

密级状态： 绝密() 秘密() 内部资料() 公开(√)
Security Class: Top-Secret () Secret () Internal () Public (√)

Android R 异声配置说明

Android_R_DualAudio_Configuration_Introduction

(第二系统产品部)

(Technical Department, R & D Dept. II)

文件状态: Status: [] 草稿 [] Draft [] 正在修改 [] Modifying [√] 正式发布 [√] Released	文件标识: File No.:	
	当前版本: Current Version:	
	作 者: Author:	
	完成日期: Finish Date:	2021-5-18
	审 核: Auditor:	
	审核日期: Finish Date:	

免责声明

本文档按“现状”提供，福州瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

Disclaimer

This document is provided “as is” and Fuzhou Rockchip Electronics Co. Ltd (“the company”) makes no express or implied statement or warranty as to the accuracy, reliability, completeness, merchantability, specific purpose and non-infringement of any statement, information and contents of the document. This document is for reference only.

This document may be updated without any notification due to product version upgrades or other reasons.

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

Brand Statement

Rockchip, Rockchip™ icon, Rockchip and other Rockchip trademarks are trademarks of Fuzhou Rockchip Electronics Co., Ltd., and are owned by Fuzhou Rockchip Electronics Co., Ltd.

All other trademarks or registered trademarks mentioned in this document are owned by their respective owners.

版权所有 © 2021 福州瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

Copyright © 2021 Fuzhou Rockchip Electronics Co., Ltd.

Beyond reasonable use, without the written permission, any unit or individual shall not extract or copy part or all of the content of this document, and shall not spread in any form.

福州瑞芯微电子股份有限公司

Fuzhou Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园 A 区 18 号

网址：www.rock-chips.com

客户服务电话：+86-4007-700-590

客户服务传真：+86-591-83951833

客户服务邮箱：fae@rock-chips.com

Fuzhou Rockchip Electronics Co., Ltd.

Address: No. 18 Building, A District, No.89,software Boulevard Fuzhou,Fujian,PRC

Website: www.rock-chips.com

Customer service tel.: +86-4007-700-590

Customer service fax: +86-591-83951833

Customer service e-mail: fae@rock-chips.com

版本历史 Revision History

版本号 Version no.	作者 Author	修改日期 Revision Date	修改说明 Revision description	备注 Remark
V1.0	刘兴亮	2021.5.18	初始版本	
V1.1				
V1.2				
V1.3				
V1.4				
V1.5				

目 录 Contents

ANDROID R 异声配置说明.....	1
1 介绍.....	1
2 前提条件.....	1
2.1 查看声卡数目.....	1
2.2 验证声卡路由是否正常.....	1
3 补丁配置.....	2
3.1 补丁内容.....	2
3.2 配置补丁.....	2
3.2.1 上层补丁.....	2
3.2.2 HAL 层补丁.....	3
4 应用端调用方法.....	3
5 补丁包选择及相应配置.....	3
5.1 SPEAKER + SPEAKER.....	3
5.1.1 场景描述.....	3
5.1.2 补丁包选择.....	4
5.1.3 扩展需求.....	4
5.2 SPEAKER + HDMI.....	4
5.2.1 场景描述.....	4
5.2.2 补丁包选择.....	4
5.2.3 扩展需求.....	4
5.3 SPEAKER + SPDIF.....	5
5.3.1 场景描述.....	5
5.3.2 补丁包选择.....	5
5.3.3 扩展需求.....	5
5.4 HDMI + SPDIF.....	5
5.4.1 场景描述.....	5
5.4.2 补丁包选择.....	6
5.4.3 扩展需求.....	6
5.5 SPEAKER + USB.....	6
5.5.1 场景描述.....	6
5.5.2 补丁包选择.....	6
5.5.3 扩展需求.....	6

1 介绍

异声即上层不同的音频解码数据通过不同的输出设备进行播放，从而达到声音分离的目的。

2 前提条件

设备需要至少两路声卡，且两路声卡互不影响。

如果只有一路声卡或者有两路声卡但是两路声卡的声音实质是来自同一个 i2s，无法实现声音分离，因为硬件不支持。

2.1 查看声卡数目

cat proc/asound/cards

```
rk3399_Android11:/ # cat proc/asound/cards
0 [hdmisound      ]: hdmi-sound - hdmi-sound
                        hdmi-sound
1 [realtekrt5651co]: realtek_rt5651- - realtek,rt5651-codec
                        realtek,rt5651-codec
2 [ROCKCHIPSPDIF  ]: ROCKCHIP_SPDIF - ROCKCHIP,SPDIF
                        ROCKCHIP,SPDIF
```

前面的 0 1 2 就是声卡对应的 id。

2.2 验证声卡通路是否正常

假如想通过声卡 0 和声卡 1 来实现异声，xxx.wav（用户自己找一个 wav 文件放到 sdcard 下），执行如下命令：

```
tinyplay sdcard/xxx.wav -D 0
```

```
tinyplay sdcard/xxx.wav -D 1
```

如果分别输出正常（设备有声音从对应声卡出来），说明声卡 0 和声卡 1 通路正常。

3 补丁配置

3.1 补丁内容

```
→ speaker_and_speaker tree
-
| 0001-dualaudio-add-define-for-secondary.patch
| 0001-dualaudio-move-spdif-to-secondary-audio-hal.patch
| 0001-DualAudio-some-change-in-framework-av.patch
| 0001-DualAudio-some-change-in-hardware-interface-audio.patch
| add_path.patch
| both_open.patch
| diff_in_primary.patch
| diff_in_secondary.patch
| dualaudio.tar.gz
| README.txt
| RKDualAudio.h
-
```

3.2 配置补丁

3.2.1 上层补丁

```
cd device/rockchip/commom
```

```
git am 0001-dualaudio-move-spdif-to-secondary-audio-hal.patch
```

```
cd hardware/interfaces
```

```
git am 0001-DualAudio-some-change-in-hardware-interface-audio.patch
```

```
cd system/media
```

```
git am 0001-dualaudio-add-define-for-secondary.patch
```

```
cd frameworks/av
```

```
git am 0001-DualAudio-some-change-in-framework-av.patch
```

将 RKDualAudio.h 复制到 frameworks/av/include/media

将 dualaudio.tar.gz 复制到 vendor/rockchip/common 并解压

3.2.2 HAL 层补丁

```
cd hardware/rockchip/audio
git apply add_path.patch
mkdir tinyalsa_hal_sercondary
复制 tinyalsa_hal 的代码至 tinyalsa_hal_sercondary
cd tinyalsa_hal_sercondary
git apply diff_in_secondary.patch
cd tinyalsa_hal
git apply diff_in_primary.patch
```

4 应用端调用方法

apk 内调用

```
MediaPlayer mp = new MediaPlayer();
mp.setAudioSessionId(audioSessionId);
mp.setDataSource(...);
```

在 setDataSource 之前调用 setAudioSessionId

```
audioSessionId = 57    从 secondary 声卡输出
audioSessionId = 121   从 primary 声卡输出
```

注意：57 和 121 是固定的值，代码里面已写死，不要用别的值，否则无法生效

5 补丁包选择及相应配置

不同的设备包含声卡类型可能不同，不同应用场景需求也不同，因此无法给出通用补丁。目前根据已知的场景整理了几个补丁包，用户根据需求选择对应补丁包。

5.1 speaker + speaker

5.1.1 场景描述

```
cat /proc/asound/cards
0 [rockchiprk809co]: rockchip_rk809- - rockchip, rk809-codec
rockchip, rk809-codec
1 [rockchipes8316c]: rockchip_es8316 - rockchip, es8316-codec
rockchip, es8316-codec
```

有些设备含有两个类型相同的声卡，这两个声卡对 Android 系统来说都是 SPEAKER 类型。因此，需要将其中一个声卡当作 SPDIF，对应 HAL 代码要做相应修改。

5.1.2 补丁包选择

选择 speaker_and_speaker.tar.gz 并解压

audioSessionId = 57 从 spdif 声卡输出

audioSessionId = 121 从 speaker 声卡输出

5.1.3 扩展需求

两路声卡同时输出一样的声音，需要加上如下补丁：

```
diff --git a/services/audiopolicy/managerdefault/AudioPolicyManager.cpp b/services/audiopolicy/managerdefault/AudioPolicyManager.cpp
index 205c28209f..487a861589 100644
--- a/services/audiopolicy/managerdefault/AudioPolicyManager.cpp
+++ b/services/audiopolicy/managerdefault/AudioPolicyManager.cpp
@@ -1001,7 +1001,13 @@ status_t AudioPolicyManager::getOutputForAttrInt(
     bool boo = false;
     audio_devices_t device = AUDIO_DEVICE_OUT_SPEAKER;
     dualaudio_D(sessionid, &this_flags, &flag1, &boo, &device);
     outputDevices = mAvailableOutputDevices.getDevicesFromType(device);
+    ALOGD("device:%x", device);
+    if(device == (AUDIO_DEVICE_OUT_SPEAKER | AUDIO_DEVICE_OUT_SPDIF)){
+        outputDevices = mAvailableOutputDevices.getDevicesFromType(AUDIO_DEVICE_OUT_SPEAKER);
+        outputDevices.merge(mAvailableOutputDevices.getDevicesFromType(AUDIO_DEVICE_OUT_SPDIF));
+    }else{
+        outputDevices = mAvailableOutputDevices.getDevicesFromType(device);
+    }

     if ((resultAttr->flags & AUDIO_FLAG_HW_AV_SYNC) != 0) {
         *flags = (audio_output_flags_t)(*flags | AUDIO_OUTPUT_FLAG_HW_AV_SYNC);
     }
 }
```

cd frameworks/av

git apply both_open.patch

5.2 speaker + hdmi

5.2.1 场景描述

设备包含 speaker 和 hdmi 声卡，期望不同声音从 speaker 和 hdmi 分别输出。

5.2.2 补丁包选择

选择 speaker_and_hdmi.tar.gz 并解压

audioSessionId = 57 从 hdmi 声卡输出

audioSessionId = 121 从 speaker 声卡输出

5.2.3 扩展需求

两路声卡同时输出一样的声音，需要加上如下补丁：

```

1 diff --git a/services/audiopolicy/managerdefault/AudioPolicyManager.cpp b/services/audiopolicy/managerdefault/AudioPolicyManager.cpp
2 index 205c28209f..487a861589 100644
3 --- a/services/audiopolicy/managerdefault/AudioPolicyManager.cpp
4 +++ b/services/audiopolicy/managerdefault/AudioPolicyManager.cpp
5 @@ -1001,7 +1001,13 @@ status_t AudioPolicyManager::getOutputForAttrInt(
6     bool boo = false;
7     audio_devices_t device = AUDIO_DEVICE_OUT_SPEAKER;
8     dualaudio_D(sessionid, &this_flags, &flag1, &boo, &device);
9     outputDevices = mAvailableOutputDevices.getDevicesFromType(device);
10 +    ALOGD("device:%x", device);
11 +    if(device == (AUDIO_DEVICE_OUT_SPEAKER | AUDIO_DEVICE_OUT_AUX_DIGITAL)){
12 +        outputDevices = mAvailableOutputDevices.getDevicesFromType(AUDIO_DEVICE_OUT_SPEAKER);
13 +        outputDevices.merge(mAvailableOutputDevices.getDevicesFromType(AUDIO_DEVICE_OUT_AUX_DIGITAL));
14 +    }else{
15 +        outputDevices = mAvailableOutputDevices.getDevicesFromType(device);
16 +    }
17
18     if ((resultAttr->flags & AUDIO_FLAG_HW_AV_SYNC) != 0) {
19         *flags = (audio_output_flags_t)(*flags | AUDIO_OUTPUT_FLAG_HW_AV_SYNC);

```

cd frameworks/av

git apply both_open.patch

5.3 speaker + spdif

5.3.1 场景描述

设备包含 speaker 和 spdif 声卡，期望不同声音从 speaker 和 spdif 分别输出。

5.3.2 补丁包选择

选择 speaker_and_spdif.tar.gz 并解压。

audioSessionId = 57 从 spdif 声卡输出

audioSessionId = 121 从 speaker 声卡输出

5.3.3 扩展需求

两路声卡同时输出一样的声音，需要加上如下补丁：

```

diff --git a/services/audiopolicy/managerdefault/AudioPolicyManager.cpp b/services/audiopolicy/managerdefault/AudioPolicyManager.cpp
index 205c28209f..487a861589 100644
--- a/services/audiopolicy/managerdefault/AudioPolicyManager.cpp
+++ b/services/audiopolicy/managerdefault/AudioPolicyManager.cpp
@@ -1001,7 +1001,13 @@ status_t AudioPolicyManager::getOutputForAttrInt(
    bool boo = false;
    audio_devices_t device = AUDIO_DEVICE_OUT_SPEAKER;
    dualaudio_D(sessionid, &this_flags, &flag1, &boo, &device);
-    outputDevices = mAvailableOutputDevices.getDevicesFromType(device);
+    ALOGD("device:%x", device);
+    if(device == (AUDIO_DEVICE_OUT_SPEAKER | AUDIO_DEVICE_OUT_SPDIF)){
+        outputDevices = mAvailableOutputDevices.getDevicesFromType(AUDIO_DEVICE_OUT_SPEAKER);
+        outputDevices.merge(mAvailableOutputDevices.getDevicesFromType(AUDIO_DEVICE_OUT_SPDIF));
+    }else{
+        outputDevices = mAvailableOutputDevices.getDevicesFromType(device);
+    }
+
    if ((resultAttr->flags & AUDIO_FLAG_HW_AV_SYNC) != 0) {
        *flags = (audio_output_flags_t)(*flags | AUDIO_OUTPUT_FLAG_HW_AV_SYNC);

```

cd frameworks/av

git apply both_open.patch

5.4 hdmi + spdif

5.4.1 场景描述

设备包含 hdmi 和 spdif 声卡，期望不同声音从 hdmi 和 spdif 分别输出。

5.4.2 补丁包选择

选择 `hdmi_and_spdif.tar.gz` 并解压。

`audioSessionId = 57` 从 `spdif` 声卡输出

`audioSessionId = 121` 从 `hdmi` 声卡输出

5.4.3 扩展需求

```
cd frameworks/av
```

```
git apply both_open.patch
```

5.5 speaker + usb

5.5.1 场景描述

设备本地只有一个声卡 `speaker`，并外接 `usb` 声卡，期望不同声音从本地声卡和 `usb` 分别输出。

5.5.2 补丁包选择

选择 `speaker_and_usb.tar.gz` 并解压。

`audioSessionId = 57` 从 `usb` 输出

`audioSessionId = 121` 从 `speaker` 输出

5.5.3 扩展需求

```
cd frameworks/av
```

```
git apply both_open.patch
```

注意： 如果还有其它应用场景，请[登记 redmine](#)。