



全志科技
Allwinner Technology

A83T

电话音频模块使用说明

Confidential

文档履历

版本号	日期	制/修订人	制/修订记录
V1.0	2014-10-15		初始版本

Confidential

目录

A83T	1
电话音频模块使用说明.....	1
文档履历	2
1. 概述.....	4
1.1. 编写目的.....	4
1.2. 适用范围.....	4
1.3. 相关人员.....	4
2. 模块介绍.....	5
2.1. 模块功能介绍.....	5
2.2. 源码结构介绍.....	5
2.3. 模块配置介绍.....	7
3. 模块体系结构描述.....	8
4. 模块数据结构描述.....	9
5. 模块接口描述.....	11
6. 客户使用详细说明.....	13
7. 客户使用案例 1.....	16
8. 客户使用案例 2.....	18
9. 客户使用案例 3.....	20
10. 模块调试.....	22

1. 概述

1.1. 编写目的

本文档目的是为了让开发者了解 A83T 音频系统框架,能够在 A83T 平台上开发新的音频方案。

1.2. 适用范围

硬件平台:

A83T 平台。

软件平台:

exdroid4.1.1_r1-a31s-v1.33 及以上版本。

1.3. 相关人员

音频系统开发人员。

2. 模块介绍

介绍本模块的模块功能，基本配置，目标代码的文件目录组织形式以及相关的硬件介绍。

2.1. 模块功能介绍

本模块基于 android 标准模块接口 audio_hw 作扩展，以适应不同 BP 硬件方案，不同硬件设计。

共三个模块，audio_hw.so libcodec_audio.so, libril_audio.so

1. 模块 libril_audio.so 可加入新的 BP 模块配置，无限扩展 BP 模块
2. 模块 libril_audio.so 可远程初始化 BP, 可设置 BP 的音量，发送 AT 指令
3. 模块 libcodec_audio.so 对 codec 端语音，可加入新的方案实现，适应不同的硬件设计。
4. 模块 libcodec_audio.so 具备芯片级接口，换芯片，更换内核接口后，只需修改少量接口代码。

2.2. 源码结构介绍

介绍本模块源码的基本目录组织形式。

```

user7@ExdroidM:~/a31/android/device/softwinner/fiber-common/hardware/audio$ tree
.
├── Android.mk
├── audio_hw.c
├── audio_iface.c
├── audio_iface.h
├── audio_policy.conf
├── libcodec_audio
│   ├── Android.mk
│   ├── asoundlib.h
│   ├── codec.c
│   ├── codec_devices
│   │   ├── pad
│   │   │   ├── manage.c
│   │   │   ├── pad.c
│   │   │   └── pad.h
│   │   ├── planOne
│   │   │   ├── manage.c
│   │   │   ├── plan_one.c
│   │   │   └── plan_one.h
│   │   └── planTwo
│   │       ├── manage.c
│   │       ├── manage-pcm_stop.c
│   │       ├── plan_two.c
│   │       └── plan_two.h
│   ├── codec_devices.c
│   ├── codec_devices.h
│   ├── codec_utils.c
│   ├── codec_utils.h
│   ├── hal_codec.h
│   ├── mixer.c
│   ├── pcm.c
│   ├── record.c
│   ├── record.h
│   ├── volume_conf.c
│   ├── wav.c
│   └── wav.h
├── libril_audio
│   ├── Android.mk
│   ├── bp.c
│   ├── bp_devices
│   │   ├── demo
│   │   │   ├── demo.c
│   │   │   └── demo.h
│   │   ├── mu509
│   │   │   ├── mu509.c
│   │   │   └── mu509.h
│   │   └── oviphone_em55
│   │       ├── em55.c
│   │       └── em55.h
│   ├── bp_devices.c
│   ├── bp_devices.h
│   ├── bp_utils.c
│   ├── bp_utils.h
│   └── hal_bp.h
└── 10 directories, 43 files

```

1. audio.primary.exdroid.so

包含 audio_hw.c audio_iface.c audio_iface.h

2. libcodec_audio.so

包含 codec.c codec_devices codec_devices.c codec_devices.h
codec_utils.c 等
codec_utils.h hal_codec.h

3 libril_audio.so

包含 bp.c bp_devices bp_devices.c bp_devices.h bp_utils.c bp_utils.h
hal_bp.h 等

2.3. 模块配置介绍

1. 音频连接选择 `ro.sw.audio.codec_plan_name`

目前支持三种硬件连接

A. pad 模式。

不做任何配置默认是 pad 模式。支持正常音乐与录音

B. BP 通过 PHONEP/PHONEN 模拟接入

FM 通过 LINEINL/LINEINR 模拟接入

蓝牙通过 PCM 数字接到 AP

软件配置为 `ro.sw.audio.codec_plan_name=PLAN_ONE`.

C. BP 通过 I2S 或 PCM 数字接入

FM 通过 LINEINL/LINEINR 模拟接入

蓝牙通过 PCM 数字接到 AP

软件配置为 `ro.sw.audio.codec_plan_name=PLAN_TWO`

2. BP 模组选择，已经有两种模组的代码结构

MU509 和 EM55.

软件配置为

`ro.sw.audio.bp_device_name=mu509` 或

`ro.sw.audio.bp_device_name=em55`

3. 是否使用听筒 `audio.without.earpiece=1`

配置为 1 表示不使用听筒。

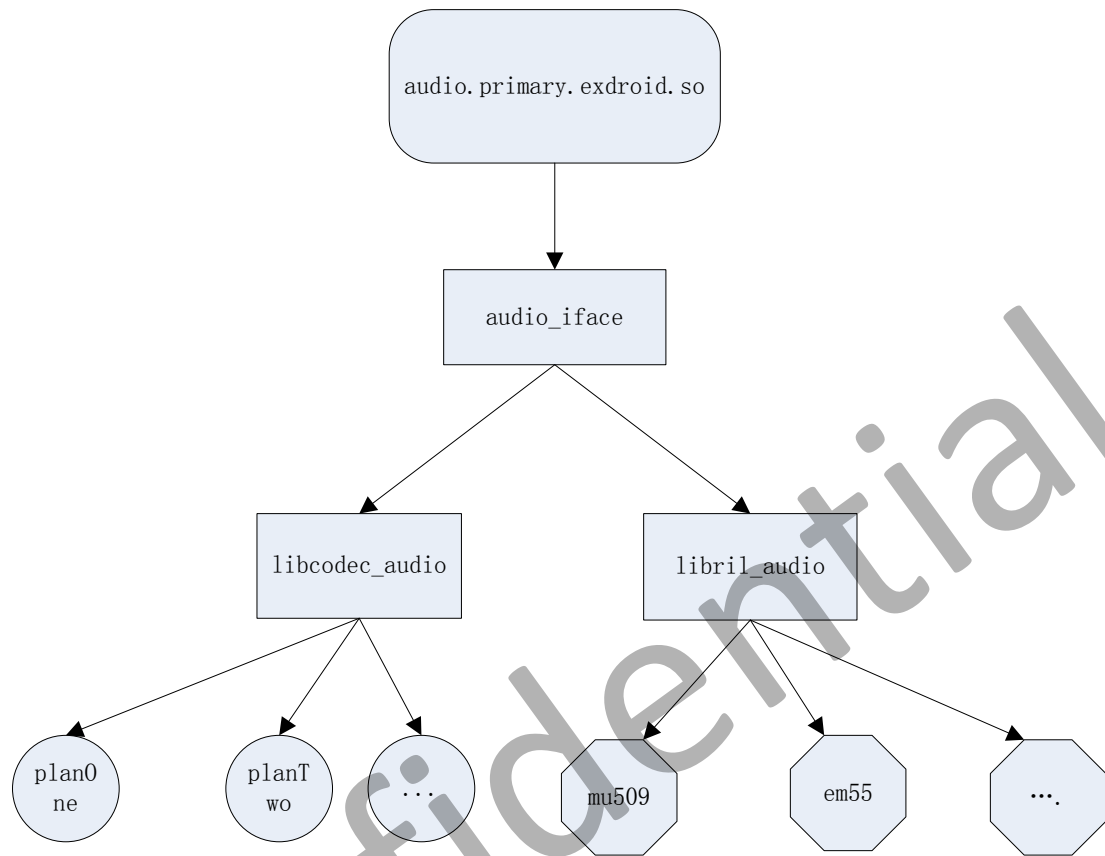
4. 语音下行音量，配置文件 `phone_volume.conf`.

5. 语音上行音量，配置文件 `sys_config.fex.`，节点时 `[audio_para]`

`headset_mic_vol` = 0x6 耳机 mic 增益

`main_mic_vol` = 0x0 主 mic 增益

3. 模块体系结构描述



4. 模块数据结构描述

基本数据结构描述。

1. 抽象音频设备

```
struct codec_client {
    struct mixer_ctls *mixer_ctls;
    struct normal_ops *normal_ops;
    struct fm_ops *fm_ops;
    struct factory_ops *factory_ops;
    struct phone_ops *phone_ops;
};
```

抽象音频方案选择

```
struct codec_device_desc codec_devices[] = {
    {
        .name = "bp_fm_Analog_bt_Pcm",
        .plan_name = "PLAN_ONE",
        .normal_ops = &plan_one_normal_ops,
        .fm_ops = &plan_one_fm_ops,
        .factory_ops = &plan_one_factory_ops,
        .phone_ops = &plan_one_phone_ops,
        .other_ops = NULL,
        .device_init = plan_one_init,
        .device_exit = plan_one_exit,
    },
    {
        .name = "bp_pcm_fm_Analog_bt_Pcm",
        .plan_name = "PLAN_TWO",
        .normal_ops = &plan_two_normal_ops,
        .fm_ops = &plan_two_fm_ops,
        .factory_ops = &plan_two_factory_ops,
        .phone_ops = &plan_two_phone_ops,
        .other_ops = NULL,
    }
};
```

```
.device_init = plan_two_init,  
.device_exit = plan_two_exit,  
  
},  
};
```

2. 抽象 BP 模块

```
struct bp_client {  
    struct bp_ops *bp_ops;  
};
```

抽象 BP 模组案

```
struct bp_device_desc bp_devices[] = {  
    {  
        .name = "mu509",  
        .board_name = "xxx1_plan",  
        .ops = &mu509_ops,  
        .other_ops = NULL,  
    },  
    {  
        .name = "em55",  
        .board_name = "xxx2_plan",  
        .ops = &em55_ops,  
        .other_ops = NULL,  
    }  
};
```

5. 模块接口描述

1. Audio_iface 接口函数。

声音路径切换接口

```
void normal_play_route(int path);
void phone_play_route(int path);
```

例如：

- **PROTOTYPE**
- void normal_play_route(int path);
- **DESCRIPTION**
正常音乐播放时接口
- **ARGUMENTS**
path: 设备名,
AUDIO_DEVICE_OUT_WIRED_HEADSET
AUDIO_DEVICE_OUT_WIRED_HEADPHONE
AUDIO_DEVICE_OUT_SPEAKER
- **RETURNS**
- **NOTE**
无

不同路径，设备声音控制

```
void normal_play_volume(int path, int vol);
void fm_volume(int path,int volume);
void phone_volume(int path, int volume);
```

例如：

- **PROTOTYPE**
- void phone_volume(int path, int volume)
- **DESCRIPTION**
电话声音接口
- **ARGUMENTS**
path: 设备名
- AUDIO_DEVICE_OUT_SPEAKER; 免提
- AUDIO_DEVICE_OUT_WIRED_HEADSET; // 带 mic 耳机
- AUDIO_DEVICE_OUT_WIRED_HEADPHONE; //不带 mic 耳机
- AUDIO_DEVICE_OUT_EARPIECE; //听筒

`volume`: 音量大小, 范围是 0~10.

- **RETURNS**
- **NOTE**

无

不同路径, 设备录音使能控制与路径控制

`void normal_record_enable(bool enable);`

`void normal_record_route(int path);`

`void phone_record_enable(bool enable);`

`void phone_record_route(int path);`

例如:

- **PROTOTYPE**
- `void phone_record_enable(bool enable)`
- **DESCRIPTION**
电话录音使能接口
- **ARGUMENTS**
`enable`: 是否使能
- **RETURNS**
- **NOTE**
无

- **PROTOTYPE**
- `void phone_record_route (int path)`
- **DESCRIPTION**
电话录音, 设备选择接口
- **ARGUMENTS**
`path`: 设备路径
- **RETURNS**
- **NOTE**
无

6. 客户使用详细说明

如果是普通平板方案，不带 BP 模块，无需配置。
以下针对带 BP 模块客户。

因为硬件客户硬件连接方案基本确定，只需更改 `ro.sw.audio.codec_plan_name` 属性。

客户所做主要工作为，1.切换 bp 声音输出接口（因为 bp 与 ap 连接时，可能是从 bp 端的听筒，也可能从 bp 端 pcm 输出声音，需发送相应的 at 命令做切换）。
2.调整声音效果。

步骤如下：

1. 查看原理图，确定 BP 模块，BT， AP 的连接方式。

2. 配置 `ro.sw.audio.codec_plan_name` 属性

已支持两种较通用方案

A. BP 通过 PHONEP/PHONEN 模拟接入
FM 通过 LINEINL/LINEINR 模拟接入
蓝牙通过 PCM 数字接到 AP
配置为 `ro.sw.audio.codec_plan_name=PLAN_ONE`.

B. BP 通过 I2S 或 PCM 数字接入
FM 通过 LINEINL/LINEINR 模拟接入
蓝牙通过 PCM 数字接到 AP

3. 配置 `ro.sw.audio.bp_device_name` 属性

已经有两种模组的代码结构，MU509 和 EM55。
软件配置为

`ro.sw.audio.bp_device_name=mu509` 或
`ro.sw.audio.bp_device_name=em55`

若没有，请参考步骤 5 或客户使用案例说明

4. 配置 `audio.without.earpiece` 属性

听筒连接，可能的方式有：

- a.直接连接到 AP, 配置为 0
- b.直接连接到 BP, 配置为 0
- c.没有任何听筒，配置为 1

5. 完成 bp 相关函数。

a.若 bp 是 mu509 或 em55, 则在相应 libril_audio 目录中修改如下接口。

```
//获得 bp 设备节点.,用于给设备发送命令
```

```
static int xxxx_get_tty_dev(char *name)
```

```
//不同路径下,BP 端声音设置调用
```

```
static int demo_set_call_volume(ril_audio_path_type_t path, int volume)
```

//切换 bp 声音输出接口，切换到听筒，或 i2s/pcm 数字接口时，需根据你的对应 at 命令修改此函数。

```
static int demo_set_call_path(ril_audio_path_type_t path)
```

b.若是其它 BP 模块，可以复制一份 libril_audio/bp_devices 目录下 demo 代码。

里面已经有基本实现，可根据不同 BP 不同的 at 指令集，修改。

5. 调节语音上行与下行音量

文件 phone_volume.conf 路径，类似
device/softwinner/fiber-common/hardware/audio/

- 配置语音下行音量，文件 phone_volume.conf 有详细说明
- 配置模拟 FM 下行音量，文件 phone_volume.conf 有详细说明
- 配置蓝牙上行音量，文件 phone_volume.conf 有详细说明
- 配置文件 sys_config1.fex 配置语音上行音量
 - headset_mic_vol = 6 //4 节耳机 mic 音量，范围 0~7
 - main_mic_vol = 6 //主 mic 音量，范围 0~7
- 配置 bp 音量，修改文件类似于 ibril_audio/bp_devices/demo/demo.c

```
static int demo_set_call_volume(ril_audio_path_type_t path, int volume)
```

//声音数组.分 6 级,顺序依次从第 1 级到第六级。数组值是音量大小。默认写的是

1,2,3,4,5,6.如果需要修改 bp 端输出音量，请修改，并发送相应 at 命令。

```
static int earpiece_vol[]={1,2,3,4,5,6};
```

```
static int headset_vol[]={1,2,3,4,5,6};
```

```
static int spk_vol[]={1,2,3,4,5,6};
```

```
static int bt_vol[]={1,2,3,4,5,6};  
static int main_mic_vol[]={1,2,3,4,5,6};  
static int headset_mic_vol[]={1,2,3,4,5,6};
```

Confidential

7. 客户使用案例 1

基本信息如下：

硬件连接为

BP 通过 PHONEP/PHONEN 模拟接入

FM 通过 LINEINL/LINEINR 模拟接入

蓝牙通过 PCM 数字接到 AP

BP 模块名为 BP_X

听筒连接到 AP

修改步骤：

1.配置 ro.sw.audio.codec_plan_name=PLAN_ONE.

2. 配置 ro.sw.audio.bp_device_name=bp_x

a.复制一份 libril_audio/bp_devices 目录下 demo 代码

demo.c 修改为 bp_x.c 替换 demo 字符。

demo.h 修改为 bp_x.h 替换 demo 字符。

b.bp_x.c 实现如下接口：

//获得 bp 设备节点.,用于给设备发送 AT 命令

```
static int bp_x_get_tty_dev(char *name)
```

//不同路径下,BP 端声音设置调用。

```
static int bp_x_set_call_volume(ril_audio_path_type_t path, int volume)
```

//路径切换调用，切换到听筒时，或免提接口时，需要调用此函数。

```
static int bp_x_set_call_path(ril_audio_path_type_t path)
```

因为大部分 bp 默认是从听筒输出的，BP 通过 PHONEP/PHONEN 模拟接入，是直接从听筒输出的。所以此种方案，可以不用此切换函数，可以为空。

c. libril_audio/Android.mk 修改

```
LOCAL_SRC_FILES := bp.c \
    bp_devices.c \
    bp_utils.c \
    bp_devices/mu509/mu509.c \
    bp_devices/oviphone_em55/em55.c \
    bp_devices/demo/demo.c \
++    bp_devices/bp_x/bp_x.c
```

d. libril_audio/bp_devices.c 修改

```
struct bp_device_desc bp_devices[] = {
{
    .name = "mu509",
```



```

        .board_name = "xxx1_plan",
        .ops = &mu509_ops,
        .other_ops = NULL,
    },
    {
        .name = "em55",
        .board_name = "xxx2_plan",
        .ops = &em55_ops,
        .other_ops = NULL,
    },
    {
        .name = "demo",
        .board_name = "demo_plan",
        .ops = &demo_ops,
        .other_ops = NULL,
+    }, {
+        .name = "bp_x",
+        .board_name = "bp_x",
+        .ops = &bp_x_ops,
+        .other_ops = NULL,
    }
};

```

e. libril_audio/bp_devices.h 修改

```

+ /* bp_x */
+ extern struct bp_ops bp_x_ops;

```

3. 配置 audio.without.earpiece=0

4. 调节语音上行与下行音量

8. 客户使用案例 2

基本信息如下：

硬件连接为

BP 通过 I2S 或 PCM 数字接入

FM 通过 LINEINL/LINEINR 模拟接入

蓝牙通过 PCM 数字接到 AP

BP 模块名为 BP_X

听筒连接到 AP

修改步骤：

1.配置 `ro.sw.audio.codec_plan_name=PLAN_TWO`.

2. 配置 `ro.sw.audio.bp_device_name=bp_x`

a.复制一份 `libril_audio/bp_devices` 目录下 `demo` 代码

`demo.c` 修改为 `bp_x.c` 替换 `demo` 字符。

`demo.h` 修改为 `bp_x.h` 替换 `demo` 字符。

b.`demo.c` 实现如下接口：

//获得 bp 设备节点.,用于给设备发送 AT 命令

```
static int bp_x_get_tty_dev(char *name)
```

//不同路径下,BP 端声音设置调用。

```
static int bp_x_set_call_volume(ril_audio_path_type_t path, int volume)
```

//路径切换调用，切换到听筒时，或免提接口时，需要调用此函数。

```
static int bp_x_set_call_path(ril_audio_path_type_t path)
```

因为大部分 bp 默认是从听筒输出的，而此处 BP 通过 I2S 或 PCM 数字接入。所以此种方案，需修改函数，发送 AT 命令，切换到 I2S 或 PCM 接口。

c. `libril_audio/Android.mk` 修改

```
LOCAL_SRC_FILES := bp.c \
    bp_devices.c \
    bp_utils.c \
    bp_devices/mu509/mu509.c \
    bp_devices/oviphone_em55/em55.c \
    bp_devices/demo/demo.c \
++    bp_devices/bp_x/bp_x.c
```

d. `libril_audio/bp_devices.c` 修改

```
struct bp_device_desc bp_devices[] = {
{
    .name = "mu509",
```

```

        .board_name = "xxx1_plan",
        .ops = &mu509_ops,
        .other_ops = NULL,
    },
    {
        .name = "em55",
        .board_name = "xxx2_plan",
        .ops = &em55_ops,
        .other_ops = NULL,
    },
    {
        .name = "demo",
        .board_name = "demo_plan",
        .ops = &demo_ops,
        .other_ops = NULL,
+    }, {
+        .name = "bp_x",
+        .board_name = "bp_x",
+        .ops = &bp_x_ops,
+        .other_ops = NULL,
    }
};

```

e. libril_audio/bp_devices.h 修改

```

+ /* bp_x */
+ extern struct bp_ops bp_x_ops;

```

3. 配置 audio.without.earpiece=0
4. 调节语音上行与下行音量

9. 客户使用案例 3

基本信息如下：

硬件连接为

BP 通过 PHONEP/PHONEN 模拟接入

FM 通过 LINEINL/LINEINR 模拟接入

蓝牙通过 PCM 数字接到 AP

BP 模块名为 BP_X

没有听筒

修改步骤：

1.配置 ro.sw.audio.codec_plan_name=PLAN_ONE.

2. 配置 ro.sw.audio.bp_device_name=bp_x

a.复制一份 libril_audio/bp_devices 目录下 demo 代码

demo.c 修改为 bp_x.c 替换 demo 字符。

demo.h 修改为 bp_x.h 替换 demo 字符。

b.bp_x.c 实现如下接口：

//获得 bp 设备节点.,用于给设备发送 AT 命令

```
static int bp_x_get_tty_dev(char *name)
```

//不同路径下,BP 端声音设置调用。

```
static int bp_x_set_call_volume(ril_audio_path_type_t path, int volume)
```

//路径切换调用，切换到听筒时，或免提接口时，需要调用此函数。

```
static int bp_x_set_call_path(ril_audio_path_type_t path)
```

因为大部分 bp 默认是从听筒输出的，BP 通过 PHONEP/PHONEN 模拟接入，是直接从听筒输出的。所以此种方案，可以不用此切换函数，可以为空。

c. libril_audio/Android.mk 修改

```
LOCAL_SRC_FILES := bp.c \
    bp_devices.c \
    bp_utils.c \
    bp_devices/mu509/mu509.c \
    bp_devices/oviphone_em55/em55.c \
    bp_devices/demo/demo.c \
++    bp_devices/bp_x/bp_x.c
```

d. libril_audio/bp_devices.c 修改

```
struct bp_device_desc bp_devices[] = {
{
    .name = "mu509",
```

```

        .board_name = "xxx1_plan",
        .ops = &mu509_ops,
        .other_ops = NULL,
    },
    {
        .name = "em55",
        .board_name = "xxx2_plan",
        .ops = &em55_ops,
        .other_ops = NULL,
    },
    {
        .name = "demo",
        .board_name = "demo_plan",
        .ops = &demo_ops,
        .other_ops = NULL,
+    }, {
+        .name = "bp_x",
+        .board_name = "bp_x",
+        .ops = &bp_x_ops,
+        .other_ops = NULL,
    }
};

```

e. libril_audio/bp_devices.h 修改

```

+ /* bp_x */
+ extern struct bp_ops bp_x_ops;

```

3. 配置 audio.without.earpiece=1
4. 调节语音上行与下行音量

10. 模块调试

一、回声问题（打开免提）：

1.可能的因素，模具设计有问题，音腔设计问题，是否选用硅 mic。

2.另外，可调小语音上行 mic 音量， 设置 `main_mic_vol = 0x0`

(参见，客户使用详细说明，配置文件 `sys_config1.fex` 配置语音上行音量)

3.可调小免提声音

二、使用 PLAN_ONE 时，蓝牙周围噪音过大

可调节 `phone_volume.conf` 中

[BLUETOOTH_UP]

`pcm_vol` =8 值越大，上行音量将越小，周围噪音也将变小。`pcm_vol` 仅对使用 PLAN_ONE 纯模拟时有效，使用 PLAN_TWO 纯数字时无效。

三、使用 PLAN_TWO 纯数字时，调节上行音量注意，部分调节无效

26 [PHONE_EARPIECE]

27 `phonepn_gain` =1,4,4,4,4,7 无效

28 `mixer_gain` =3,3,3,4,5,7 无效

29 `hp_gain` =60,60,60,60,60,60

30

31 [PHONE_HEADSET]

32 `phonepn_gain` =1,4,5,5,6,7 无效

33 `mixer_gain` =3,3,4,5,6,7 无效

34 `hp_gain` =35,38,40,45,47,50

35

36 [PHONE_SPEAKER]

37 `phonepn_gain` =1,5,5,5,6,7 无效

38 `mixer_gain` =3,3,5,5,6,7 无效

39 `spk_gain` =25,26,28,29,31,31

Declaration

**This 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