

H3 项目

音频模块说明书 V1.0

文档履历

版本号	日期	制/修订人	内容描述
V1.0	2015-01-08		正式版本

confidential

目 录

1	前言	4
1.1	编写目的	4
1.2	适用范围	4
1.3	相关人员	4
1.4	相关术语	4
2	H3 音频系统框架概述	5
2.1	H3 原型机音频硬件框架图.....	5
2.2	H3 软件框架图	6
3	H3 音频模块介绍	7
3.1	Audiocodec 模块功能	7
3.2	Daudio1 模块功能.....	8
3.3	Daudio2(hdmiaudio)模块功能	8
3.4	Owa(Spdif)模块功能	8
4	H3 音频配置	9
4.1	源码结构	9
4.2	内核配置	9
4.2.1	Menuconfig 配置	9
4.3	系统配置 (sys_config.fex)	11
4.3.1	H3 内置模拟音频通道(codec)配置.....	11
4.3.2	数字音频总线 (S/PDIF)	11
4.3.3	外挂音频芯片接口(I2S/PCM).....	12
5	音频输入输出切换策略	14
5.1	FVD 音频输出策略	14
5.2	阿里云方案音频输出策略	15
5.3	音频输入策略	15
5.4	音频策略定制	16
6	模块开发 Demo	17
7	Declaration.....	18

1 前言

1.1 编写目的

本文档目的是让开发者了解 H3 音频系统框架，能够在 H3 平台上开发新的音频方案。

1.2 适用范围

本模块说明适用于 H3 平台。

1.3 相关人员

音频系统开发人员。

1.4 相关术语

Audio Driver: Acronyms	
Acronym	Definition
ALSA	Advanced Linux Sound Architecture
DMA	即直接内存存取，指数据不经 cpu，直接在设备和内存，内存和内存，设备和设备之间传输。
OSS	Open Sound System
样本长度 (sample)	样本是记录音频数据最基本的单位，常见的有 8 位和 16 位
通道数 (channel)	该参数为 1 表示单声道，2 则是立体声。
帧(frame)	帧记录了一个声音单元，其长度为样本长度与通道数的乘积。
采样率(rate)	每秒钟采样次数，该次数是针对帧而言。
周期(period)	音频设备一次处理所需要的帧数，对于音频设备的数据访问以及音频数据的存储，都是以此为单位。
交错模式 (interleave)	是一种音频数据的记录模式，在交错模式下，数据以连续帧的形式存放，即首先记录完帧 1 的左声道样本和右声道样本（假设为立体声格式），再开始帧 2 的记录，而在非交错模式下，首先记录的是一个周期内所有帧的左声道样本，再记录右声道样本，数据是以连续通道的方式存储。不过多数情况下，我们只需要使用交错模式就可以了。
Hdmiaudio	内置 hdmi 音频接口
SPDIF	外置音响音频设备接口，一般使用同轴电缆或光纤接口
I2S	外置音频通道接口
AGC	录音自动增益控制
DRC	音频输出动态范围控制
daudio	数字音频接口，可配置成 i2s/pcm 格式标准音频接口
aif	数字音频接口
xrun	音频流异常状态

2 H3 音频系统框架概述

2.1 H3 原型机音频硬件框架图

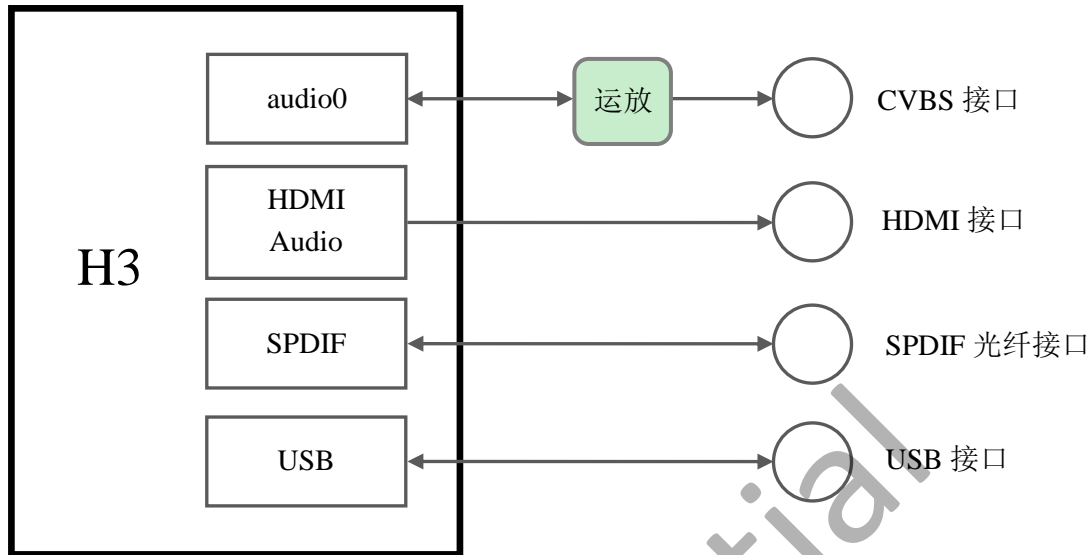


图 2-1 H3 原型机音频硬件框架图

各个设备对应的设备节点：

硬件接口/设备	设备节点	/sys/class/sound/cardX /id	备注
CVBS/耳机接口	/dev/snd/controlC0 /dev/snd/pcmC0D0c /dev/snd/pcmC0D0p	snddaudio	Controlx 为控制器 pcmCxDxc 为输入节点 pcmCxDxp 为输出节点 设备属性节点在： /sys/class/sound/cardX,
HDMI 接口	/dev/snd/controlC1 /dev/snd/pcmC1D0p	sndhdm	
SPDIF 接口	/dev/snd/controlC2 /dev/snd/pcmC2D0c /dev/snd/pcmC2D0p	sndspdif	
USB Audio	/dev/snd/controlC3 /dev/snd/pcmC3D0c /dev/snd/pcmC3D0p	从 usb 设备中读取获得	

可以输入一下命令查看系统挂载上的声卡：

```
cat /proc/asound/cards
0 [snddaudio    ]: snddaudio - snddaudio
                      snddaudio
1 [sndhdm       ]: sndhdm - sndhdm
                      sndhdm
2 [sndspdif     ]: sndspdif - sndspdif
                      sndspdif
3 [Device       ]: USB-Audio - USB Audio Device
                      Elite Silicon USB Audio Device at usb-sunxi-ohci-1, full speed
```

2.2 H3 软件框架图

H3 音频软件框架，大部分沿用原生系统框架。

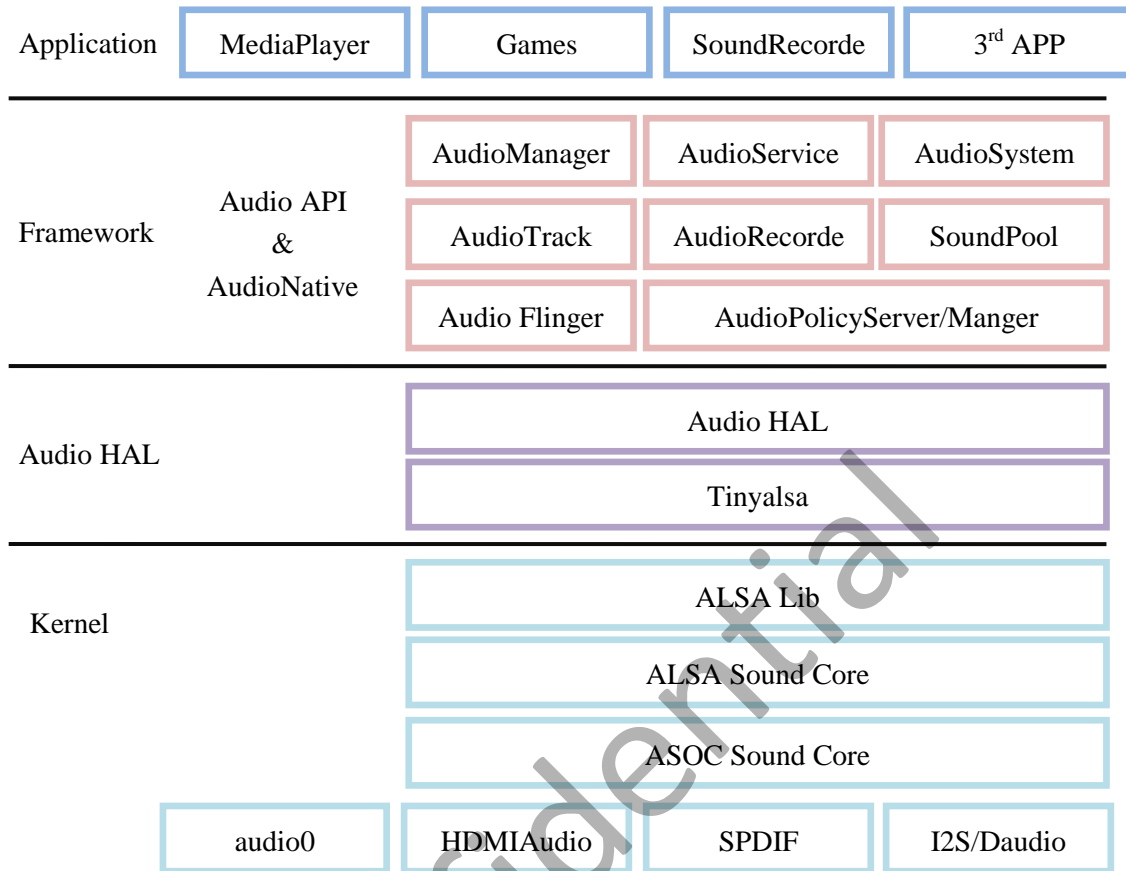


图 3-1 H3 系统软件结构图

与 Android 原生框架不同是，H3 盒子有一套独立的音频输入输出策略。同时具备原生系统所不具备的功能，例如支持 HDMI，USB，CVBS 等接口的热插拔，支持单/多路输出，支持音频透传等。

3 H3 音频模块介绍

在 H3 中,存在 5 个音频设备。 存在 5 个音频设备。分别为 audiocodec,daudio0,daudio1,daudio2, owa(spdif)。每一个音频设备都采用 asoc 架构实现。

asoc 是建立在标准 alsa 驱动层上,为了更好地支持嵌入式处理器和移动设备中的音频 codec 的一套软件体系, asoc 将音频系统分为 3 部分: Machine, Platform 和 Codec。

(1) Codec 驱动

ASoC 中的一个重要设计原则就是要求 Codec 驱动是平台无关的,它包含了一些音频的控件(Controls), 音频接口, DAPM (动态音频电源管理) 的定义和某些 Codec IO 功能。为了保证硬件无关性,任何特定于平台和机器的代码都要移到 Platform 和 Machine 驱动中。

所有的 Codec 驱动都要提供以下特性:

Codec DAI (Digital Audio Interface) 和 PCM 的配置信息;

Codec 的 IO 控制方式 (I2C, SPI 等);

Mixer 和其他的音频控件;

Codec 的 ALSA 音频操作接口;

必要时, 也可以提供以下功能:

DAPM 描述信息;

DAPM 事件处理程序;

DAC 数字静音控制

(2) Platform 驱动

它包含了该 SoC 平台的音频 DMA 和音频接口的配置和控制 (I2S, PCM, AC97 等等); 一般不包含与板子或 codec 相关的代码。

(3) Machine 驱动

单独的 Platform 和 Codec 驱动是不能工作的,它必须由 Machine 驱动把它们结合在一起才能完成整个设备的音频处理工作。

3.1 Audiocodec 模块功能

Audio Codec 驱动所具有的功能:

- 支持多种采样率格式 (11.025 KHz, 12 KHz, 16 KHz, 22.05 KHz, 24 KHz, 32 KHz, 44.1 KHz , 48 KHz, 96KHz, 192KHz);
- 支持 mono 和 stereo 模式;
- 支持同时 playback 和 record (全双工模式);
- 支持 mixer 接口
- 支持 dapm 接口
- 支持 16bit 24bit 数据精度

3.2 Daudio1 模块功能

驱动所具有的功能：

- 支持多种采样率格式(8khz, 11.025khz, 16khz, 22.05khz, 24khz, 32khz, 44.1khz, 48khz, 88.2khz, 96khz, 176.4khz, 192khz);
- 支持 mono 和 stereo 模式;
- 支持同时 playback 和 record(全双工模式);
- 支持 i2s、pcm 配置。
- 支持 16bit 24bit 数据精度

3.3 Daudio2(hdmiaudio)模块功能

Daudio2(hdmiaudio)驱动所具有的功能：

- 支持多种采样率格式(32khz, 44.1, 48khz, 96khz, 192khz);
- 支持 mono 和 stereo 模式;
- 只支持 playback 模式, 不支持 record 模式。
- 支持 16bit 24bit 数据精度
- 支持 raw 数据输出

Hdmiaudio 功能采用 daudio2 接口。

3.4 Owa(Spdif)模块功能

spdif 驱动所具有的功能：

- 支持多种采样率格式(22.05khz, 24khz, 32khz, 44.1khz, 48khz, 88.2khz, 96khz, 176.4khz, 192khz);
- 支持 mono 和 stereo 模式;
- 只支持 playback 模式, 不支持 record 模式。
- 支持 16bit 24bit 数据精度
- 支持 raw 数据输出

4 H3 音频配置

4.1 源码结构



4.2 内核配置

4.2.1 Menuconfig 配置

在编译服务器上，目录为\lichee\linux-3.4上，输入命令：

```
make ARCH=arm menuconfig
```

```
.config - Linux/arm 3.4.39 Kernel Configuration  
Linux/arm 3.4.39 Kernel Configuration  
[*] Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y>  
includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend:  
[*] built-in [ ] excluded <M> module <> module capable  
  
[?] Patch physical to virtual translations at runtime  
General setup --->  
[*] Enable loadable module support --->  
[*] Enable the block layer --->  
System Type --->  
[ ] FIQ Mode Serial Debugger  
Bus support --->  
Kernel Features --->  
Boot options --->  
CPU Power Management --->  
Floating point emulation --->  
Userspace binary formats --->  
Power management options --->  
[*] Networking support --->  
Device Drivers --->  
File systems --->  
Kernel hacking --->  
Security options --->  
*- Cryptographic API --->  
Library routines --->  
Modules --->  
---  
Load an Alternate Configuration File  
Save an Alternate Configuration File  
  
<Select> <Exit> <Help>
```

➤ 音频驱动配置:

Device Drivers --->

<*> Sound card support --->

<*> Advanced Linux Sound Architecture --->

```

#####
-- Advanced Linux Sound Architecture
< > Sequencer support
< > OSS Mixer API
< > OSS PCM (digital audio) API
< > HR-timer backend support
[ ] Dynamic device file minor numbers
[*] Support old ALSA API
[*] Verbose procfs contents
[ ] Verbose printk
[ ] Debug
[*] Generic sound devices --->
[*] ARM sound devices --->
[*] SPI sound devices --->
[*] USB sound devices --->
<*> ALSA for SoC audio support --->

```

<*> ALSA for SoC audio support --->

```

#####
-- ALSA for SoC audio support
<*> Audiocodec for the SUNXI chips
<*> Audiocodec Machine for codec chips
<*> Audiocodec for the SUN8IW7 chips
< > SoC daudio0 tdm interface for SUNXI chips
< > Public Machine for SUNXI chips
< > SoC daudio1 tdm interface for SUNXI chips
<*> HDMI Audio for SUNXI chips
<*> HDMIPCM for the SUN8IW7 chips
<M> sunxi On-Chip spdif
< > SoC audiohub interface for the allwinnertech SUNXI chips
< > AUDIOHUB for the SUN8IW7 chips
[*] Support Audio Raw Data

```

原型机上，CODEC 和 HDMI 都直接编入内核，SPDIF 和 USB Audio 的驱动着编成 ko。

➤ SPDIF 的驱动加载在:

android\device\softwinner\dolphin-fvd-pl\init.rc

```

on fs
#spdif
insmod /system/vendor/modules/sunxi_spdif.ko
insmod /system/vendor/modules/sunxi_spdma.ko
insmod /system/vendor/modules/sndspdif.ko
insmod /system/vendor/modules/sunxi_sndspdif.ko

```

➤ USB Audio 的驱动加载在:

android\device\softwinner\dolphin-common\init.sun8i.rc

```

on post-fs
#insmod usb audio card ko

```

```
insmod /system/vendor/modules/snd-hwdep.ko
insmod /system/vendor/modules/snd-usbmidi-lib.ko
insmod /system/vendor/modules/snd-usb-audio.ko
```

4.3 系统配置 (sys_config.fex)

配置文件的位置: lichee\tools\pack\chips\sun8iw7p1\configs\dolphin-xxx\sys_config.fex 目录下。音频中需要配置的有 audio0, spdif0 两个音频驱动。

4.3.1 H3 内置模拟音频通道(codec)配置

➤ [audio0]

配置项	配置项含义
audio_used	是否使用 H3 模拟音频输入输出 1: 使用, 0: 不使用
lineout_vol	Lineout 默认音量设置, 最大是 0x1f
cap_vol	Mic 的增益, 最大是 0x7
audio_hp_ldo	耳机口电压设置, H3 设置为 none, 不需改动
adcagc_used	录音音效增强, 1: 开启, 0: 不开启
adcdrc_used	录音动态音效调节, 1: 开启, 0: 不开启
dacdrc_used	播放动态音效调节, 1: 开启, 0: 不开启
adchpf_used	录音通路高通滤波开启, 1: 开启, 0: 不开启
dachpf_used	播放通路高通滤波开启, 1: 开启, 0: 不开启
audio_pa_ctrl	外部功放使能脚

配置举例:

```
audio_used      = 1
lineout_vol     = 0x1f
cap_vol        = 0x5
audio_hp_ldo    = none
adcagc_used     = 0
adcdrc_used     = 0
dacdrc_used     = 0
adchpf_used     = 0
dachpf_used     = 0
audio_pa_ctrl   = port:PA16<1><default><default><0>
```

4.3.2 数字音频总线 (S/PDIF)

Spdif 数字音频总线, 一般有同轴电缆和光纤接口两种, 使用光纤接口较多。Spdif 接口提供输入和输出功能。可根据需求配置。

➤ [spdif0]

配置项	配置项含义
spdif_used	是否开启 spdif, 1: 开启, 0: 不开启
spdif_dout	Spdif 输出管脚

配置举例：

```
[spdif0]
spdif_used          = 1
spdif_dout          = port:PA17<2><1><default><default>
```

4.3.3 外挂音频芯片接口(I2S/PCM)

➤ [pcm0]

配置项	配置项含义
daudio_used	是否使用该接口，默认要配置为 1 1: 使用 0: 不使用
daudio_master	该参数为 tdm 的主从设置，通常设为 4。 1: SND_SOC_DAIFMT_CBM_CFM(codec clk & FRM master) use 2: SND_SOC_DAIFMT_CBS_CFM(codec clk slave & FRM master) not use (CODEC as master, i2s as slave) 3: SND_SOC_DAIFMT_CBM_CFS(codec clk master & frame slave) not use 4: SND_SOC_DAIFMT_CBS_CFS(codec clk & FRM slave) use (CODEC as slave, i2s as master)
daudio_select	该参数为设置 tdm 支持 i2s 格式还是 pcm 格式，使用默认值 i2s 格式 0: pcm. 1: i2s
audio_format	该参数配置传输的数据格式，默认配置配置为 i2s 标准格式 1:SND_SOC_DAIFMT_I2S(standard i2s format). use 2:SND_SOC_DAIFMT_RIGHT_J(right justified format). 3:SND_SOC_DAIFMT_LEFT_J(left justified format) 4:SND_SOC_DAIFMT_DSP_A(pcm. MSB is available on 2nd BCLK rising edge after LRC rising edge). use 5:SND_SOC_DAIFMT_DSP_B(pcm. MSB is available on 1nd BCLK rising edge after LRC rising edge)
signal_inversion	bclk 时钟信号和帧信号要不要翻转，一般都为 normal 1:SND_SOC_DAIFMT_NB_NF(normal bit clock + frame) use 2:SND_SOC_DAIFMT_NB_IF(normal BCLK + inv FRM) 3:SND_SOC_DAIFMT_IB_NF(invert BCLK + nor FRM) use 4:SND_SOC_DAIFMT_IB_IF(invert BCLK + FRM)
mclk_fs	support 128fs/192fs/256fs/384fs/512fs/768fs（使用默认值）
sample_resolution	该参数指采样精度，通常默认 16 16bits/20bits/24bits
slot_width_select	Slot 宽度（使用默认值 16）
pcm_lrck_period	单声道 blk 个数/lrck 个数（使用默认值 32）
pcm_lrckr_period	Lrckr 参数：使用默认值
msb_lsb_first	0: msb first; 1: lsb first（使用默认值）
sign_extend	Pcm 格式参数：选用默认
slot_index	Pcm 格式参数：选用默认
frame_width	Pcm 格式参数：选用默认

tx_data_mode	Pcm 格式参数：选用默认
rx_data_mode	Pcm 格式参数：选用默认
i2s_mclk	i2s mclk GPIO 配置
i2s_bclk	i2s bclk GPIO 配置
i2s_lrclk	i2s lrclk GPIO 配置
i2s_dout0	i2s dout0 GPIO 配置
i2s_dout1	i2s dout1 GPIO 配置
i2s_dout2	i2s dout2 GPIO 配置
i2s_dout3	i2s dout3 GPIO 配置
i2s_din	i2s din GPIO 配置

配置举例：

```

daudio_used          = 0
daudio_master        = 4
daudio_select        = 1
audio_format         = 1
signal_inversion     = 1
mclk_fs              = 128
sample_resolution    = 16
slot_width_select    = 32
;pcm_sync_period     = 256
pcm_lrck_period      = 32
pcm_lrckr_period     = 1
msb_lsb_first        = 0
sign_extend          = 0
slot_index           = 0
slot_width           = 32
frame_width          = 0
tx_data_mode         = 0
rx_data_mode         = 0
i2s_mclk              = port:PA18<2><1><default><default>
i2s_bclk              = port:PA19<2><1><default><default>
i2s_dout0             = port:PA20<2><1><default><default>
i2s_din              = port:PA21<2><1><default><default>

```

5 音频输入输出切换策略

5.1 FVD 音频输出策略

音频输出策略如表所述。

表 1 TV-BOX 音频输出切换策略

当前状态	音频输出	具体操作	操作后显示状态	音频输出
只插着 HDMI	HDMI 输出 或用户手动更改	拔出 HDMI 线	输出 NONE	切换到 CODEC
		插入 CVBS 线	不做任何处理	不做任何处理
		插入 USB 音频输出设备	不做任何处理	切换到仅 USB 输出
		待机, 唤醒, 关机	输出 NONE	不做任何处理
		开机	HDMI 输出	添加 HDMI 输出
只插着 CVBS	CODEC 输出 或用户手动更改	拔出 CVBS 线	输出 NONE	不做任何处理
		插入 HDMI 线	切换到 HDMI	添加 HDMI 输出
		插入 USB 音频输出设备	不做任何处理	切换到仅 USB 输出
		待机, 唤醒, 关机	输出 NONE	不做任何处理
		开机	CVBS 输出	添加 CODEC 输出
同时插着 HDMI 和 CVBS	默认跟随当前显示输出或用户手动更改	拔出当前显示类型的设备线	切换到另一显示类型	跟随显示添加输出
		拔出非当前显示类型的设备线	不做任何处理	不做任何处理
		插入 USB 音频输出设备	不做任何处理	切换到仅 USB 输出
		手动切换到设备支持的显示类型, 包括 setting 和遥控 (phoneWindowManage)	成功切换后, 保存该显示类型到数据库	跟随显示添加输出
		待机, 唤醒, 关机	输出 NONE	不做任何处理
		开机	HDMI 输出	添加 HDMI 输出
无插着任何设备	默认 CODEC 输出	插入 HDMI 线	HDMI 输出	添加 HDMI 输出
		插入 CVBS 线	CVBS 输出	添加 CODEC 输出
		插入 USB 音频输出设备	不做任何处理	切换到仅 USB 输出
		待机, 唤醒, 关机	输出 NONE	不做任何处理
		开机	输出 NONE	不做任何处理
USB 音频输出	仅 USB 音频输出	插拔 HDMI, 插拔 CVBS, 待机, 唤醒, 关机, 开机	相应变化	不做任何处理
		拔出 USB 音频设备	不做任何处理	切换到当前显示对应音频输出
		手动选择其他音频	不做任何处理	切换到对应输出
开启透传模式	SPDIF 透传	插拔 HDMI, 插拔 CVBS, 待机, 唤醒, 关机, 开机	相应变化	不做任何处理
	HDMI 透传	插拔 CVBS, 待机, 唤醒, 关机, 开机	不做任何处理	不做任何处理
		插入 USB 音频输出设备	不做任何处理	切换到仅 USB 输出

		拔出 USB 音频输出设备	不做任何处理	切换到当前显示对应音频输出
--	--	---------------	--------	---------------

说明：

1. SPDIF 仅支持手动设置输出。无法自动检测切换。除了插入 USB 音频输出设备外，其他情况的插拔不会影响 SPDIF 的输出状态。
2. 插入 USB 音频设备后不能复选多路。
3. 显示自动切换到 CVBS 时，CODEC 会自动开启。但手动拔出 CVBS 后 CODEC 会保留。只有在插入 USB 音频或手动清除选项后才会不复选上 CODEC。
4. 勾选了透传模式，只能选择单路输出。
5. 只有 HDMI 和 SPDIF 才能实现透传。透传模式下，选择 CODEC 和 USB 音频输出，不会有透传的效果，但不会影响正常的音频输出。
6. 透传模式下无法使用遥控器控制音量和静音。

5.2 阿里云方案音频输出策略

阿里云方案音频输出策略如下：

1. 系统默认从 CODEC，HDMI，SPDIF 三路输出，开关机、休眠唤醒，皆如此。
2. 插入 USB 音频输出设备后，切换到 USB 设备输出。拔出后还原默认三路输出。开关机、休眠唤醒，皆如此。

注意：

1. 阿里云方案由于设置限制，无法使用音频透传功能。
2. 阿里云方案中音频只有一个全局的开关设置。默认为 3 路输出。开关也是 3 路同时。

5.3 音频输入策略

音频输入策略如表 2 所述。

表 2 音频输入策略

当前设备	音频输入	具体操作	音频输入
CODEC	内置 MIC 或 LINE IN	插入 USB 音频输入设备	切换到 USB 音频输入
USB 音频设备	USB 音频输入	拔出 USB 音频设备	切换到 CODEC 音频输入

说明：

1. 系统默认的音频输入设备是 CODEC。
2. 内置 MIC 或者 LINE IN 使用的均为 CODEC 声卡。
3. USB 音频输入设备一般有：USB MIC、带 MIC 的 USB Camera、带 MIC 的 USB 空鼠。

5.4 音频策略定制

- Audio 设备插拔事件广播管理：

```
android/frameworks/base/services/java/com/android/server/AudioDeviceManagerObserver.java
```

- Audio 切换策略：

```
android/frameworks/base/services/java/com/android/server/AudioManagerPolicy.java
```

- Audio 服务启动：

```
android/frameworks/base/services/java/com/android/server/SystemServer.java
```

- Audio 输入输出接口扩展：

```
android/frameworks/base/media/java/android/media/AudioManager.java
```

- Audio 系统设置：

```
android/vendor/tvd/packages/TvdSettings/src/com/android/settings/AudioChannelsSelect.java
```

- 首次开机音频输出通路配置和默认音频策略配置：

```
android/device/softwinner/eagle-fvd-pl/overlay/frameworks/base/packages/SettingsProvider/res/values/defaults.xml
```

H8 fvd 方案默认首次开机 HDMI 和 CODEC 两路输出。

```
<string name="def_audio_output_channel" translatable="false">AUDIO_HDMI, AUDIO_CODEC</string>
```

多路使用逗号隔开，可以填写 AUDIO_HDMI, AUDIO_CODEC, AUDIO_SPDIF 三个通路的任意组合。

可以设置 def_audio_manage_policy 的来配置默认策略。A80 默认为 0，AliyunOS 默认为 1，H8 FVD 默认为 2，建议对应方案适用该对应的默认配置，原厂的方案已预先配好，使用原厂配置即可。

```
<!-- tvd set default 0, AliyunOS set default 1, fvd set default 2-->
<integer name="def_audio_manage_policy">2</integer>
```


6 模块开发 Demo

音频的外部接口跟普通的驱动不同。音频的 application 需要额外的 alsa-lib 进行外部接口的封装。在 android2.3.4 中，用的是 small alsa 应用库，而在 android4.0 以后，采用 tinyalsa 应用库进行外部接口的封装。所有的接口都是音频的标准接口。在这里不一一罗列。下面给出播放和录音的应用。

写一个音频应用程序涉及到以下几步

- opening the audio device
- set the parameters of the device
- receive audio data from the device or deliver audio data to the device
- close the device

在 H3 中，涉及 4 个音频驱动，alsa 的 lib 库支持任何一个音频驱动。请参考最小 playback 应用和录音应用以及 mixer 接口的使用方法。

demo 采用 tiny_alsa 库，可以从\android4.4\external\tinyalsa 中获得。

Confidential

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