



# **D1-H Linux SPDIF 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 S/PDIF 音频模块使用说明文档



# 目 录

<b>1 前言</b>	<b>1</b>
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
<b>2 模块介绍</b>	<b>2</b>
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	2
2.3.1 Device Tree 配置介绍	2
2.3.2 board.dts 板级配置介绍	4
<b>3 模块使能说明</b>	<b>5</b>
3.1 board.dts 模块使能	5
3.2 kernel menuconfig 使能	5
<b>4 模块功能使用说明</b>	<b>9</b>
4.1 模块声卡/设备查看	9
4.2 模块音频控件及通路配置说明	9
4.2.1 模块音频控件说明	10
4.2.2 模块音频通路配置说明	10
4.3 模块功能验证说明	10
4.3.1 同源输出功能使用说明	10
4.3.2 SPDIF-OUT 播放	11
4.3.3 SPDIF-IN 录音	12
4.3.4 SPDIF loopback 回环测试	12
4.3.5 SPDIF 透传播放	12
<b>5 FAQ</b>	<b>14</b>

## 插 图

3-1 Device Driver . . . . .	6
3-2 Sound . . . . .	6
3-3 Advanced . . . . .	7
3-4 ALSA . . . . .	7
3-5 Allwinner . . . . .	8
3-6 module . . . . .	8



# 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 S/PDIF 接口的具体使用方法，能够更快地基于 AW SUNXI 平台完成对 S/PDIF 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-H	Linux-5.4	sound/soc/*

## 1.4 相关术语

- S/PDIF：外置音响音频设备接口，一般使用同轴电缆或光纤接口；
- 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；
- 音频透传：即 Audio Passthrough，是一种不通过解码将压缩数据直通输出给外部专业硬件设备解码音频流播放的音频技术；
- 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；

## 2 模块介绍

对 AW SUNXI 平台的 S/PDIF 接口模块的基础介绍

### 2.1 模块功能规格介绍

AW SUNXI 平台 S/PDIF 接口模块功能及规格：

- 常规支持 playback 播放功能；
- 常规支持 record 录音功能；
- 支持音频透传输出播放功能；
- 支持内部回环 loopback 测试；
- 播放支持多种采样率格式 (44.1KHz, 48KHz, 96KHz, 192KHz)；
- 录音支持多种采样率格式 (8KHz, 11.025KHz, 16KHz, 22.05KHz, 24KHz, 32KHz, 44.1KHz, 48KHz)；
- 常规播放/录音最高可支持至 2 通道；
- 支持 16bit/20bit/24bit 数据精度；

### 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，其源码结构介绍如下所示：

```
/tina/lichee/linux-5.4/sound/soc/  
├── sunxi                               // Sunxi平台  
│   ├── sunxi-spdif.c                 // Sunxi平台SPDIF接口代码  
│   ├── sunxi-spdif.h                 // Sunxi平台SPDIF驱动头文件  
│   ├── sunxi-pcm.c                   // Sunxi平台platform部分dma代码  
│   ├── sunxi-pcm.h                   // Sunxi平台platform部分dma代码头文件  
└── sunxi-simple-card.c               // Sunxi平台machine部分代码
```

### 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 SPDIF 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-H 的研发代号为 sun20iw1p1 等。

举例 D1-H 的设备树模块配置如下所示：

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

```
spdif:spdif@2036000 {
    #sound-dai-cells = <0>;
    compatible = "allwinner,sunxi-spdif";
    reg = <0x0 0x02036000 0x0 0x58>;
    clocks = <&ccu CLK_PLL_AUDIO0>,
            <&ccu CLK_SPDIF_TX>,
            <&ccu CLK_BUS_SPDIF>,
            <&ccu CLK_PLL_AUDIO1>,
            <&ccu CLK_PLL_AUDIO1_DIV5>,
            <&ccu CLK_SPDIF_RX>;
    clock-names = "pll_audio", "spdif", "spdif_bus",
                  "pll_audio1", "pll_audio1_div5", "spdif_rx";
    resets = <&ccu RST_BUS_SPDIF>;
    dmas = <&dma 2>, <&dma 2>;
    dma-names = "tx", "rx";
    interrupts-extended = <&plic0 41 IRQ_TYPE_LEVEL_HIGH>;
    clk_parent = <0x1>;
    playback_cma = <128>;
    capture_cma = <128>;
    device_type = "spdif";
    status = "disabled";
};

soundspdif:soundspdif@2036040 {
    reg = <0x0 0x02036040 0x0 0x4>;
    compatible = "sunxi,simple-audio-card";
    simple-audio-card,name = "sndspdif";
    status = "disabled";
    /* simple-audio-card,format = "i2s"; */
    simple-audio-card,cpu {
        sound-dai = <&spdif>;
    };
    simple-audio-card,codec {
        /*snd-soc-dummy*/
    };
};
```

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

## 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-H 的 board.dts 板级配置文件模块配置如下所示：

```
(/longon/device/config/chips/d1-h/configs/nezha/board.dts)
```

```
spdif_pins_a: spdif@0 {
    /* SPDIF_PIN: SPDIF_OUT */
    pins = "PB0";
    function = "spdif";
    drive-strength = <20>;
    bias-disable;
};

spdif_pins_b: spdif_sleep@0 {
    pins = "PB0";
    function = "io_disabled";
    drive-strength = <20>;
    bias-disable;
};

&spdif {
    pinctrl-names = "default", "sleep";
    pinctrl-0 = <&spdif_pins_a>;
    pinctrl-1 = <&spdif_pins_b>;
    status = "okay";
};

&soundspdif {
    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
pins	模块需要使用到的引脚组定义
function	模块引脚组复用功能
drive-strength	模块引脚驱动力，默认配置为 20 即可
bias-disable	失能上下拉



## 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 spdif 节点及 sndspdif 节点下的 “status” 修改为 “okay” 并保存退出即可，具体修改示例如下所示：

```
&spdif {  
    ...  
    status = "okay";  
};  
  
&soundspdif {  
    ...  
    status = "okay";  
};
```

### 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：

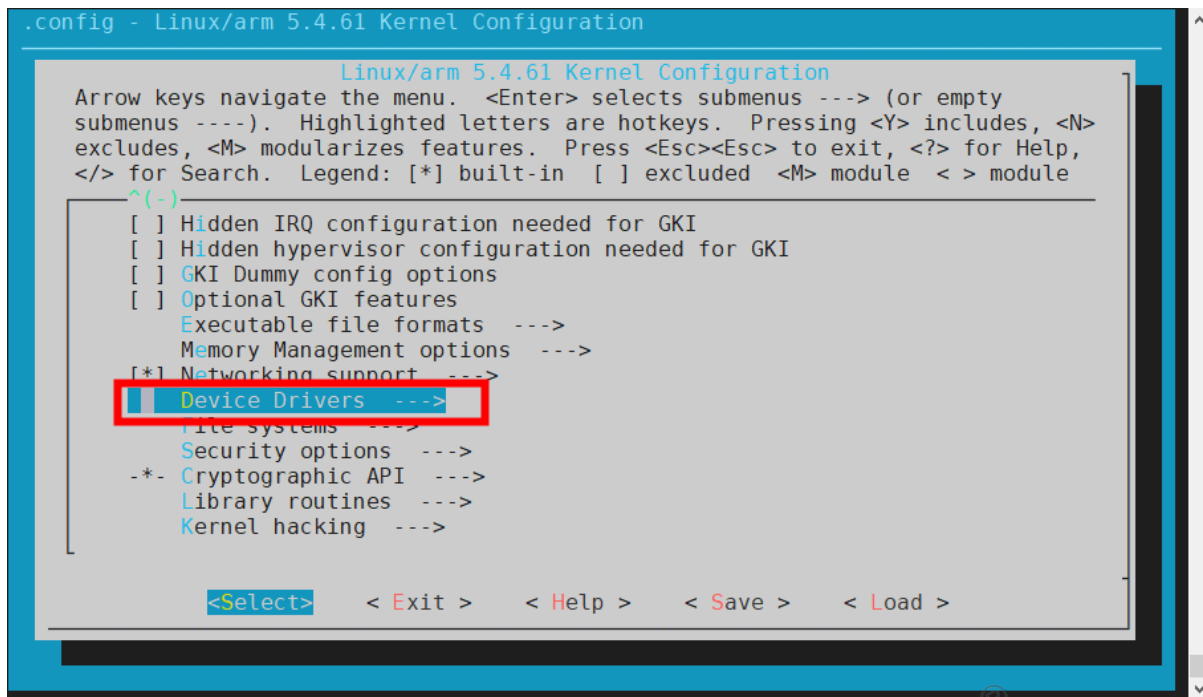


图 3-1: Device Driver

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

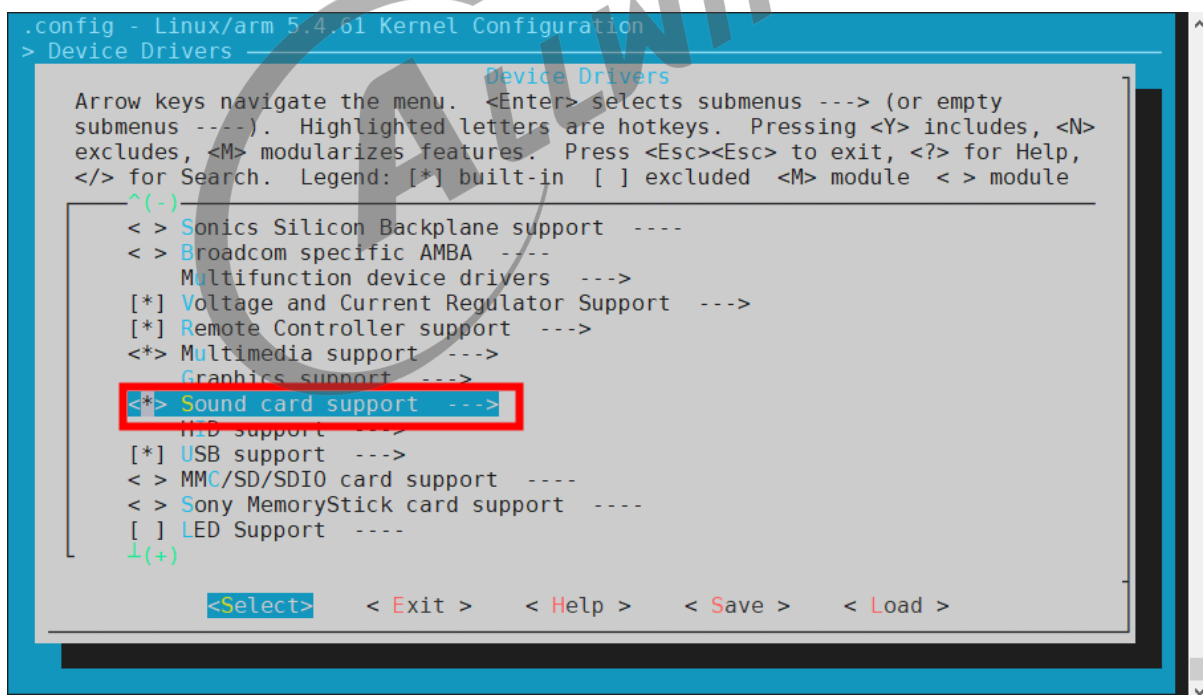


图 3-2: Sound

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

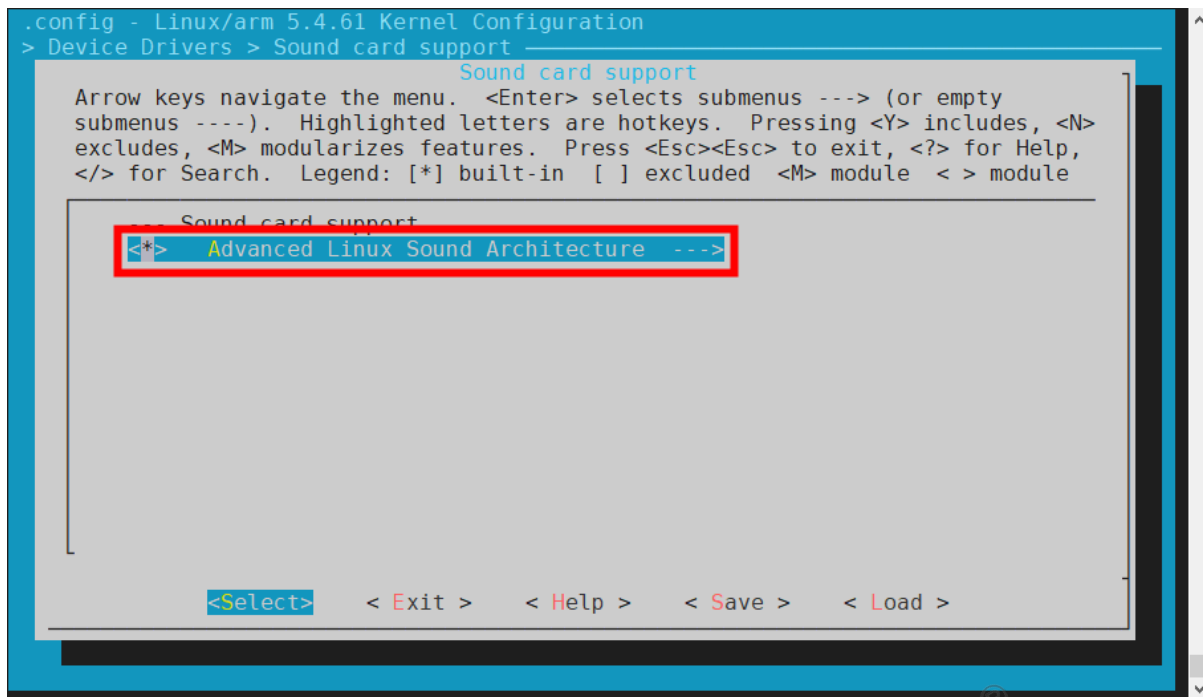


图 3-3: Advanced

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

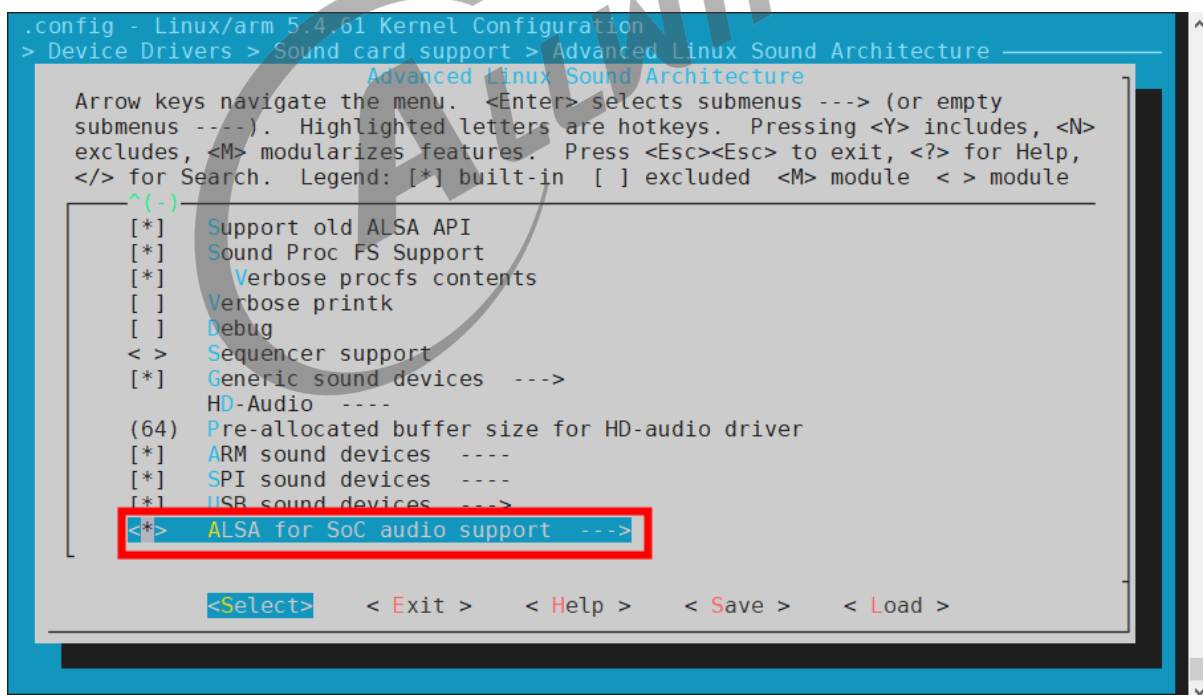


图 3-4: ALSA

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

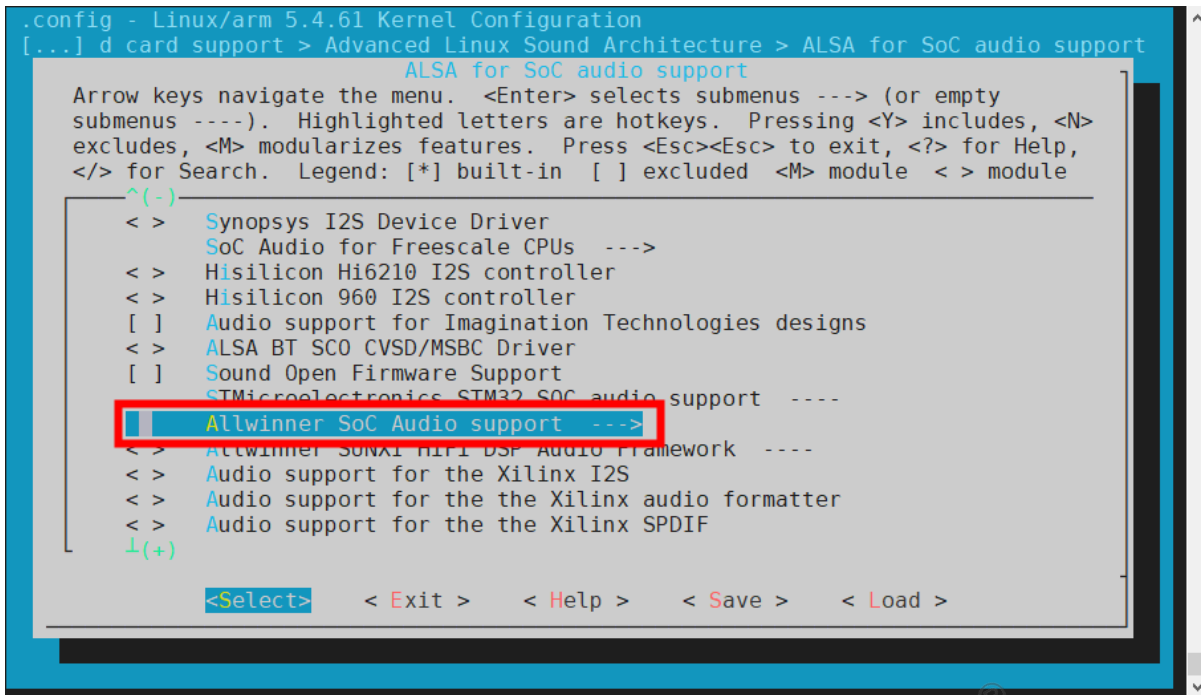


图 3-5: Allwinner

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：

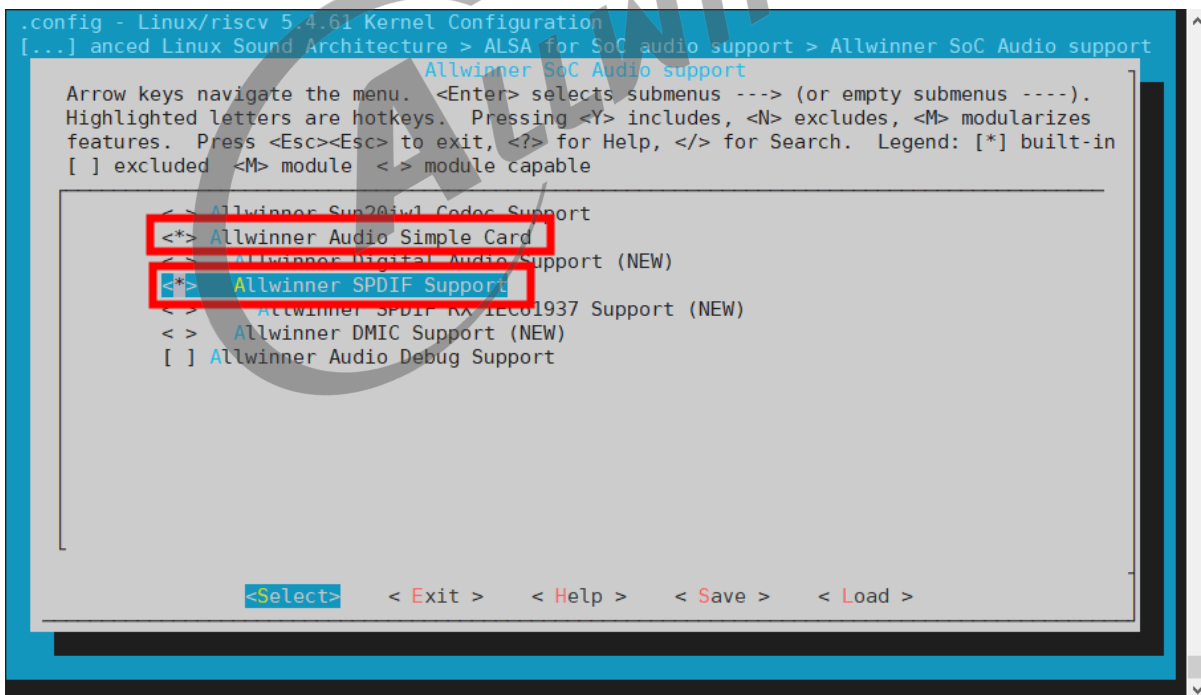


图 3-6: module

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

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 spdif 声卡及设备，具体查看及确认操作如下所示：

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

/ #
/ # ls -l /proc/asound/sndspdif/
total 0
-r--r--r--    1 root    root          0 Jan  1 00:00 id      // spdif模块声卡ID名称
dr-xr-xr-x    3 root    root          0 Jan  1 00:00 pcm0c   // spdif模块声卡录音设备
dr-xr-xr-x    3 root    root          0 Jan  1 00:00 pcm0p   // spdif模块声卡播放设备
/ #
```

查看模块音频流的相关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）
/ # tinyplay tmp/test.wav &
/ # cat /proc/asound/sndspdif/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
```

### 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 spdif 模块声卡音频控件列表及音频路由：

```
/ # cat /proc/asound/cards
0 [sndspdif      ]: sndspdif - sndspdif
                        sndspdif

/ #
/ # tinymix -D 0          // 查看默认声卡序号为0的声卡列表状态
```

```
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               Disable
2        BOOL    1      sunxi spdif loopback debug         Off
/ #
```

## 4.2.1 模块音频控件说明

模块音频控件使用说明如下所示：

音频控件序号	音频控件名称	配置可选值	音频控件说明
0	spdif audio format Function	PCM/DTS	SPDIF 接口播放输出音频类型选择
1	sunxi spdif hub mode	0（关闭）/1（使能）	SPDIF 接口同源输出开关选择
2	sunxi spdif loopback debug	0（关闭）/1（使能）	SPDIF 接口回录开关选择

## 4.2.2 模块音频通路配置说明

由于 SPDIF 的输入输出通路都是直通型，音频驱动也未提供任何路由控件来做通路选择，所以操作上直接对模块声卡设备进行读/写、播/录操作即可。

## 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“sunxi spdif hub mode”等一并使能打开；
- 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；

具体验证操作示例如下所示：

```

/ # 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
...
15        INT       1         LINEOUT volume                      26
...
31        BOOL      1         LINEOUT Switch                      0n

/ #
/ # 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 SPDIF-OUT 播放

注意：SPDIF 接口播放验证需要通过板子的光纤头输出与数字功放连接并最后通过与数字功放连接喇叭播放出来，在播放验证前请注意将相关硬件设备及连线环境等准备好，减少不必要的工作。

具体功能验证操作示例如下所示：

```

/ # cat /proc/asound/cards
0 [sndspdif        ]: sndspdif - sndspdif           // 查看spdif声卡序号
                                sndspdif

/ # tinymix 0 1
/ # tinyplay

```

```
Usage: tinypplay file.wav [-D card] [-d device] [-p period_size] [-n n_periods]
/ #
/ # tinypplay /tmp/test.wav -D 0 -d 0 // 使用声卡序号为0的声卡的pcm0p设备播放/tmp/
    test.wav音频
Playing sample: 2 ch, 44100 hz, 16 bit 50572800 bytes // 播放过程中
^C/ # // Ctrl + C 退出播放程序，停止播放
```

### 4.3.3 SPDIF-IN 录音

注：录音操作类似播放操作，且一般用不上 SPDIF-IN 做录音输入，故在此不多做操作说明。

### 4.3.4 SPDIF loopback 回环测试

具体功能验证操作示例如下所示：

```
/ # cat /proc/asound/cards
0 [sndspdif      ]: sndspdif - sndspdif // 查看spdif声卡序号
    sndspdif
/ # tinymix 3 1
/ #
/ # tinypplay /tmp/play.wav -D 0 -d 0 & // 使用声卡序号为0的声卡的pcm0p设备后台播放/
    tmp/play.wav音频
Playing sample: 2 ch, 44100 hz, 16 bit 50572800 bytes // 播放过程中
/ # tinycap /tmp/cap.wav -D 0 -d 0
Capturing sample: 2 ch, 44100 hz, 16 bit
^C/ # // Ctrl + C 退出录音程序，停止录音
/ # killall tinypplay // 结束后台播放
```

### 4.3.5 SPDIF 透传播放

注意：

(1) SPDIF 接口透传播放验证需要通过板子的光纤头输出与数字功放连接并最后通过与数字功放连接喇叭播放出来，在播放验证前请注意将相关硬件设备及连线环境等准备好，减少不必要的工作。

(2) 纯驱动层实现 SPDIF 接口透传播放需要修改相应的 tinyalsa 工具方能实现。

具体功能验证操作示例如下所示：

```
/ # cat /proc/asound/cards
0 [sndspdif      ]: sndspdif - sndspdif // 查看spdif声卡序号
    sndspdif
/ # tinymix 0 2
/ # tinypplayrww /tmp/DTS -D 0 -d 0 // 使用声卡序号为0的声卡的pcm0p设备播放/tmp/DTS透
    传音频
Playing sample: 2 ch, 44100 hz, 16 bit 50572800 bytes // 播放过程中
```



^C/ #

// Ctrl + C 退出播放程序，停止播放



## 5 FAQ

---

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



## 著作权声明

版权所有 © 2022 珠海全志科技股份有限公司。保留一切权利。

本文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。

本文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、发表或传播本文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。在本文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。本文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，并严格遵循本文档的使用说明。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。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（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。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文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。