密级状态：绝密(　　)　秘密(　　)　内部资料(　)　公开(　√　)

Rockchip BOX 视频开发指南

文件状态： ［　］草稿 ［√］正式发布 ［　］正在修改	文件标识：	Rockchip BOX 视频开发指南
	当前版本：	1.1
	作　　者：	许含瑞，洪涛
	完成日期：	2014-03-14
	审　　核：	黄激流
	审核日期：	2014.03.19

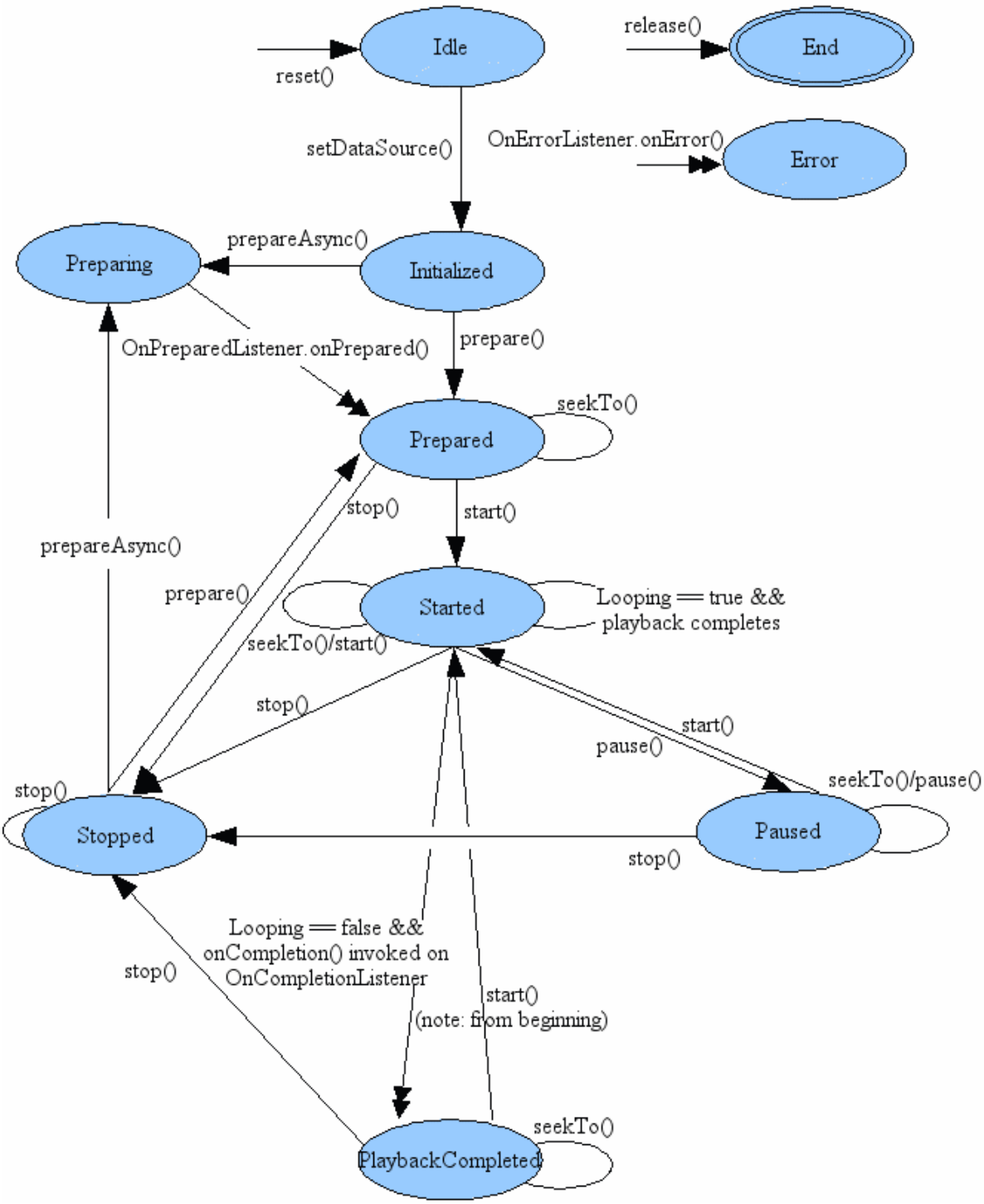
目 录

1 FFmpeg 介绍

FFmpeg 是一个开源免费跨平台的视频和音频流方案，属于自由软件，采用 LGPL 或 GPL 许可证（依据你选择的组件）。它提供了录制、转换以及流化音视频的完整解决方案。它包含了非常先进的音频/视频编解码库 libavcodec，为了保证高可移植性和编解码质量，libavcodec 里很多 codec 都是从头开发的。

2. 基于 ffmpeg 的 mediaplayer 的接口调用说明

2.1 状态图



2.2 接口说明：

2.2.1 获取网络速度的接口使用：

通过 MediaPlayer::getParameter(int key, Parcel *reply)来获取，需要传入的 key=1205.
此机制不采用主动上报，所以需要在 APK 需要的时候主动去获取，获取到数据类型是 int32 的。因为传出的是以 kb 作为单位的，所以当需要显示 kB 时，需要把获取到的网络速度除以 8。

2.2.2 Loding 进度的接口使用（这是卡住以后到重新播放的进度）

通过 setOnBufferingUpdateListener 这个监听函数来获取现在 Laoding 的进度。
这个 Android 官方标准是进度是相对整个片源文件来使用的，这里我们先把这个监听接口拿来使用。这个是主动上报的，只用在卡住的时候才会有上报的动作。

2.2.3 卡住和重新播放的接口使用

通过 SetInfoListener 这个监听函数来监听卡住暂停播放和重新播放的状态。MediaPlayer 需要配合 prepareAsync()这个函数使用，而不能使用 prepare()使用。
what = 701 暂停状态 （MEDIA_INFO_BUFFERING_START = 701）
What = 702 播放状态 (MEDIA_INFO_BUFFERING_END = 702)

2.2.4 设置缓冲区大小的接口使用

建议：不要为了减少了缓冲时间，而把缓冲区开的特别的小，对切换台要求高点的客户，可以把 KEY_PARAMETER_SET_TIME_MAX_UPDATA 这个 key 值传入 3。请不要出现 1252（key）传入的值比 1253（key）传入的值大的情况，1254（key）传入的值比 1255（key）传入的值大的情况。请传入的是 unint_t 型的值。
因为 ffmpeg 会根据不同的视频文件通过缓冲时间或缓冲数据来进行 buffering，所以建议这四个都进行设置
对应的单位为 100ms 或 1KB。

1. KEY_PARAMETER_SET_TIME_MIN_UPDATA=1252

这个 key 值是用来设置缓冲数据在少于多少秒时，进入缓冲状态。

例子：传入 100ms

```
Parcel data = Parcel.obtain();
data.writeInt(1);
mMediaPlayer.setParameter(1252, data);
```

2. KEY_PARAMETER_SET_TIME_MAX_UPDATA=1253

这个 key 值是用来设置缓冲数据在多于多少秒时，缓冲结束进入播放状态。

例子：传入 300ms

```
Parcel data = Parcel.obtain();
data.writeInt(3);
mMediaPlayer.setParameter(1253, data);
```

3. KEY_PARAMETER_SET_DATA_MIN_UPDATA=1254

这个 key 值是用来设置缓冲数据在少于多少 KB 时，进入缓冲状态。

例子：传入 2KB

```
Parcel data = Parcel.obtain();
data.writeInt(2);
mMediaPlayer.setParameter(1254, data);
```

4. KEY_PARAMETER_SET_DATA_MAX_UPDATA= 1255

这个 key 值是用来设置缓冲数据在少于多少 KB 时，缓冲结束进入播放状态。

例子：传入 8KB

```
Parcel data = Parcel.obtain();
data.writeInt(8);
```

mMediaPlayer.setParameter(1255, data);

PS:这些函数的设置都要放到 MediaPlayer.start()以后执行才能生效。

在框架层的修改如下：

/framworks/av/media/libmedia/mediaplayer.cpp 的 start 函数。

在 start 开始的时候添加如下：

```
Parcel requestMinTime;
requestMinTime.writeInt32(20);
setParameter(1252,requestMinTime);
Parcel requestMaxTime;
requestMaxTime.writeInt32(30);
setParameter(1253,requestMaxTime);
```

2.2.5 监听播放完成事件

MEDIA_INFO_BLURAY_COMPLETE = 950

APK 层通过监听 info，当 what = 950,说明播放完成了，可以进行其他的操作。

注：其他接口的调用方法都不变。

3. 基于 ffmpeg 的系统属性说明

3.1 sys.dts_ac3.shield

此属性用于设置是否屏蔽 dts, ac3 音效。

sys.dts_ac3.shield = 1 屏蔽 dts，ac3 音效。

sys.dts_ac3.shield=0 或不设置此属性，不屏蔽 dts，ac3 音效。

在 device/rockchip/rksdk/device.mk 添加上面的属性值。不同的平台对应的目录结构不同，但是都是在 device.mk 文件里修改。

注：如果 media.decoder.cfg 和 sys.dts_ac3.shield 属性同时配置的话，sys.dts_ac3.shield 属性将没有作用。

3.2 media.decoder.cfg

此属性用于设置支持解码器类型，如果设置了此属性，那在属性中没有出现的解码器类型的视频将不能播放。

先支持的屏蔽解码器类型，且书写格式如下：

AC3_DTS_ATARC_EAC3_AAC_MP3_FLAC_AMRNB_AMRWB_PCM_MP2_M1V_M2V_MPEG4_FLV_H264_RV30_RV40_VC1_WMV3_VP8_MJPEG_WMV1_WMV2_WMV3_H263

在 device/rockchip/rksdk/device.mk 添加上面的属性值。不同的平台对应的目录结构不同，但是都是在 device.mk 文件里修改。

AAC 默认支持：

AAC

AAC_LATM

先增加了对 PCM 的解码器的支持，只要配置了 PCM 这个选项，就默认支持一下 PCM 解码格式：

/* various PCM "codecs" */

AV_CODEC_ID_FIRST_AUDIO = 0x10000,

AV_CODEC_ID_PCM_S16LE = 0x10000,

AV_CODEC_ID_PCM_S16BE,

AV_CODEC_ID_PCM_U16LE,

AV_CODEC_ID_PCM_U16BE,

AV_CODEC_ID_PCM_S8,

AV_CODEC_ID_PCM_U8,

AV_CODEC_ID_PCM_MULAW,

AV_CODEC_ID_PCM_ALAW,

AV_CODEC_ID_PCM_S32LE,

AV_CODEC_ID_PCM_S32BE,

AV_CODEC_ID_PCM_U32LE,

AV_CODEC_ID_PCM_U32BE,

AV_CODEC_ID_PCM_S24LE,

AV_CODEC_ID_PCM_S24BE,

AV_CODEC_ID_PCM_U24LE,

AV_CODEC_ID_PCM_U24BE,

AV_CODEC_ID_PCM_S24DAUD,

AV_CODEC_ID_PCM_ZORK,

AV_CODEC_ID_PCM_S16LE_PLANAR,

AV_CODEC_ID_PCM_DVD,

AV_CODEC_ID_PCM_F32BE,

AV_CODEC_ID_PCM_F32LE,

AV_CODEC_ID_PCM_F64BE,

AV_CODEC_ID_PCM_F64LE,

AV_CODEC_ID_PCM_BLURAY,

AV_CODEC_ID_PCM_LXF,

AV_CODEC_ID_S302M,

AV_CODEC_ID_PCM_S8_PLANAR,

3.3 media.rmvb.support

此属性用于设置是否支持 rmvb 格式播放。

media.rmvb.support = 1 支持 rmvb 播放

Media.rmvb.support = 0 或不设置此属性，不支持 rmvb 播放

在 device/rockchip/rksdk/device.mk 添加上面的属性值。不同的平台对应的目录结构不同，但是都是在 device.mk 文件里修改。

注：如果 media.decoder.cfg 和 media.rmvb.support 属性同时配置的话，media.rmvb.support 属性将没有作用。

3.4 sys.sub_pgs.shield

此属性用于设置是否屏蔽内嵌 PGS 内嵌字幕显示

sys.sub_pgs.shield = 1 屏蔽内嵌 PGS 字幕

sys.sub_pgs.shield = 0 或不设置此属性， 显示内嵌 PGS 字幕

在 device/rockchip/rksdk/device.mk 添加上面的属性值。不同的平台对应的目录结构不同，但是都是在 device.mk 文件里修改。

3.5 sys.ffmpeg_sf.switch

此属性用于切换多媒体框架的。

sys.ffmpeg_sf.switch = 1 使用 stagefright 这套多媒体框架

sys.ffmpeg_sf.switch = 0 或不设置此属性，使用 ffmpeg 这套多媒体框架

在 device/rockchip/rksdk/device.mk 添加上面的属性值。不同的平台对应的目录结构不同，但是都是在 device.mk 文件里修改。

默认的是使用 ffmpeg 这套多媒体框架

3.6 media.decoder.unsupported

此属性用于设置不支持的音频，如果设置了此属性，那么属性中出现的格式将不能播放。

书写格式如下, “_”为分隔符。

AC3_DTS_ATARC_E-AC-3_TRUEHD

在 device/rockchip/rksdk/device.mk 添加上面的属性值。不同的平台对应的目录结构不同，但是都是在 device.mk 文件里修改。

3.7 media.audio.playback

此属性用于设置当有不支持的音频时的播放效果

media.audio.playback =1　当有不支持的音频时，视频也不播放。

media.audio.playback = 0 或者不设置，在有不支持的音频时，视频播放。

在 device/rockchip/rksdk/device.mk 添加上面的属性值。不同的平台对应的目录结构不同，但是都是在 device.mk 文件里修改。