



D1 Linux AUDIOCODEC 开发指南

版本号: 1.0
发布日期: 2021.04.14

版本历史

版本号	日期	制/修订人	内容描述
1.0	2021.04.14	AWA1692	1. 添加 1.0 版 AUDIOCODEC 音频模块使用说明文档



目 录

1 前言	1
1.1 文档简介	1
1.2 目标读者	1
1.3 适用范围	1
1.4 相关术语	1
2 模块介绍	2
2.1 模块功能规格介绍	2
2.2 模块源码结构介绍	2
2.3 模块配置介绍	3
2.3.1 Device Tree 配置介绍	3
2.3.2 board.dts 板级配置介绍	4
3 模块使能说明	6
3.1 board.dts 模块使能	6
3.2 kernel menuconfig 使能	6
4 模块功能使用说明	11
4.1 模块声卡/设备查看说明	11
4.2 模块音频控件及通路配置说明	12
4.2.1 音频控件说明	12
4.2.2 Audio codec 模块音量调节说明	14
4.2.2.1 音频输出音量、增益控制	14
4.2.2.2 音频输入音量、增益控制	15
4.2.3 模块音频通路配置说明	16
4.3 模块功能验证说明	17
4.3.1 同源输出功能使用说明	17
4.3.2 LINEOUT 双通道喇叭输出	19
4.3.3 HPOUT 双通道耳机输出	19
4.3.4 MIC1~3 三通道录音输入	20
4.3.5 耳机 MIC 单通道录音输入	20
4.3.6 LINEIN 或 FMIN 双通道录音输入	20
4.3.7 耳机插拔检测功能	21
4.4 通路组合说明	21
5 FAQ	22

插 图

3-1 menuconfig Device Driver	7
3-2 menuconfig Sound card support	7
3-3 menuconfig Advanced	8
3-4 menuconfig ALSA	8
3-5 menuconfig Allwinner	9
3-6 menuconfig module	9
4-1 HP_OUT	16
4-2 HP_MIC	17



表 格

1-1 适用产品列表	1
2-1 模块 DTS 节点配置说明	4
2-2 模块 board.dts 板级配置文件配置说明	5
4-1 模块音频控件使用说明	13
4-2 耳机播放输出的音频控件及通路配置示例	17
4-3 耳机 MIC 录音输入的音频控件及通路配置示例	17
4-4 通路组合配置示例	21



1 前言

1.1 文档简介

本文档编写目的是为了音频系统相关的开发者能够了解清楚 AW SUNXI 平台下内置 audiocodec 接口的具体使用方法，能够更快地基于 AW SUNXI 平台完成对内置 audiocodec 接口的使用及二次开发等。

1.2 目标读者

音频系统相关开发人员。

1.3 适用范围

表 1-1: 适用产品列表

产品名称	内核版本	驱动文件
D1	Linux-5.4	sound/soc/sunxi/*

1.4 相关术语

- audiocodec：音频编解码器，包含 DAC/ADC，负责将音频信号数/模转换输出或是模/数转换输入；
- LINEOUT：AW SUNXI 平台 IC 的一种内置模拟音频输出接口，常用于外接喇叭输出等；
- HPOUT：AW SUNXI 平台 IC 的一种内置模拟音频输出接口，常用于耳机输出等；
- SPK：Speaker 的缩写，即喇叭；
- MIC：麦克风，录音采集用；
- DAC：数模转换器，用于实现数字信号向模拟信号转换；
- ADC：模数转换器，用于实现模拟信号向数字信号转换；
- 同源输出：AW SUNXI 平台的一种不同音频接口同时输出实现的硬件功能；
- TinyALSA：tiny + ALSA 即微型 ALSA 库，用于实现用户空间与内核空间的交互，能够实现播放/录音等基本功能，常用于 AW SUNXI 平台驱动层音频模块功能验证；

2 模块介绍

对 AW SUNXI 平台的内置 audiocodec 接口模块的基础介绍

2.1 模块功能规格介绍

AW SUNXI D1 平台 audiocodec 接口模块功能及规格:

- 支持两路 DAC 通道输出;
- 支持三路 ADC 通道输入;
- 支持一路立体声 HPOUT 输出;
- 支持一路 LINEOUT 差分输出;
- 支持三路差分 MIC 输入;
- 支持一路立体声 LINEIN 输入;
- 支持一路立体声 FMIN 输入;
- 支持同源输出功能;
- 播放输出支持多种采样率格式: 8KHz~192KHz;
- 录音输入支持多种采样率格式: 8KHz~48KHz;
- 播放最高可支持至 2 通道;
- 录音最高可支持至 3 通道;
- 播放/录音支持 8bit/16bit/20bit 数据精度;
- 支持耳机插拔检测功能;

2.2 模块源码结构介绍

模块驱动的源代码位于内核的/sound/soc/sunxi/目录下, 而一般 Audiocodec 模块会由于音频通路实现的差异, 驱动源码也会有差异, 以 D1 平台为例, audiocodec 的驱动源码结构示例如下所示:

```
lichee/linux-5.4/sound/soc/sunxi/  
├─ sun20iw1-codec.c           // Sunxi平台 codec 接口实现代码  
├─ sun20iw1-codec.h         // Sunxi平台 codec 驱动头文件  
├─ sun20iw1-sndcodec.c      // Sunxi平台 codec Machine 部分代码, 同时包含有耳机插拔检测功能实现部分的相关代码  
├─ sunxi-dummy-cpudai.c     // Sunxi平台 Platform 部分接口注册代码  
├─ sunxi-pcm.c              // Sunxi平台 platform 部分dma代码  
└─ sunxi-pcm.h              // Sunxi平台 platform 部分头文件
```

2.3 模块配置介绍

2.3.1 Device Tree 配置介绍

对应内核设备树中存在着每款芯片的所有平台的 audiocodec 模块配置，而 AW SUNXI 平台的设备树配置文件的路径为：

```
/tina/lichee/linux-5.4/arch/arm64/boot/dts/sunxi/CHIP.dtsi (64bit平台)
/tina/lichee/linux-5.4/arch/arm/boot/dts/CHIP.dtsi (32bit平台)
/tina/lichee/linux-5.4/arch/riscv/boot/dts/sunxi/CHIP.dtsi (riscv平台)
```

其中 CHIP 为研发代号，如 D1 的研发代号为 sun20iw1p1 等。

举例 D1 的设备树模块配置如下所示：

(/tina/lichee/linux-5.4/arch/riscv/boot/dts/sunxi/sun20iw1p1.dtsi)

```
codecs:codec@2030000 {
    #sound-dai-cells = <0>;
    compatible = "allwinner,sunxi-internal-codec";
    reg = <0x0 0x02030000 0x0 0x34c>;
    clocks = <&ccu CLK_PLL_AUDIO0>,
            <&ccu CLK_PLL_AUDIO1_DIV5>,
            <&ccu CLK_AUDIO_DAC>,
            <&ccu CLK_AUDIO_ADC>,
            <&ccu CLK_BUS_AUDIO_CODEC>;
    clock-names = "pll_audio0", "pll_audio1_div5",
                 "audio_clk_dac", "audio_clk_adc",
                 "audio_clk_bus";
    resets = <&ccu RST_BUS_AUDIO_CODEC>;
    playback_cma = <128>;
    capture_cma = <256>;
    device_type = "codec";
    status = "disable";
};

dummy_cpudai:dummy_cpudai@203034c {
    compatible = "allwinner,sunxi-dummy-cpudai";
    reg = <0x0 0x0203034c 0x0 0x4>;
    tx_fifo_size = <128>;
    rx_fifo_size = <256>;
    dac_txdata = <0x02030020>;
    adc_txdata = <0x02030040>;
    playback_cma = <128>;
    capture_cma = <256>;
    device_type = "cpudai";
    dmas = <&dma 7>, <&dma 7>;
    dma-names = "tx", "rx";
    status = "disable";
};

sndcodec:sound@2030340 {
    compatible = "allwinner,sunxi-codec-machine";
    reg = <0x0 0x02030340 0x0 0x4>;
    interrupts = <GIC_SPI 25 IRQ_TYPE_LEVEL_HIGH>;
```



```

sunxi,audio-codec = <&codec>;
sunxi,cpudai-controller = <&dummy_cpudai>;
device_type = "sndcodec";
status = "disable";
};

```

其中, 各项配置参数及其说明如下所示:

表 2-1: 模块 DTS 节点配置说明

节点配置	解释说明
reg	模块在 IC 中的模块基址及其最大偏移地址
clock	模块使用的时钟, 一般分别为时钟源及模块时钟
status	模块使能/关闭开关, "okay"使能, "disabled"关闭
其他	模块驱动注册绑定相关, 一般情况下不应改动

2.3.2 board.dts 板级配置介绍

board.dts 用于保存每一个板级平台的设备信息 (如 demo 板, perf1 板, ver 板等等), 里面的同名配置信息会覆盖上面的 DTS 设备树默认配置信息。

board.dts 板级配置文件路径为:

```
/tina/device/config/chips/IC/configs/BOARD/board.dts
```

举例 D1 的 board.dts 板级配置文件模块配置如下所示:

```
(/tina/device/config/chips/d1/configs/nezha/board.dts)
```

```

&codec {
    /* MIC and headphone gain setting */
    mic1gain    = <0x1F>;
    mic2gain    = <0x1F>;
    mic3gain    = <0x1F>;
    /* ADC/DAC DRC/HPF func enabled */
    /* 0x1:DAP_HP_EN; 0x2:DAP_SPK_EN; 0x3:DAP_HPSPK_EN */
    adcdrc_cfg  = <0x0>;
    adchpf_cfg  = <0x1>;
    dacdrc_cfg  = <0x0>;
    dachpf_cfg  = <0x0>;
    /* Volume about */
    digital_vol = <0x00>;
    lineout_vol = <0x1a>;
    headphonegain = <0x03>;
    /* Pa enabled about */
    pa_level    = <0x01>;
    pa_pwr_level = <0x01>;
    pa_msleep_time = <0x78>;
    /* gpio-spkr = <&gpio PF 2 GPIO_ACTIVE_HIGH>; */
    /* gpio-spkr-pwr = <&gpio PF 4 GPIO_ACTIVE_HIGH>; */
}

```

```

/* CMA config about */
    playback_cma    = <128>;
    capture_cma     = <256>;
/* regulator about */
/* avcc-supply = <&reg_aldol>; */
/* hpvcc-supply = <&reg_eldol>; */
    status = "okay";
};

&sndcodec {
    hp_detect_case = <0x00>;
    jack_enable    = <0x01>;
    status = "okay";
};

&dummy_cpudai {
    status = "okay";
};
    
```

其中，各项配置参数及其说明如下所示：

表 2-2: 模块 board.dts 板级配置文件配置说明 [®]

节点配置	解释说明
mic1gain	MIC1 录音增益默认配置，可选值：0~31，Min：0，Max：31
mic2gain	MIC2 录音增益默认配置，可选值：0~31，Min：0，Max：31
mic3gain	MIC3 录音增益默认配置，可选值：0~31，Min：0，Max：31
adcdrc_cfg	ADCDRC 功能开关选择，可选值：0/1/2/3，0 关闭其他开
adchpf_cfg	ADCHPF 功能开关选择，可选值：0/1/2/3，0 关闭其他开
dacdrp_cfg	DACDRC 功能开关选择，可选值：0/1/2/3，0 关闭其他开
dachpf_cfg	DACHPF 功能开关选择，可选值：0/1/2/3，0 关闭其他开
digital_vol	数字音量大小默认配置值，可选值：0~63，Min：0，Max：63
lineout_vol	LINEOUT 输出音量大小默认配置，可选值：0~31，Min：0，Max：31
headphonegain	HPOUT 输出音量大小默认配置，可选值：0~7，Min：0，Max：7
pa_level	外围功放使能电平，可选值：0/1，0：低电平使能；1：高电平使能
pa_pwr_level	外围功放供电使能电平，可选值：0/1，0：低电平使能；1：高电平使能
pa_msleep_time	等待功放工作正常的延时时间（ms），一般用于规避播放 pop 声，默认值 160ms
gpio-spk	外围功放芯片使能开关控制的 GPIO 脚，需对照实际外围电路实现进行配置
gpio-spk-pwr	外围功放芯片供电使能开关控制的 GPIO 脚，需对照实际外围电路实现进行配置
avcc-supply	外围音频输出电路相关电源管理
cpvin-supply	外围音频输出电路相关电源管理
hp_detect_case	耳机插拔检测逻辑选择，可选值：0/1，0：插入为低电平；1：插入为高电平
jack_enable	耳机总使能开关，可选值：0/1，0：不使用耳机功能；1：使用耳机功能
status	模块使能/关闭开关，“okay”使能，“disabled”关闭

3 模块使能说明

详细介绍模块使能的步骤

3.1 board.dts 模块使能

在相应的板级配置文件（board.dts）下，选择将 codec 节点及 sndcodec 节点下的“status”一并修改为“okay”并保存退出即可，具体修改示例如下所示：

```
&codec {
    ...
    status = "okay";    //codec 模块使能
};

&sndcodec {
    ...
    status = "okay";    //machine 模块使能
};

&dummy_cpudai {
    ...
    status = "okay";    //cpudai 模块使能
};
```

3.2 kernel menuconfig 使能

除了上述模块使能操作外，还需注意的是，需要保证内核配置的模块使能也已选中使能，具体操作步骤如下所示：

- 1、在 /tina/ 目录下执行“make kernel_menuconfig”命令进入内核配置界面。（需先选择对应平台）
- 2、选择 Device Drivers 选项进入下一级配置，如下图所示：

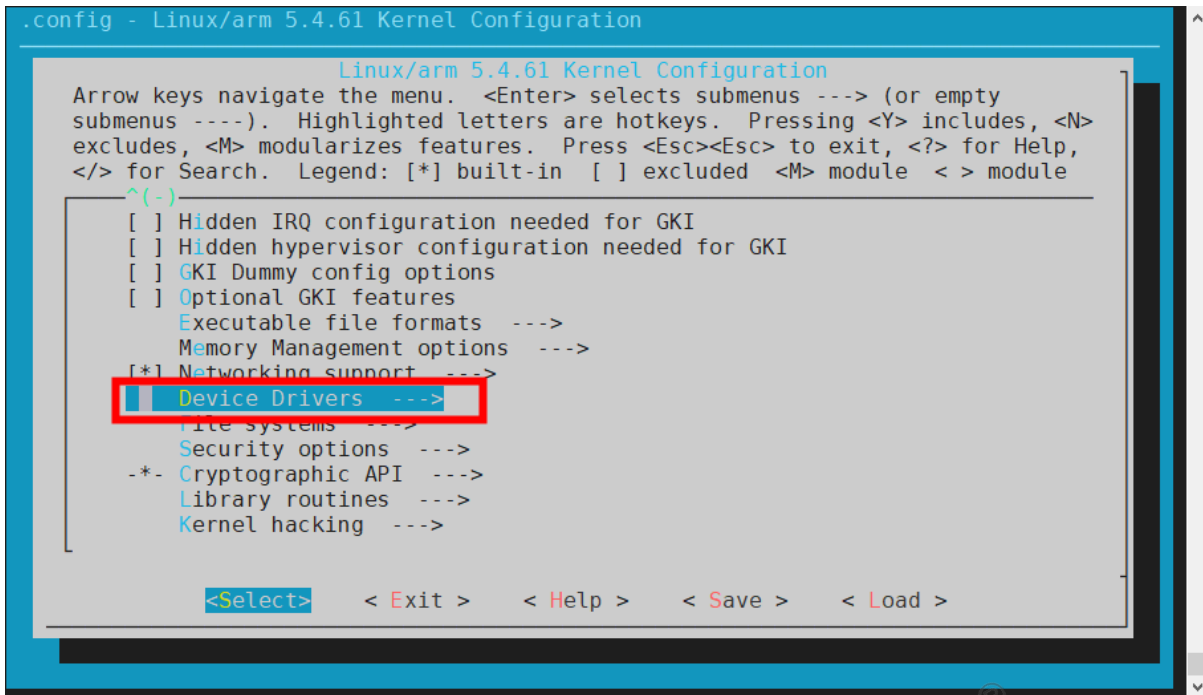


图 3-1: menuconfig Device Driver

3、选择 Sound card support 选项，进入下一级配置，如下图所示：

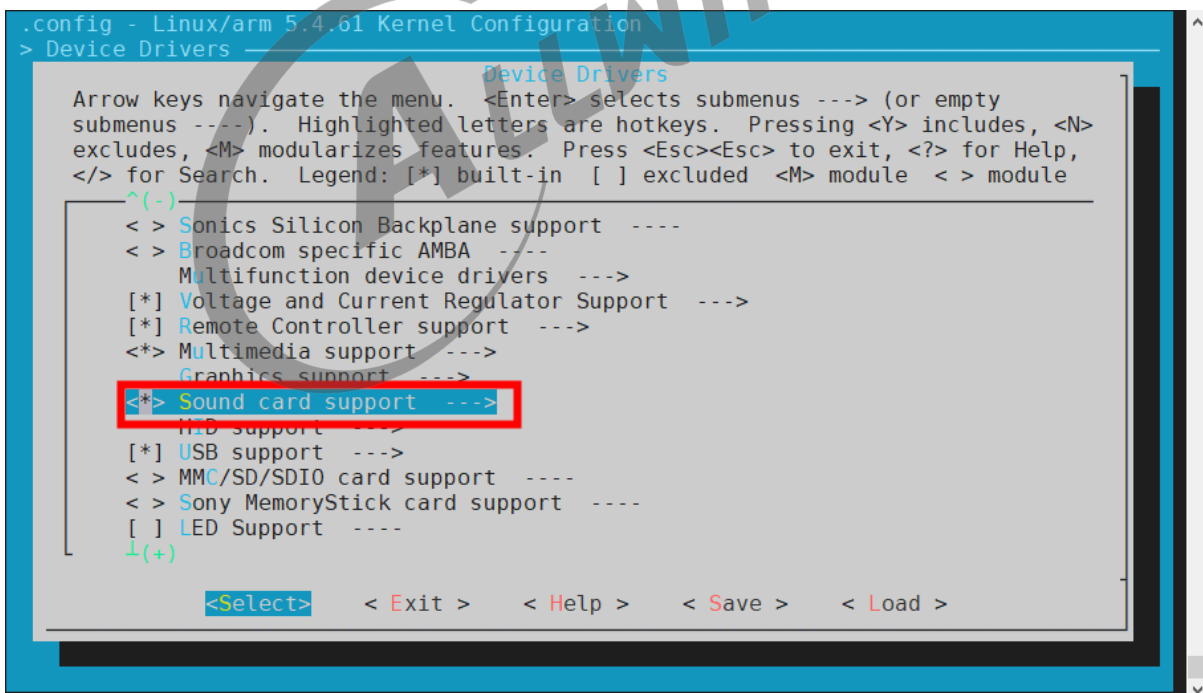


图 3-2: menuconfig Sound card support

4、选择 ALSA 框架，即 Advanced Linux Sound Architecture 选项，如下图所示：

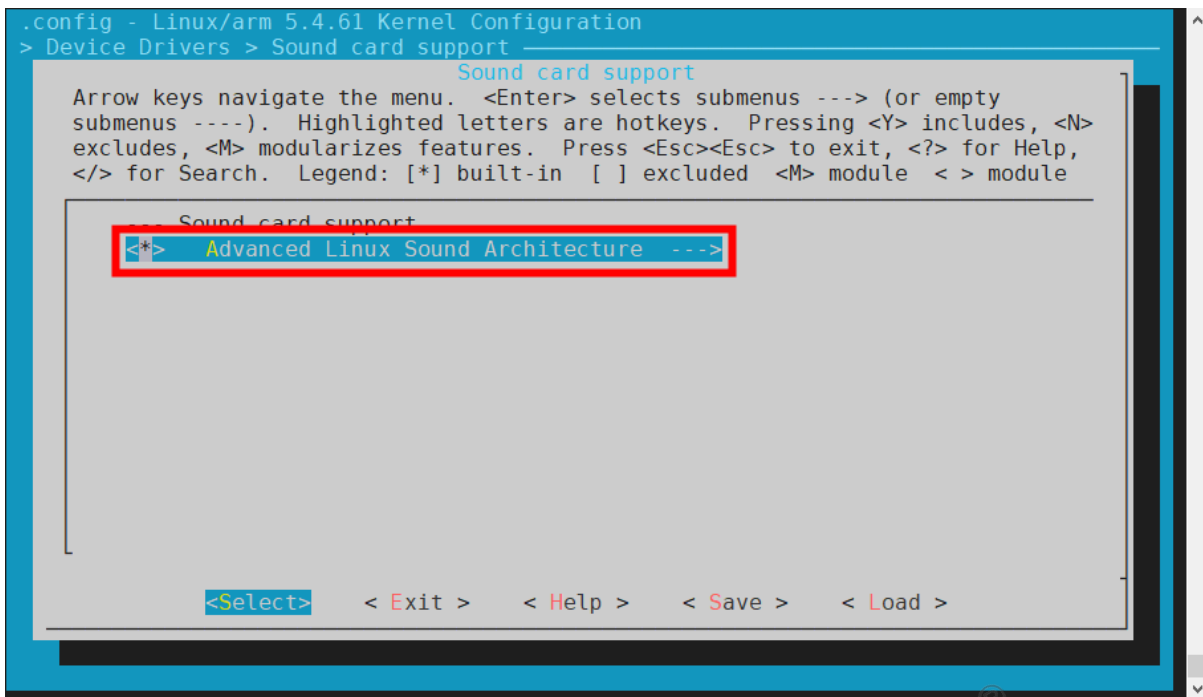


图 3-3: menuconfig Advanced

5、选择 ALSA for SoC audio support 选项，进入下一级配置，如下图所示：

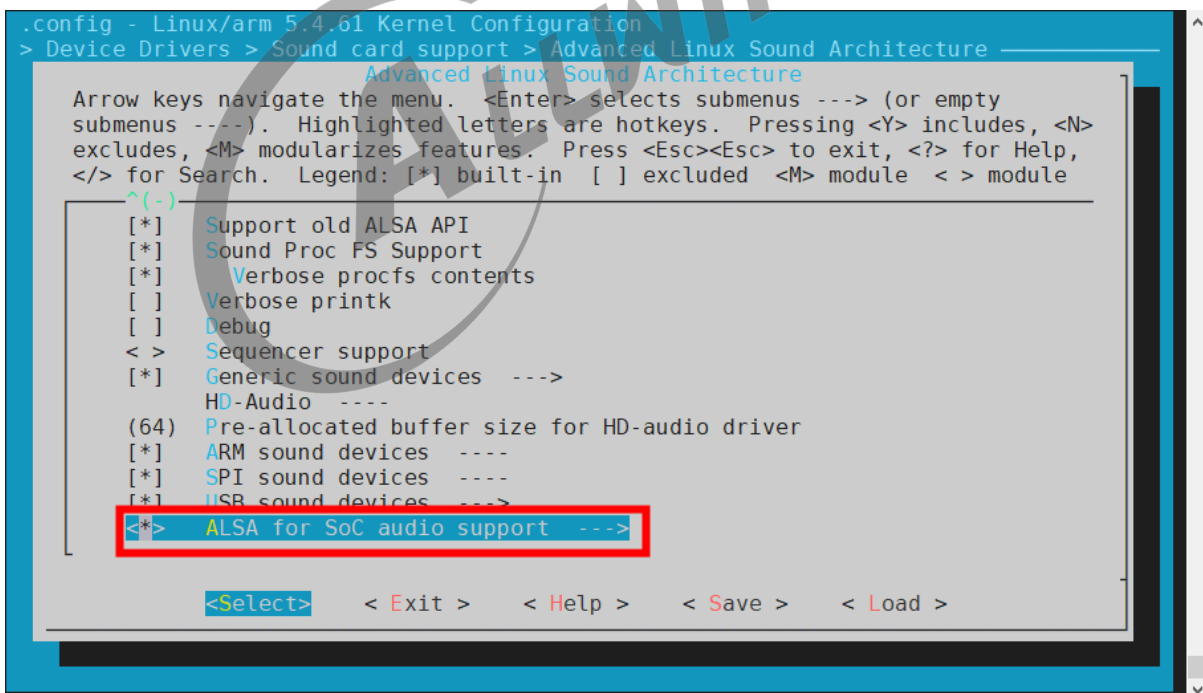


图 3-4: menuconfig ALSA

6、选择 Allwinner SoC Audio support 选项，如下图所示：

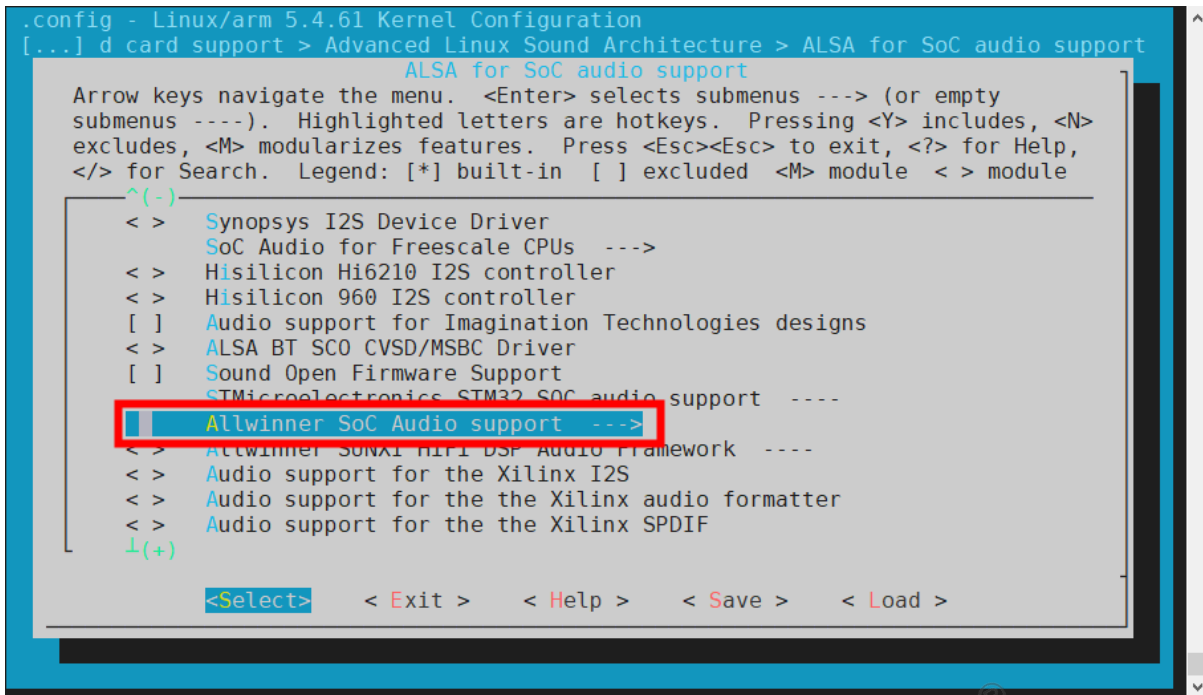


图 3-5: menuconfig Allwinner

7、选择需要的模块，可选择直接编译进内核，也可编译成模块。如下图所示：

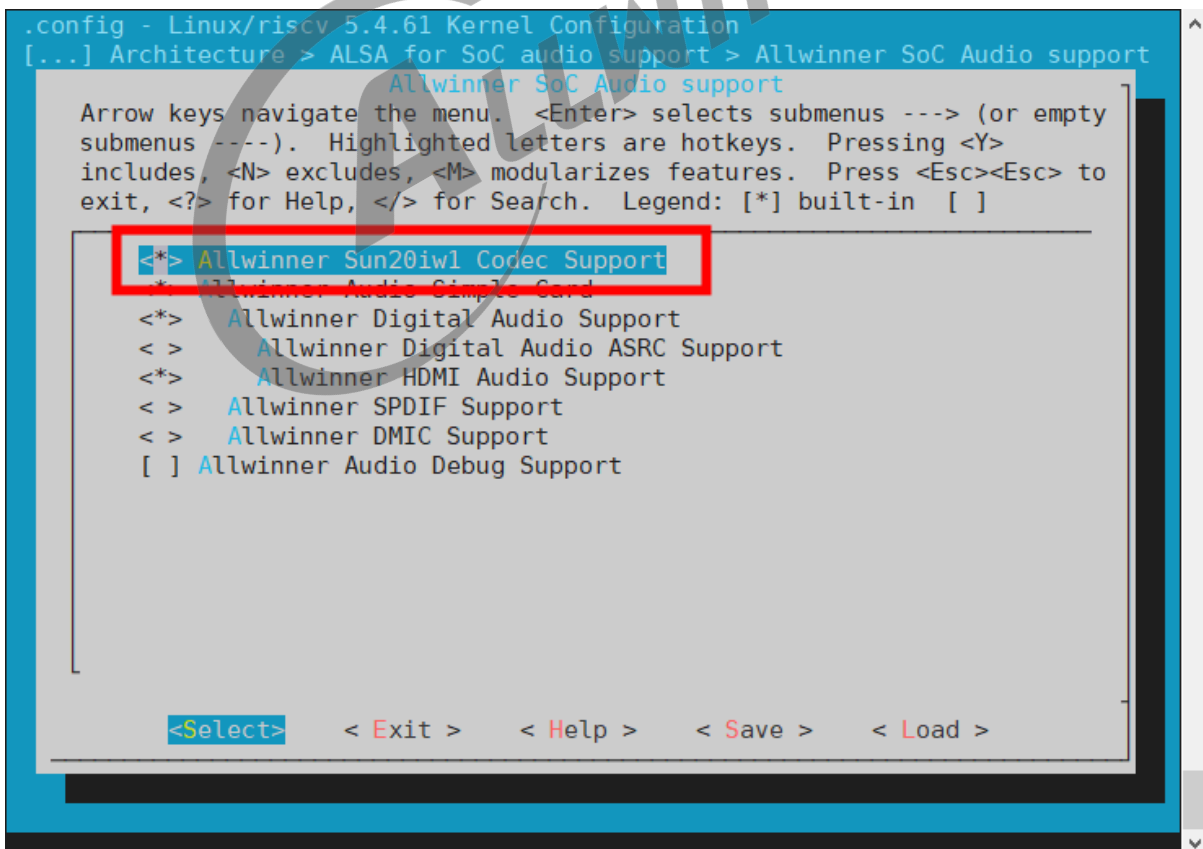


图 3-6: menuconfig module

综上，即可完成该模块的使能配置，重新编译烧录固件即可生成相应的模块声卡及设备（所使用的引脚与其它模块无冲突）。



4 模块功能使用说明

详细介绍模块接口的使用方法

4.1 模块声卡/设备查看说明

当相应的模块使能都打开并编译、烧录固件成功起来后，正常会生成相应的 audiocodec 声卡及设备，具体查看及确认操作如下所示：

```
/ # cat /proc/asound/cards
0 [audiocodec      ]: audiocodec - audiocodec      // audiocodec模块声卡
                        audiocodec

/ #
/ # ls -l /proc/asound/audiocodec/
total 0
-r--r--r-- 1 root root 0 2020-12-10 01:42 id      // audiocodec模块声卡ID名称
dr-xr-xr-x 3 root root 0 2020-12-10 01:42 pcm0c   // audiocodec模块声卡录音设备
dr-xr-xr-x 3 root root 0 2020-12-10 01:42 pcm0p   // audiocodec模块声卡播放设备
```

查看模块音频流的相关音频设置参数操作示例如下：

```
#查看播放参数（需在播放过程中查看）
/ # tinymix 31 1
/ # tinyplay tmp/test.wav &
/ # cat /proc/asound/audiocodec/pcm0p/sub0/hw_params
access: RW_INTERLEAVED
format: S16_LE                // 采样精度：16bit
subformat: STD
channels: 2                   // 通道数：2 channels
rate: 44100 (44100/1)         // 采样率：44.1KHz
period_size: 1024
buffer_size: 4096
/ # killall tinyplay

#查看录音参数（需在录音过程中查看）
/ # tinymix 22 1
/ # tinymix 25 1
/ # tinymix 28 1
/ # tinycap tmp/test.wav -c 3 &
/ # cat /proc/asound/audiocodec/pcm0c/sub0/hw_params
access: RW_INTERLEAVED
format: S16_LE                // 采样精度：16bit
subformat: STD
channels: 3                   // 通道数：3 channels
rate: 44100 (44100/1)         // 采样率：44.1KHz
period_size: 1024
buffer_size: 4096
/ # killall tinycap
```


4.2 模块音频控件及通路配置说明

本章说明将会基于 TinyALSA 工具的使用上进行说明

查看 audiocodec 模块音频控件列表及音频路由：

```
/ # cat /proc/asound/cards
0 [audiocodec      ]: audiocodec - audiocodec      // audiocodec模块声卡，声卡序号为0
                        audiocodec

/ #
/ # tinymix -D 0          // 查看默认声卡序号为0的声卡列表状态
Mixer name: 'audiocodec'
Number of controls: 32
ctl      type      num      name                                value
0        ENUM      1        codec hub mode                      hub_disable
1        ENUM      1        ADC1 ADC2 Swap                      Off
2        ENUM      1        ADC3 ADC4 swap                      Off
3        INT       1        digital volume                      63
4        INT       2        DAC volume                          160 160
5        INT       1        ADC1 volume                         160
6        INT       1        ADC2 volume                         160
7        INT       1        ADC3 volume                         160
8        INT       1        MIC1 gain volume                    31
9        INT       1        MIC2 gain volume                    31
10       INT       1        MIC3 gain volume                    31
11       BOOL      1        FMINL gain volume                   Off
12       BOOL      1        FMINR gain volume                   Off
13       BOOL      1        LINEINL gain volume                 Off
14       BOOL      1        LINEINR gain volume                 Off
15       INT       1        LINEOUT volume                      26
16       INT       1        Headphone Volume                    3
17       ENUM      1        LINEOUTL Output Select              DAC_DIFFER
18       ENUM      1        LINEOUTR Output Select              DAC_DIFFER
19       ENUM      1        MIC1 Input Select                   MIC_DIFFER
20       ENUM      1        MIC2 Input Select                   MIC_DIFFER
21       ENUM      1        MIC3 Input Select                   MIC_DIFFER
22       BOOL      1        ADC1 Input MIC1 Boost Switch        Off
23       BOOL      1        ADC1 Input FMINL Switch              Off
24       BOOL      1        ADC1 Input LINEINL Switch           Off
25       BOOL      1        ADC2 Input MIC2 Boost Switch        Off
26       BOOL      1        ADC2 Input FMINR Switch              Off
27       BOOL      1        ADC2 Input LINEINR Switch           Off
28       BOOL      1        ADC3 Input MIC3 Boost Switch        Off
29       BOOL      1        Headphone Switch                    Off
30       BOOL      1        HpSpeaker Switch                    Off
31       BOOL      1        LINEOUT Switch                      Off
```

4.2.1 音频控件说明

表 4-1: 模块音频控件使用说明

控件 序号	控件名称	配置可选值	控件说明
0	codec hub mode	0 (关闭) /1 (使能)	模块同源输出功能开关
1	ADC1 ADC2 Swap	0 (关闭) /1 (使能)	ADC1、ADC2 输入通道交换使能开关
2	ADC3 ADC4 swap	0 (关闭) /1 (使能)	ADC3、ADC4 输入通道交换使能开关
3	digital volume	0~63	DAC 输出数字音量大小控制， Max: 63; Min: 0
4	DAC volume	0~255	DAC 输出音量大小控制，默认 160 (一般不修改)
5	ADC1 volume	0~255	ADC1 输出数字音量大小控制，默认 160 (一般不修改)
6	ADC2 volume	0~255	ADC2 输出数字音量大小控制，默认 160 (一般不修改)
7	ADC3 volume	0~255	ADC3 输出数字音量大小控制，默认 160 (一般不修改)
8	MIC1 gain volume	0~31	MIC1 录音增益大小控制，Max: 31; Min: 0
9	MIC2 gain volume	0~31	MIC2 输入增益大小控制，Max: 31; Min: 0
10	MIC3 gain volume	0~31	MIC3 输入增益大小控制，Max: 31; Min: 0
11	FMINL gain volume	0 (关闭) /1 (使能)	FMIN 左声道输入增益开关
12	FMINR gain volume	0 (关闭) /1 (使能)	FMIN 右声道输入增益开关
13	LINEINL gain volume	0 (关闭) /1 (使能)	LINEIN 左声道输入增益开关
14	LINEINR gain volume	0 (关闭) /1 (使能)	LINEIN 右声道输入增益开关
15	LINEOUT volume	0~31	LINEOUT 输出音量大小控制， Max: 31; Min: 0
16	Headphone Volume	0~7	HPOUT 输出音量大小控制，Max: 7; Min: 0
17	LINEOUTL Output Select	0 (DAC_SINGLE) /1 (DAC_DIFFER)	LINEOUT 左声道单端输出和差分输 出选择，一般默认差分输出
18	LINEOUTR Output Select	0 (DAC_SINGLE) /1 (DAC_DIFFER)	LINEOUT 右声道单端输出和差分输 出选择，一般默认差分输出

控件序号	控件名称	配置可选值	控件说明
19	MIC1 Input Select	0 (MIC_DIFFER) /1 (MIC_SINGLE)	MIC1 单端输入和差分输入选择，一般默认差分输入
20	MIC2 Input Select	0 (MIC_DIFFER) /1 (MIC_SINGLE)	MIC2 单端输入和差分输入选择，一般默认差分输入
21	MIC3 Input Select	0 (MIC_DIFFER) /1 (MIC_SINGLE)	MIC3 单端输入和差分输入选择，一般默认差分输入
22	ADC1 Input MIC1 Boost Switch	0 (Off) /1 (On)	MIC1 输入开关选择
23	ADC1 Input FMINL Switch	0 (Off) /1 (On)	FMIN 左声道输入开关选择
24	ADC1 Input LINEINL Switch	0 (Off) /1 (On)	LINEIN 左声道输入开关选择
25	ADC2 Input MIC2 Boost Switch	0 (Off) /1 (On)	MIC2 输入开关选择 [®]
26	ADC2 Input FMINR Switch	0 (Off) /1 (On)	FMIN 右声道输入开关选择
27	ADC2 Input LINEINR Switch	0 (Off) /1 (On)	LINEIN 右声道输入开关选择
28	ADC3 Input MIC3 Boost Switch	0 (Off) /1 (On)	MIC3 输入开关选择
29	Headphone Switch	0 (Off) /1 (On)	headphone 输出开关选择
30	HpSpeaker Switch	0 (Off) /1 (On)	HPOUT 后接功放喇叭使能开关选择
31	LINEOUT Switch	0 (Off) /1 (On)	LINEOUT 后接功放喇叭使能开关选择

4.2.2 Audio codec 模块音量调节说明

4.2.2.1 音频输出音量、增益控制

1、输出数字音量控制：可调整相应 audiocodec 模块的音频输出数字音量，可调节范围：0~63，驱动默认最大值 63。

- 相应音频控件：digital volume

2、DAC 左右通道音量控制：可调整相应 audiocodec 模块的 DAC 的左右通道输出音量，可调节范围：0~255，驱动默认配置值 160（一般不修改）。

- 相应音频控件：DAC volume

3、HPOUT 耳机、喇叭输出增益控制：可调整相应 audiocodec 模块的 HPOUT 输出增益控制，可调节范围：0~7，驱动默认配置值 7。

- 相应音频控件：Headphone Volume

4、LINEOUT 输出音量控制：可调整相应 audiocodec 模块的 LINEOUT 输出的音量控制，可调节范围：0~31，驱动默认配置值 26。

- 相应音频控件：LINEOUT volume

注意：该音量控制仅用于 IC LINEOUT 信号后接的输出音量控制，其他情况无效。

4.2.2.2 音频输入音量、增益控制

1、ADC 左右通道音量控制：可调整相应 audiocodec 模块的 ADC 的左右通道输入音量，可调节范围：0~255，驱动默认配置值 160（一般不修改）。

- 相应音频控件：

- ADC1 volume
- ADC2 volume
- ADC3 volume

2、MIC1 输入增益控制：可调整相应 audiocodec 模块的 MIC1 的输入增益，可调节范围：0~31，驱动默认配置值 31。

- 相应音频控件：MIC1 gain volume

3、MIC2 输入增益控制：可调整相应 audiocodec 模块的 MIC2 的输入增益，可调节范围：0~31，驱动默认配置值 31。

- 相应音频控件：MIC2 gain volume

4、MIC3 输入增益控制：可调整相应 audiocodec 模块的 MIC3 的输入增益，可调节范围：0~31，驱动默认配置值 31。

- 相应音频控件：MIC3 gain volume

注意：若是需要进行音频输入、输出功率测试则需要事先确认相应的驱动层 audiocodec 模块的输入输出 DRC 功能配置已关闭。

4.2.3 模块音频通路配置说明

由于当前 audiocodec 模块从 IC 出来共有两路输出：LINEOUT 及 HPOUT，三路输入：MIC1/LINEINL/FMINL，MIC2/LINEINR/FMINR，MIC3，但实际最终的输入输出与外围电路的连接有着十分紧密的关系，基本来说音频外围输出输入电路的连接方式决定了音频通路配置的有效性，接下来会举例说明。

1、HPOUT 耳机播放输出通路配置通路

HPOUT 输出信号从 IC 内部出来后直接外接耳机输出电路再接耳机插座做耳机播放输出，具体通路实现原理图示例如下图所示：

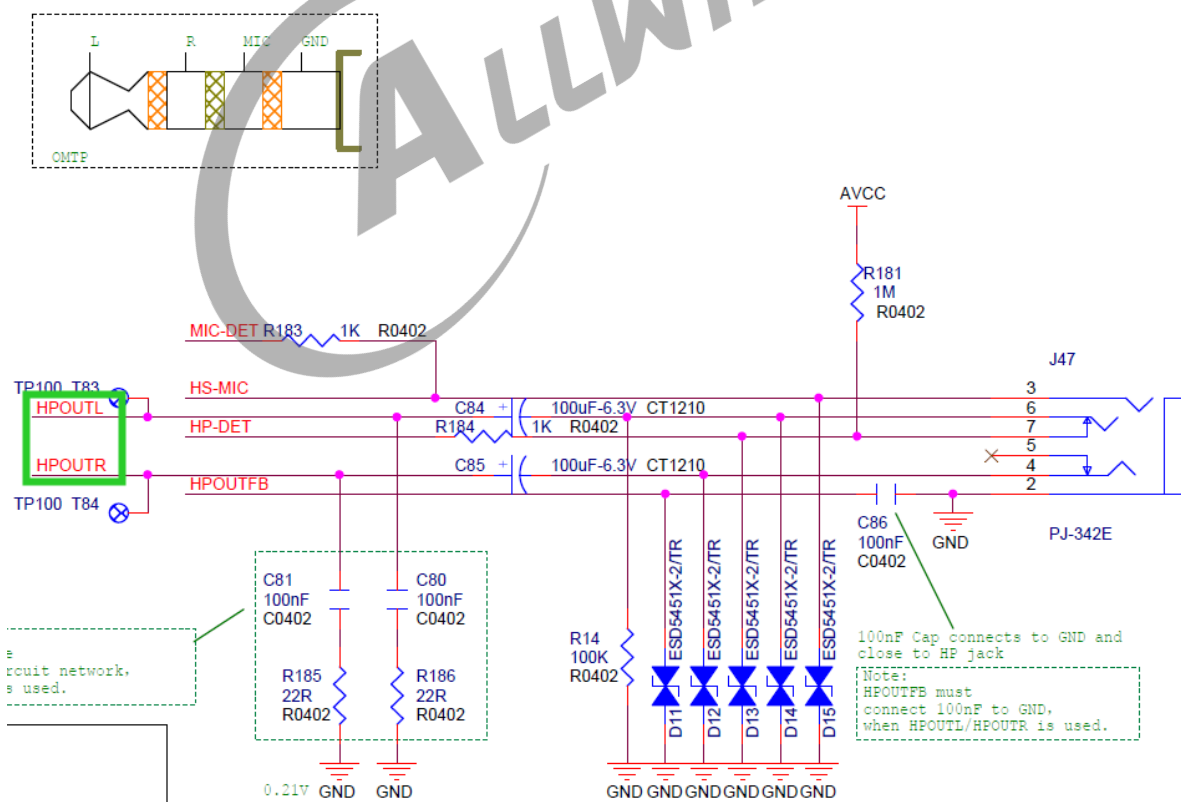


图 4-1: HP_OUT

表 4-2: 耳机播放输出的音频控件及通路配置示例

控件序号	控件名称	控件设置
16	Headphone Volume	3 //HPOUT 输出音量选择
13	Headphone Switch	on // HPOUT 输出开关打开

2、耳机麦克风录音输入通路配置

通过带麦耳机插入外接耳机插座连接至 MICP、MICN 输入到 IC 内部完成外围输入，具体通路实现原理图示例如下图所示：

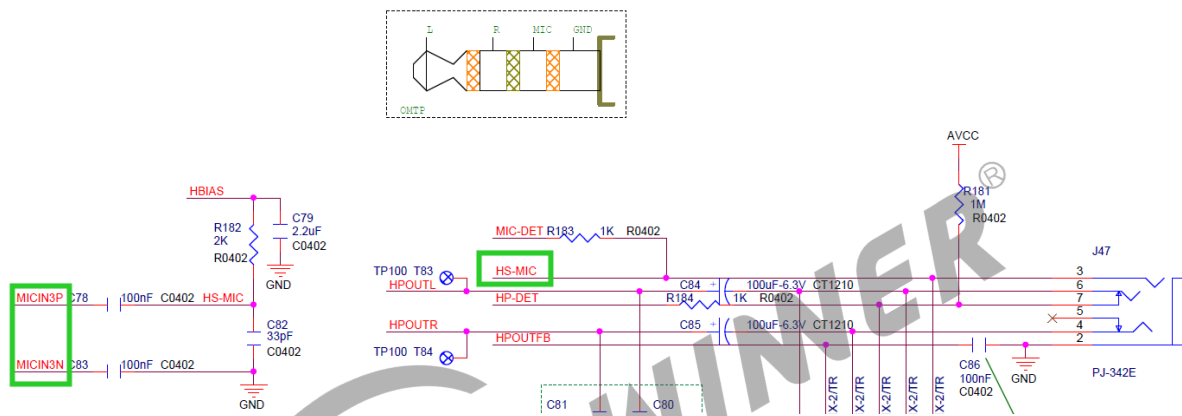


图 4-2: HP_MIC

表 4-3: 耳机 MIC 录音输入的音频控件及通路配置示例

控件序号	控件名称	控件设置
10	MIC3 gain volume	31 // MIC3 录音增益大小选择
21	MIC3 Input Select	MIC_DIFFER //差分输入方式
28	ADC3 Input MIC3 Boost Switch	on // MIC3 录音输入开关打开

4.3 模块功能验证说明

4.3.1 同源输出功能使用说明

同源输出功能：该功能是 AW SUNXI 平台通过硬件方法实现的一种能够让不同音频接口同时输出播放同一份音频数据的一个功能。

使用方法：

- 将需要进行同源输出的两个或多个声卡的同源输出控件“hub mode”一并使能打开
- 打开需要进行同源输出的两个或多个声卡（pcm_open）；
- 配置相关播放参数等（pcm_config）；
- 往其中的一个音频声卡开始写入数据（pcm_write）即可；
- 最后关闭相应的已打开的音频声卡（pcm_close）；

具体验证操作示例如下所示（以 audiocodec 和 sndspdif 两个声卡做同源输出为例）：

```

/ # cat /proc/asound/cards // 查看当前声卡序号
0 [audiocodec ]: audiocodec - audiocodec
  audiocodec
1 [snddmic ]: snddmic - snddmic
  snddmic
2 [snddaudio2 ]: snddaudio2 - snddaudio2
  snddaudio2
3 [sndspdif ]: sndspdif - sndspdif
  sndspdif

/ # tinymix -D 0 31 1 // 播放通路配置
/ # tinymix -D 0 0 1 // hub功能打开
/ # tinymix -D 0
Mixer name: 'audiocodec'
Number of controls: 32
ctl type num name value
0 ENUM 1 codec hub mode hub_enable
1 ENUM 1 ADC1 ADC2 Swap Off
2 ENUM 1 ADC3 ADC4 swap Off
3 INT 1 digital volume 63
4 INT 2 DAC volume 160 160
5 INT 1 ADC1 volume 160
6 INT 1 ADC2 volume 160
7 INT 1 ADC3 volume 160
8 INT 1 MIC1 gain volume 31
9 INT 1 MIC2 gain volume 31
10 INT 1 MIC3 gain volume 31
11 BOOL 1 FMINL gain volume Off
12 BOOL 1 FMINR gain volume Off
13 BOOL 1 LINEINL gain volume Off
14 BOOL 1 LINEINR gain volume Off
15 INT 1 LINEOUT volume 26
16 INT 1 Headphone Volume 3
17 ENUM 1 LINEOUTL Output Select DAC_DIFFER
18 ENUM 1 LINEOUTR Output Select DAC_DIFFER
19 ENUM 1 MIC1 Input Select MIC_DIFFER
20 ENUM 1 MIC2 Input Select MIC_DIFFER
21 ENUM 1 MIC3 Input Select MIC_DIFFER
22 BOOL 1 ADC1 Input MIC1 Boost Switch Off
23 BOOL 1 ADC1 Input FMINL Switch Off
24 BOOL 1 ADC1 Input LINEINL Switch Off
25 BOOL 1 ADC2 Input MIC2 Boost Switch Off
26 BOOL 1 ADC2 Input FMINR Switch Off
27 BOOL 1 ADC2 Input LINEINR Switch Off
28 BOOL 1 ADC3 Input MIC3 Boost Switch Off
29 BOOL 1 Headphone Switch Off
30 BOOL 1 HpSpeaker Switch Off
31 BOOL 1 LINEOUT Switch On

/ #
/ # tinymix -D 3 1 1 // hub功能打开
/ # tinymix -D 3

```

```

Mixer name: 'sndspdif'
Number of controls: 3
ctl      type      num      name                                     value
0        ENUM      1        spdif audio format function            PCM
1        ENUM      1        sunxi spdif hub mode                   Enable
2        BOOL      1        sunxi spdif loopback debug            Off
/ #
/ # tinyplay_ahub                        // 多声卡播放工具
Usage: tinyplay_ahub file.wav [-D card] [-d device] [-ad ahub_card] [-ad ahub_device] [-p
      period_size] [-n n_periods] [-m loop_minutes] [-i loop_num]
/ #
// 打开 audiocodec、sndspdif 声卡并往 audiocodec 声卡写数据
/ # tinyplay_ahub /tmp/test.wav -ad 0 -ad 0 -D 3 -d 0
Playing sample: 2 ch, 48000 hz, 16 bit, 1024 period, 4 count
Playing sample: 2 ch, 48000 hz, 16 bit
<----->
start --> loop_minutes = 0; loop_num = 1

^C/ #                                // 退出播放

```

4.3.2 LINEOUT 双通道喇叭输出

以 D1 nezha 板型为例，LINEOUT + SPK 喇叭播放输出，具体音频通路配置及播放验证操作示例如下所示：

```

/ # tinymix 15 31
/ # tinymix 17 1
/ # tinymix 18 1
/ # tinymix 31 1
/ #
/ # tinyplay /tmp/test.wav -D 0 -d 0
Playing sample: 2 ch, 48000 hz, 16 bit
^C/ #

```

4.3.3 HPOUT 双通道耳机输出

以 D1 nezha 板型为例，HPOUT + 耳机插座耳机播放输出，具体音频通路配置及播放验证操作示例如下所示：

```

/ # tinymix 16 7
/ # tinymix 29 1
/ #
/ # tinyplay /tmp/test.wav -D 0 -d 0
Playing sample: 2 ch, 48000 hz, 16 bit
^C/ #

```


4.3.4 MIC1~3 三通道录音输入

以 D1 nezha 板型为例，MIC1~3 为板载 MIC 录音输入，具体音频通路配置及录音验证操作示例如下所示：

```
/ # tinymix 8 31
/ # tinymix 9 31
/ # tinymix 10 31
/ # tinymix 19 1
/ # tinymix 20 1
/ # tinymix 21 1
/ # tinymix 22 1
/ # tinymix 25 1
/ # tinymix 28 1
/ #
/ # tinycap /tmp/test.wav -D 0 -d 0 -c 3 -r 48000 -b 16
Capturing sample: 3 ch, 48000 hz, 16 bit

^CCaptured 446464 frames
/ #
```

注：每个 ADC 通道只允许一个输入源，如 ADC1 的输入源使用 MIC1 为输入源，则需把 FMINL、LINEINL 关闭，其它 ADC 通道同理。

4.3.5 耳机 MIC 单通道录音输入

以 D1 nezha 板型为例，MIC1 为耳机 MIC 的接口，具体音频通路配置及录音验证操作示例如下所示：

```
/ # tinymix 8 31
/ # tinymix 19 1
/ # tinymix 22 1
/ #
/ # tinycap /tmp/test.wav -D 0 -d 0 -c 1 -r 48000 -b 16
Capturing sample: 1 ch, 48000 hz, 16 bit

^CCaptured 446464 frames
/ #
```

4.3.6 LINEIN 或 FMIN 双通道录音输入

以 D1 nezha 板型为例，具体音频通路配置及录音验证操作示例如下所示：

```
#LINEININ 双通道输入
/ # tinymix 24 1
/ # tinymix 27 1
/ #
/ # tinycap /tmp/test.wav -D 0 -d 0 -c 2 -r 48000 -b 16
Capturing sample: 2 ch, 48000 hz, 16 bit
```

```

^CCaptured 446464 frames
/ #

#FMININ 双通道输入
/ # tinymix 23 1
/ # tinymix 26 1
/ #
/ # tinycap /tmp/test.wav -D 0 -d 0 -c 2 -r 48000 -b 16
Capturing sample: 2 ch, 48000 hz, 16 bit

^CCaptured 446464 frames
/ #

```

4.3.7 耳机插拔检测功能

耳机插拔检测功能已集成在相应 audiocodec 驱动中，可通过内核 board.dts 配置项中打开，默认情况下使能打开，具体方法见“board.dts”板级配置介绍。

4.4 通路组合说明

D1 支持最多 3 路录音 + 2 路播放同时进行，如下：

表 4-4: 通路组合配置示例

通路组合	通路名称
1 路录音	MIC1、MIC2、MIC3、FMINL、FMINR、LINEINL、LINEINR
2 路录音	MIC1、FMINL、LINEINL + MIC2、FMINR、LINEINR MIC1、FMINL、LINEINL + MIC3 MIC2、FMINR、LINEINR + MIC3
3 路录音	MIC1、FMINL、LINEINL + MIC2、FMINR、LINEINR + MIC3
1 路播放	LINEOUTL、LINEOUTR、HPOUTL、HPOUTR
2 路播放	LINEOUTL + LINEOUTR、HPOUTL、HPOUTR LINEOUTR + LINEOUTL、HPOUTL、HPOUTR HPOUTL + LINEOUTL、LINEOUTR、HPOUTR HPOUTR + LINEOUTL、LINEOUTR、HPOUTL

5 FAQ

- 按要求进行 menuconfig 配置，并且在 board.dts 将 audiocodec 打开，但无声卡生成。
- 查看 audiocodec 所使用的引脚是否被其它模块占用。






著作权声明

版权所有 © 2021 珠海全志科技股份有限公司。保留一切权利。

本档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本档内容的部分或全部，且不得以任何形式传播。

商标声明

、 **全志科技** （不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本档作为使用指导仅供参考。由于产品版本升级或其他原因，本档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本档中提供准确的信息，但并不确保内容完全没有错误，因使用本档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。