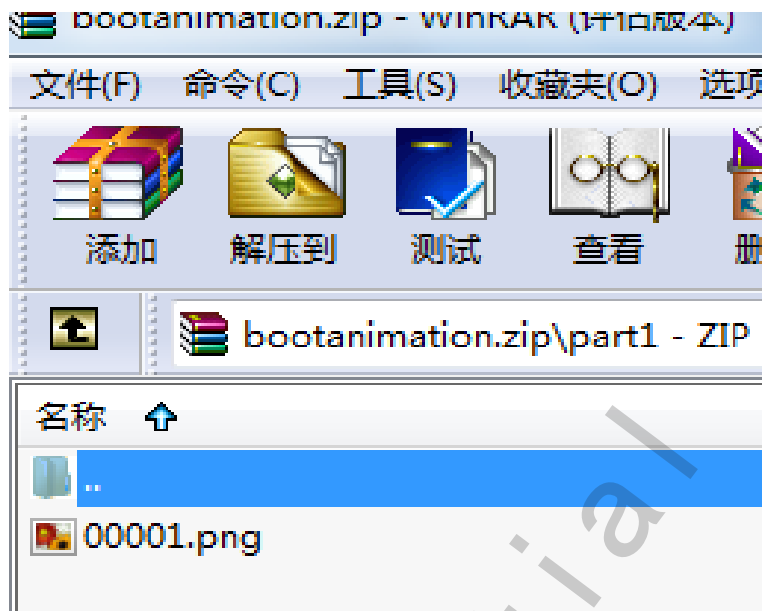


图 3: bootanim02

/desc.txt 文件及内容分析:

```
1 1280 720 1↓
2 ↓
3 p 1 5 part1↓
4 p 1 5 part2↓
5 p 1 5 part3↓
6 p 1 5 part4↓
7 p 0 0 part5↓
8 //
```

图 4: bootanim03

1280 720 是指前面文件夹里 png 的分辨率，1 是指每秒播放帧数；p 是标识符，1 5 两个数字分别指循环次数和阶段间隔时间，0 0 就代表循环播放；part1 就是文件夹的名字，文件夹的名称和存放图片的目录名一致，设计结构：上图/desc.txt 文件设置为：part1-part4 播放一次，间隔 5 秒，part5 循环播放。

4.3.3 客户开机动画视频定制使用说明

设置或者更新开机动画视频时，客户需要自行实现开机动画视频文件下载逻辑（开机动画 bootanimation.zip 或开机视频 boot.mp4），把文件存放到制定目录即可。如果已经有开机动画视频文件，覆盖即可。

4.4 第三方应用播放蓝光使用说明

第三方应用播放蓝光，请参考 TvVideo 播放器应用，SDK 中 TvVideo 应用目录：

- android44: android/vendor/tvd/packages/TvdVideo
- androidN: android/vendor/fvd/package/TvdVideo

4.4.1 android44 平台蓝光使用

- 蓝光原盘文件夹播放

使用方法：

- 1、检查是否为蓝光文件夹，设置播放类型为“bdmv”；
- 2、调用MediaPlayer.setDataSource (“bdmv:///xxx/xxx/blueDirXXX”)。

- 蓝光 iso 文件播放

使用方法：

- 1、先挂载iso；
- 2、再判断是否为蓝光文件夹，设置播放类型为“bdmv”；
- 3、MediaPlayer.setDataSource (“bdmv:///xxx/xxx/blueDirXXX”)。

4.4.2 android N 平台蓝光使用

- 蓝光原盘文件夹播放

使用方法：

- 1、检查是否为蓝光文件夹，设置播放类型为“bdmv”；
- 2、MediaPlayer.setDataSource (“bdmv:///xxx/xxx/blueDirXXX”)。

- 蓝光 iso 文件播放

使用方法：

- 1、TvdVideo播放器应用或者第三方播放器应用,设置蓝光文件路径或fd给MediaPlayer;
- 2、MediaPlayer.setDataSource (“file:///xxx/xxx.iso”)或者 MediaPlayer.setDataSource (fd), 其中fd为open iso对应的fd。

confidential

5. 调试说明

5.1 多媒体中间件如何修改打印等级

针对多媒体音视频播放问题，客户可以修改 CedarX 配置文件打印等级，增加打印：
android/frameworks/av/media/libcedarx/config/petrel_cedarx.conf

```
38 # log will output if level >= log_level
39 #VERBOSE = 2,
40 #DEBUG = 3,
41 #INFO = 4,
42 #WARNING = 5,
43 #ERROR = 6,
44 log_level = 3
```

5.2 使用 MediaPlayer 时，如何简单定位播放问题

遇到播放问题，可以先将 log_level 改为 VERBOSE，反馈给 FAE。

5.2.1 不能播放定位

第一步：播放器有没有创建成功

查看 adb logcat，搜集 CedarX 打印

```
D/awplayer( 1515): <LogVersionInfo:34>:
D/awplayer( 1515): >>>>>> CedarX <<<<<<
D/awplayer( 1515): tag   : CedarX-2.7.0
D/awplayer( 1515): branch: master
```

```
D/awplayer( 1515): commit: 348ce25f100af07dd70b9d927a266a0c83e9c3ce
D/awplayer( 1515): date : Mon, 31 Oct 2016 05:17:22 +0000 (13:17 +0800)
D/awplayer( 1515): author:
D/awplayer( 1515): -----
```

查看 adb logcat，搜集 CedarC 打印

```
D/cedarc ( 1515): <LogVersionInfo:38>:
D/cedarc ( 1515): >>>>>> Cedar Codec <<<<<<<<
D/cedarc ( 1515): tag : CedarC-H6-dev-v1.0.4
D/cedarc ( 1515): branch: h6-dev
D/cedarc ( 1515): commit: ea6449684eb94d90980b01061e4683b16a27c731
D/cedarc ( 1515): date : Thu Jan 12 16:42:02 2017 +0800
D/cedarc ( 1515): author: jenkins
D/cedarc ( 1515): -----
```

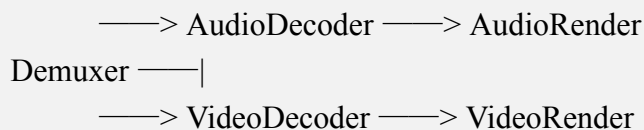
第二步：parser 有没有解析成功

```
D/demuxComponent( 1515): <PrintMediaInfo:426>: *****PrintMediaInfo begin*****
D/demuxComponent( 1515): <PrintMediaInfo:440>: fileSize = 47174592, bSeekable = 1,
duration = 202996, audioNum = 1, videoNum = 1, subtitleNum = 0
D/demuxComponent( 1515): <PrintMediaInfo:458>: **Video[0]** eCodecFormat = 0x115,
nWidth = 720, nHeight = 480, nFrameRate = 24000,
nFrameDuration = 0, bIs3DStream = 0
D/demuxComponent( 1515): <PrintMediaInfo:476>: ***Audio[0]*** eCodecFormat = 0x3,
eSubCodecFormat = 0x0, nChannelNum = 2,
nBitsPerSample = 0, nSampleRate = 44100
D/demuxComponent( 1515): <PrintMediaInfo:492>: *****PrintMediaInfo end*****
```

没有如上类似的打印，说明 parser 没有解析成功。如果需要的话，搜集 CedarC 和 CedarX 打印，反馈给 FAE 定位问题。

5.2.2 播放异常定位

播放的处理流程如下：



播放异常的分析流程与之相反：Render ---> Decoder ---> Demuxer

1、查看 Audio/Video Render 是否正常

android/frameworks/av/media/libcedarx/libcore/playback/player.c 文件

```
diff --git a/media/libcedarx/libcore/playback/player.c b/media/libcedarx/libcore/playback/
player.c
index 5c60b39..76a4440 100755
--- a/media/libcedarx/libcore/playback/player.c
+++ b/media/libcedarx/libcore/playback/player.c
@@ -3191,7 +3191,7 @@ static int CallbackProcess(void* pSelf, int eMessageId,
    void* param)

-    logv("notify video pts = %" PRId64 " ms, curTime = %" PRId64 " ms, diff = %"
+    logd("notify video pts = %" PRId64 " ms, curTime = %" PRId64 " ms, diff = %"
        PRId64 " ms", nVideoPts/1000, nCurTime/1000, nTimeDiff/1000);

    if (p->onResetNotSync)
@@ -3376,7 +3376,7 @@ static int CallbackProcess(void* pSelf, int eMessageId,
    void* param)

    }

-    logv("notify audio pts %" PRId64 " ms, curTime %" PRId64 " ms, diff %" PRId64
+    logd("notify audio pts %" PRId64 " ms, curTime %" PRId64 " ms, diff %" PRId64
        " ms, cacheTime %" PRId64 " ms", nAudioPts/1000, nCurTime/1000,
```

```
nTimeDiff/1000, nCachedTimeInSoundDevice/1000);
```

分析打印，看看 Audio 和 video 的 pts 是否正常。

2、查看 Audio/Video Decoder 是否正常

android/frameworks/av/media/libcedarx/libcore/playback/audioDecComponent.c 文件

```
diff --git a/media/libcedarx/libcore/playback/audioDecComponent.c b/media/libcedarx/
libcore/playback/audioDecComponent.c
index 6c0796b..fd796f3 100755
--- a/media/libcedarx/libcore/playback/audioDecComponent.c
+++ b/media/libcedarx/libcore/playback/audioDecComponent.c
@@ -1139,11 +1139,11 @@ static void doDecode(AwMessage *msg, void *arg)
    }
}

ret = DecodeAudioStream(p->pDecoder,
                        &p->pStreamInfoArr[p->nStreamSelected],
                        pOutputBuf,
                        &nPcmDataLen);
- logv("DecodeAudioStream, ret = %d", ret);
+ logd("DecodeAudioStream, ret = %d", ret);
if (ret == ERR_AUDIO_DEC_NONE)
{
    if (p->pStreamInfoArr[p->nStreamSelected].nSampleRate !=
    p->bsInfo.out_samplerate ||
```

android/frameworks/av/media/libcedarx/libcore/playback/videoDecComponent.c 文件

```
diff --git a/media/libcedarx/libcore/playback/videoDecComponent.c b/media/libcedarx/
libcore/playback/videoDecComponent.c
index 4dfbf8a..ca3294e 100755
--- a/media/libcedarx/libcore/playback/videoDecComponent.c
```

```
+++ b/media/libcedarx/libcore/playback/videoDecComponent.c
@@@ -866,13 +866,13 @@@ static void doDecode(AwMessage *msg, void *arg)

    int64_t nCurTime = p->pAvTimer->GetTime(p->pAvTimer);

    int ret = DecodeVideoStream(p->pDecoder,
                                p->bEosFlag,
                                p->bConfigDecodeKeyFrameOnly,
                                p->bConfigDropDelayFrames,
                                nCurTime);

    - logv("DecodeVideoStream return = %d, p->bCrashFlag(%d)", ret, p->bCrashFlag);
    + logd("DecodeVideoStream return = %d, p->bCrashFlag(%d)", ret, p->bCrashFlag);
```

分析解码库的返回值意义：

- VDECODE_RESULT_FRAME_DECODED(1): 解码成功，输出了一帧图像；
- VDECODE_RESULT_CONTINUE(2): 码流被解码，但没有图像输出，需继续解码；
- VDECODE_RESULT_KEYFRAME_DECODED(3): 解码成功，输出了一帧关键帧图像；
- VDECODE_RESULT_NO_FRAME_BUFFER(4): 当前无法获取到图像Buffer；
- VDECODE_RESULT_NO_BITSTREAM(5): 当前无法获取到码流数据；
- VDECODE_RESULT_RESOLUTION_CHANGE(6): 视频分辨率发生变化；
- VDECODE_RESULT_UNSUPPORTED(-1): 不能支持的格式或申请内存失败，无法继续解码；

2、查看 Demuxer 给解码库 submit Audio/Video 数据是否正常

android/frameworks/av/media/libcedarx/libcore/playback/player.c 文件

```
diff --git a/media/libcedarx/libcore/playback/player.c b/media/libcedarx/libcore/playback/
player.c
index 5c60b39..e191cc6 100755
--- a/media/libcedarx/libcore/playback/player.c
```



```
+++ b/media/libcedarx/libcore/playback/player.c
@@@ -1356,7 +1356,8 @@@ int PlayerSubmitStreamData(Player*      pl,

    p = (PlayerContext*)pl;

-   logv("submit stream data, eMediaType = %d", eMediaType);
+   logd("submit stream data, eMediaType[%d] pts[%lld] nStreamIndex[%d]",
+       eMediaType, pDataInfo->nPts, nStreamIndex);

    if(p->eStatus == PLAYER_STATUS_STOPPED)
    {
```

根据这句打印，判断Demuxer后的音视频数据是否送给解码库，每笔数据的PTS。其中 eMediaType 含义如下。

```
enum EMEDIATYPE
{
    MEDIA_TYPE_VIDEO = 0,
    MEDIA_TYPE_AUDIO,
    MEDIA_TYPE_SUBTITLE
};
```

经过以上三步分析，可以初步定位出播放异常问题属于 Render ---> Decoder ---> Demuxer (Audio/Video/Subtitle) 哪个模块。

5.3 使用 MediaPlayer 时，如何保存解码前的码流和解码后的图片

android/frameworks/av/media/libcedarc/vdecoder/vdecoder.c 文件

```
diff --git a/media/libcedarc/vdecoder/vdecoder.c b/media/libcedarc/vdecoder/vdecoder.c
index 47c1cdd..9346fdc 100755
--- a/media/libcedarc/vdecoder/vdecoder.c
+++ b/media/libcedarc/vdecoder/vdecoder.c
@@ -32,8 +32,8 @@
#include <sys/ioctl.h>
#include <fcntl.h>

#define DEBUG_SAVE_BITSTREAM (0)
#define DEBUG_SAVE_PICTURE (0)
#define DEBUG_SAVE_BITSTREAM (1) /* 保存解码前的码流
#define DEBUG_SAVE_PICTURE (1) /* 保存解码出来的图片

/* show decoder speed */
#define AW_VDECODER_SPEED_INFO (0)
```

解码出来的图片，根据格式使用 YUV 或 NV 工具查看。

5.4 使用 MediaCodec 时，omx_vdec 如何保存解码后的图片

android/frameworks/av/media/libcedarc/openmax/vdec/omx_vdec_newDisplay.cpp 文件

```
diff --git a/media/libcedarc/openmax/vdec/omx_vdec_newDisplay.cpp b/media/libcedarc/
openmax/vdec/omx_vdec_newDisplay.cpp
index 6a80535..17769fb 100755
--- a/media/libcedarc/openmax/vdec/omx_vdec_newDisplay.cpp
+++ b/media/libcedarc/openmax/vdec/omx_vdec_newDisplay.cpp
@@ -70,7 +70,7 @@
#define debug logi("LINE %d, FUNCTION %s", __LINE__, __FUNCTION__);
```

```
#define OPEN_STATISTICS (0)
#define SAVE_PICTURE (0)
#define SAVE_PICTURE (1)
```

5.5 使用 MediaCodec 时，omx_venc 如何保存编码好的数据

android/frameworks/av/media/libcedarc/openmax/venc/omx_venc.cpp 文件

```
diff --git a/media/libcedarc/openmax/venc/omx_venc.cpp b/media/libcedarc/openmax/
venc/omx_venc.cpp
index 99dbe25..d943838 100755
--- a/media/libcedarc/openmax/venc/omx_venc.cpp
+++ b/media/libcedarc/openmax/venc/omx_venc.cpp
@@ -52,7 +52,7 @@ using namespace android;

#endif

#define SAVE_BITSTREAM 0
#define SAVE_BITSTREAM 1
```

5.6 使用 MediaCodec 时，如何统计 OMX 解码效率

android/frameworks/av/media/libcedarc/openmax/vdec/omx_vdec_newDisplay.cpp 文件

```
diff --git a/media/libcedarc/openmax/vdec/omx_vdec_newDisplay.cpp b/media/libcedarc/
openmax/vdec/omx_vdec_newDisplay.cpp
index 6a80535..0cd303e 100755
--- a/media/libcedarc/openmax/vdec/omx_vdec_newDisplay.cpp
```

```
+++ b/media/libcedarc/openmax/vdec/omx_vdec_newDisplay.cpp
@@ -69,7 +69,7 @@
#endif

#define debug logi("LINE %d, FUNCTION %s", __LINE__, __FUNCTION__);
-#define OPEN_STATISTICS (0)
+#define OPEN_STATISTICS (1)
#define SAVE_PICTURE (0)
```

5.7 遇到解码器内部问题，怎么处理

打开 special.data 保存宏，把保存的数据反馈给 FAE。注意：如果保存失败，请查看当前盒子是否有 "/data/camera/" 目录，权限是否正确。

```
diff --git a/media/libcedarc/vdecoder/vdecoder.c b/media/libcedarc/vdecoder/vdecoder.c
index 47c1cdd..df86dc2 100755
--- a/media/libcedarc/vdecoder/vdecoder.c
+++ b/media/libcedarc/vdecoder/vdecoder.c
@@ -42,7 +42,7 @@

#define TRANSFORM_DEV_TIMEOUT 50 /* ms , tr timeout value

-#define DEBUG_MAKE_SPECIAL_STREAM (0)
+#define DEBUG_MAKE_SPECIAL_STREAM (1)
#define SPECIAL_STREAM_FILE "/data/camera/special.awsp"
```

6. FAQ

6.1 为什么有些音视频不能播放？

1、判断封装格式、流媒体协议是否支持

2、判断音频/视频的 Codec 格式是否支持

有些音频/视频格式，属于版权视频，需要查看支持列表，看看是否支持。如果需要支持，请咨询 FAE。

比如：Audio 格式：ac3, dts 等 Video 格式：DIVX, rmvb 等

3、判断视频 Codec 规格是否支持对应 size 格式播放

比如：Video 格式：H264 4K@30fps

详见《H6 多媒体支持列表》。

如果 H6 规格支持，又不能播放，可以尝试使用 # 5.2 使用 MediaPlayer 时，如何简单定位播放问题 # 提高的方法定位问题，反馈给 FAE。

7. Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner. The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.

confidential