



# Unisoc Confidential For hiar

## Audio DT 介绍

文档版本	V1.0
发布日期	2021-02-05

**版权所有 © 紫光展锐（上海）科技有限公司。保留一切权利。**

本文件所含数据和信息都属于紫光展锐（上海）科技有限公司（以下简称紫光展锐）所有的机密信息，紫光展锐保留所有相关权利。本文件仅为信息参考之目的提供，不包含任何明示或默示的知识产权许可，也不表示有任何明示或默示的保证，包括但不限于满足任何特殊目的、不侵权或性能。当您接受这份文件时，即表示您同意本文件中内容和信息属于紫光展锐机密信息，且同意在未获得紫光展锐书面同意前，不使用或复制本文件的整体或部分，也不向任何其他方披露本文件内容。紫光展锐有权在未经事先通知的情况下，在任何时候对本文件做任何修改。紫光展锐对本文件所含数据和信息不做任何保证，在任何情况下，紫光展锐均不负任何与本文件相关的直接或间接的、任何伤害或损失。

请参照交付物中说明文档对紫光展锐交付物进行使用，任何人对紫光展锐交付物的修改、定制化或违反说明文档的指引对紫光展锐交付物进行使用造成的任何损失由其自行承担。紫光展锐交付物中的性能指标、测试结果和参数等，均为在紫光展锐内部研发和测试系统中获得的，仅供参考，若任何人需要对交付物进行商用或量产，需要结合自身的软硬件测试环境进行全面的测试和调试。

Unisoc Confidential For hiar

**紫光展锐（上海）科技有限公司**



# 前言

## 概述

本文档主要介绍了 Android 10.0 平台上，Linux Kernel 里 Audio 模块所涉及的 DTS 文件，旨在帮助开发人员较快的熟悉 DT 的相关配置，并指导其进行 Audio 模块的开发。

## 读者对象




本文档主要适用于 Audio 模块开发人员。

## 缩略语

缩略语	英文全名	中文解释
ALSA	Advanced Linux Sound Architecture	高级 Linux 音频架构
DT	Device Tree	设备树
DTS	Device Tree Source	设备树源码
DTB	Device Tree Blob	设备树块
DTC	Device Tree Compiler	设备树编译器
DMA	Direct Memory Access	直接存储器访问
VBC	Voice Band Controller	语音控制器
MSB	Most Significant Bit	最高有效位
LSB	Least Significant Bit	最低有效位
FE	Front End	前端
BE	Back End	后端
PA	Power Amplifier	功率放大器

## 符号约定

在本文中可能出现下列标志，它所代表的含义如下。

符号	说明
 <b>说明</b>	用于突出重要/关键信息、补充信息和小窍门等。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害。
 <b>注意</b>	用于突出容易出错的操作。 “注意”不是安全警示信息，不涉及人身、设备及环境伤害。
 <b>警告</b>	用于可能无法恢复的失误操作。 “警告”不是危险警示信息，不涉及人身及环境伤害。

## 变更信息

文档版本	发布日期	修改说明
V1.0	2021-02-05	第一次正式发布

## 关键字

Audio、DT。

Unisoc Confidential For hiar

# 目 录

1 概述.....	1
2 Audio DTS 文件架构 .....	2
3 Audio 各模块 DT 详解.....	3
3.1 Codec 级 DT.....	3
3.1.1 自带 Codec DT .....	3
3.1.2 外接 Codec 或 SmartPA DT .....	5
3.2 CPU 级 DT .....	5
3.2.1 VBC DT .....	5
3.2.2 I2S DT .....	6
3.2.3 sprd_fe_dai DT.....	8
3.3 Platform 级 DT.....	9
3.3.1 sprd_pcm platform DT .....	9
3.3.2 sprd_compr platform DT .....	10
3.3.3 sprd_pcm_iis platform DT .....	10
3.3.4 sprd_route_pcm platform DT .....	11
3.4 声卡 DT.....	12
3.4.1 Sprdphone 声卡 DT .....	12
3.4.2 I2S 声卡 DT .....	16
3.5 其他 DT.....	17
3.5.1 Headset DT .....	17
3.5.2 Driver MISC 模块 DT.....	19
4 SOC 级 DT.....	20
5 Kernel 里常用的 DT 解析接口 .....	21

# 表目录

表 3-1 Analog DT 参数 .....	3
表 3-2 Digital DT 参数 .....	4
表 3-3 SOC 文件里 Codec DT 参数 .....	5
表 3-4 VBC DT 参数 .....	6
表 3-5 I2S DT 参数 .....	7
表 3-6 I2S0 总线参数 .....	7
表 3-7 sprd_fe_dai DT 参数 .....	9
表 3-8 sprd_pcm platform DT 参数 .....	10
表 3-9 sprd_compr platform DT 参数 .....	10
表 3-10 sprd_pcm_iis platform DT 参数 .....	11
表 3-11 sprd_route_pcm platform DT 参数 .....	11
表 3-12 Audio 文件里 Sprdphone 声卡 DT 参数 .....	14
表 3-13 I2S 声卡 DT 参数 .....	16
表 3-14 Headset DT Common 部分参数 .....	18
表 3-15 Headset DT SOC 部分参数 .....	19

# 1 概述

DT (Device Tree) 是一种描述硬件的数据结构, 它起源于 OpenFirmware (OF), DTS (Device Tree Source) 即设备树源码。在 Linux 2.6 中, ARM 架构的板级硬件细节过多地被硬编码在 arch/arm/plat-xxx 和 arch/arm/mach-xxx, 比如板上的 platform 设备、resource、i2c\_board\_info、spi\_board\_info 以及各种硬件的 platform\_data, 这些板级细节代码对内核来讲只是垃圾代码。而采用 Device Tree 后, 许多硬件细节可以直接透过它传递给 Linux, 不需要在 kernel 中进行大量的冗余编码。

用户如果要使用 Device Tree, 首先要了解自己的硬件配置和系统运行参数, 并把这些信息组织成 Device Tree source file。通过 DTC (Device Tree Compiler), 将这些适合人类阅读的 Device Tree source file 变成适合机器处理的 Device Tree binary file (也叫 DTB, Device Tree Blob)。

在系统启动的时候, boot program (如 firmware、bootloader) 可以将保存在 flash 中的 DTB 复制到内存, 并把 DTB 的起始地址传递给 client program (如 OS kernel 或者其他特殊功能的程序)。用户也可以通过 bootloader 的交互式命令加载 DTB, 或者 firmware 可以探测到 device 的信息, 组织成 DTB 保存在内存中。

- 对于计算机系统, 处理流程一般是 firmware->bootloader->OS。
- 对于嵌入式系统, 处理流程一般是 bootloader->OS。

Unisoc Confidential For hiar

# 2

## Audio DTS 文件架构

不同 SOC 所使用的 ARM 核不同，对应的 DTS 文件目录有如下两个，但涉及的 DTS 文件基本都相同。

- \bsp\kernel\kernel4.14\arch\arm\boot\dtb
- \bsp\kernel\kernel4.14\arch\arm64\boot\dtb\sprd

以目前 Android 10.0 上 UMS512 为例，DTS 文件路径在 arm64 文件夹下，涉及 Audio 的 DTS 文件如下：

- ums512.dts //SOC 级相关节点
  - sc2730.dtsi //codec 相关节点
  - sharkl5Pro.dtsi //platform、VBC、I2S、audio misc 等相关节点
  - sprd-sound-fe-be.dtsi //headset、sound card 等相关节点
  - extra\_sound\_card\_component.dtsi //Ext SmartPA 的 dai-link 相关节点

ums512.dts 是 SOC 级的 DTS 文件，主要实现一些节点的使能，该文件包含很多 dtsi 类型的文件。

如果同一个 SOC 对应多个不同的 board，则需要添加对应类似的 ums512-1h10-overlay.dtsi 文件加以区分。其他的各个 dtsi 文件，主要是包含 Audio 不同模块的节点信息，下面将分别介绍各个模块节点。

Unisoc Confidential For hiai



# 3

## Audio 各模块 DT 详解

以目前 Android 10.0 上 UMS512 为例。

### 3.1 Codec 级 DT

展锐 SOC 平台一般都内置 Audio Codec，也支持外接 Codec 或 Smart PA 等音频设备。

#### 3.1.1 自带 Codec DT

Audio Codec 相关的 DT 主要由三部分组成：

- sc2730.dtsi 文件里，Codec Analog 部分节点。
- sharkl5Pro.dtsi 文件里，Codec Digital 部分节点。
- ums512.dts 文件里，与 SOC 配置相关的节点信息并关联 Digital 部分节点。

##### 3.1.1.1 Analog DT

展锐平台自带的 Audio Codec 的 Analog 部分在 sc2730 芯片上，因此 DT 节点在 sc2730.dtsi 文件里，如下：

```
sprd_audio_codec_ana: audio-codec@1000 {
compatible = "sprd,sc2730-audio-codec",
"sprd,sc27xx-audio-codec";
#sound-dai-cells = <1>;
reg = <0x1000>;
sprd,syscon-pmu-apb = <&pmu_apb_regs>;
nvmem-cells = <&aud_pabst_vcal>,
<&neg_cp_efuse>,
<&fgu_calib>;
nvmem-cell-names = "aud_pabst_vcal_efuse",
"neg_cp_efuse",
"fgu_4p2_efuse";
status = "disabled";
};
```

表3-1 Analog DT 参数

参数	取值及含义
compatible	驱动代码里匹配 codec 设备的名称

参数	取值及含义
#sound-dai-cells	1 表示需要使用一个参数来指定具体使用的 dai
reg	0x1000, 指 codec 寄存器的偏移地址
sprd,syscon-pmu-apb	指定 pmu apb 的节点
nvmem-cells	指定芯片 efuse 相关的三个节点, 用于读取 nvmem 里面的数值
nvmem-cells-names	efuse 相关的三个节点对应的 name
status	disable 表示默认不使能这个 DT 节点

### 3.1.1.2 Digital DT

Audio Codec 的 Digital 部分放在 AP 侧, 因此它的 DT 节点放在 sharkl5Pro.dtsi 文件里, 信息如下:

```
sprd_audio_codec_dig: audio-codec@33750000 {
compatible = "sprd,audio-codec-dig-agcp";
reg = <0 0x33750000 0 0x1000>;
sprd,syscon-agcp-ahb = <&audcp_ahb_regs>;
};
```

表3-2 Digital DT 参数

参数	取值及含义
compatible	codec digital 设备节点的标识
reg	指定 digital 部分的寄存器起始地址和长度
sprd,syscon-agcp-ahb	指定 agcp ahb 的节点

### 3.1.1.3 SOC 文件里 Codec DT

在 SOC 级的 DTS 文件里 (ums512.dts), 还会有一部分 Audio Codec 相关节点的信息, 主要是与 SOC 间差异相关的配置, 如下:

```
&sprd_audio_codec_ana {
status = "okay";
hp-use-inter-pa = <1>;
fixed-sample-rate = <48000 48000 48000>;
digital-codec = <&sprd_audio_codec_dig>;
};
```

表3-3 SOC 文件里 Codec DT 参数

参数	取值及含义
status	okay 表示 SOC 使能 codec 这个设备节点
hp-use-inter-pa	HP 模式是否使用内部的 PA，目前平台代码没有使用
fixed-sample-rate	指定 codec 的 DA/AD/AD1 使用的固定采样率值
digital-code	关联 codec 数字部分的节点

### 3.1.2 外接 Codec 或 SmartPA DT

如果是外接 SmartPA、Codec 等第三方芯片，还需要添加这些 Codec 芯片的 DT 节点。这部分展锐有专门的文档进行介绍，这里不做阐述。[2 Audio DTS 文件架构](#)提到的文件 `extra_sound_card_component.dtsi` 就是存放外接芯片、声卡 dai-link 的节点信息。

## 3.2 CPU 级 DT

### 3.2.1 VBC DT

VBC 是 AP 连接 Codec 的重要部分，VBC 模块位于 AP 芯片，因此其 DT 节点信息在 `sharkl5Pro.dtsi` 文件里，如下：

```
vbc_v4: vbc@33480000 {
    compatible = "sprd,sharkl5-vbc", "sprd,sharkl5pro-vbc";
    #sound-dai-cells = <1>;
    reg = <0 0x33480000 0 0x1000>;
    sprd,syscon-agcp-ahb = <&audcp_ahb_regs>;
    sprd,vbc-phy-offset = <0x32000000>;
    /* iis pin map */
    pinctrl-names =
    /* iis interface 0 */
    "vbc_iis3_0", "vbc_iism0_0",
    "vbc_iis_to_pad", "vbc_iis_to_aon_usb", "vbc_iism0_1", "vbc_iism0_3",
    "vbc_iis0_3", "vbc_iis1_3", "vbc_iis2_3", "vbc_iis3_3";
    pinctrl-0 = <&vbc_iis3_0>;
    pinctrl-1 = <&vbc_iism0_0>;
    pinctrl-2 = <&vbc_iis_to_pad>;
    pinctrl-3 = <&vbc_iis_to_aon_usb>;
    pinctrl-4 = <&vbc_iism0_1>;
    pinctrl-5 = <&vbc_iism0_3>;
    pinctrl-6 = <&vbc_iis0_3>;
}
```

```
pinctrl-7 = <&vbc_iis1_3>;
pinctrl-8 = <&vbc_iis2_3>;
pinctrl-9 = <&vbc_iis3_3>;
};
```

表3-4 VBC DT 参数

参数	取值及含义
compatible	驱动代码里匹配 VBC 设备的名称
#sound-dai-cells	1 表示 VBC 的 dai 有多个，需要使用一个参数来指定
reg	指定 VBC 部分的寄存器起始地址和长度
sprd,syscon-agcp-ahb	指定 agcp ahb 的节点
sprd,vbc-phy-offset	指定 VBC 物理偏移地址
pinctrl-names/pinctrl	设置 VBC 各种应用场景下对应的 pinmap

### 3.2.2 I2S DT

I2S 相关的 DT 主要由两部分组成：

- sharkl5Pro.dtsi 文件里，I2S 设备部分节点。
- sprd-sound-fe-be.dtsi 文件里，I2S 总线相关配置参数节点。

当直接使用 I2S 作为音频数据的输入/输出时，包含这两部分。

- 需要使用到 I2S 设备的 DT，在文件 sharkl5Pro.dtsi 里，如下：

```
i2s0: i2s@70c00000 {
compatible = "sprd,i2s";
reg = <0 0x70c00000 0 0x1000>;
sprd,dai_name = "i2s_bt_sco0";
sprd,hw_port = <0>;
sprd,syscon-ap-apb = <&ap_apb_regs>;
#sound-dai-cells = <0>;
status = "disable";
clock-names = "clk_iis0",
"clk_twpll_128m",
"clk_twpll_153m6";
clocks = <&ap_clk CLK_AP_IIS0>,
<&pll2 CLK_TWPLL_128M>,
<&pll2 CLK_TWPLL_153M6>;
};
```

表3-5 I2S DT 参数

参数	取值及含义
compatible	驱动代码里匹配 I2S0 设备的名称
reg	指定 I2S0 部分的寄存器起始地址和长度
sprd,dai_name	指定 I2S0 的 dai name
sprd,hw_port	指定 I2S 的端口号，因为是 I2S0，因此 hw_port 为 0
sprd,syscon-ap-apb	指定 ap apb 的节点
#sound-dai-cells	0 表示 I2S0 的 dai 只有 1 个，不需要使用参数来指定
status	disable，表示默认情况下 I2S0 不使能
clock-names/clocks	表示 I2S 需要用到的 clock 节点，有 IIS0/128M/153M6 三个

- 需要设置 I2S0 总线的相关配置参数，在文件 sprd-sound-fe-be.dtsi 里，如下：

```
&i2s0 {
    sprd,config_type = "pcm";
    sprd,slave_timeout = <0xF11>;
    sprd,fs = <8000>;
    sprd,bus_type = <1>;
    sprd,rtx_mode = <3>;
    sprd,byte_per_chan = <1>;
    sprd,slave_mode = <0>;
    sprd,lsb = <0>;
    sprd,lrc = <0>;
    sprd,low_for_left = <1>;
    sprd,clk_inv = <0>;
    sprd,pcm_short_frame = <1>;
    sprd,pcm_slot = <0x1>;
    sprd,pcm_cycle = <1>;
    sprd,tx_watermark = <12>;
    sprd,rx_watermark = <20>;
    status = "disable";
};
```

表3-6 I2S0 总线参数

参数	取值及含义
sprd,config_type	指定 I2S 总线的类型，可以是 PCM 也可以是 I2S
sprd,slave_timeout	指定 Slave 模式下没有采样超时值

参数	取值及含义
sprd,fs	指定同步信号的频率
sprd,bus_type	指定 bus 使用的类型： <ul style="list-style-type: none"> <li>0 为 I2S</li> <li>1 为 PCM</li> </ul>
sprd,rtx_mode	指定 I2S 的 data mode，0/1/2/3 分别表示 idle mode/receive mode/transmit mode/transmit and receive mode
sprd,byte_per_chan	表示每个 channel 传输的比特数，0/1/2 分别为 8/16/32bit
sprd,slave_mode	指定 master/slave mode
sprd,lsb	指定数据传输的格式，0/1 分别表示 MSB/LSB
sprd,lrck	指定同步信号的来源，0/1 分别表示 I2S_SYNC/I2S_LRCK。 <ul style="list-style-type: none"> <li>PCM 方式使用 I2S_SYNC</li> <li>I2S 方式使用 I2S_LRCK</li> </ul>
sprd,low_for_left	表示同步信号 L/R 电平，0/1 分别表示左声道高电平右声道低电平/左声道低电平右声道高电平
sprd,clk_inv	表示采样的 clock 信号，0/1 分别表示 I2S_CLK_N/I2S_CLK_R
sprd,pcm_short_frame	PCM 模式下同步信号的长短，0/1 分别表示 long/short
sprd,pcm_slot	PCM 模式下设置的 slot 值，1 表示使用 slot0
sprd,pcm_cycle	PCM 模式下设置的 cycle 值
sprd,tx_watermark	设置 I2S tx 时 FIFO 的 FULL 和 Empty <ul style="list-style-type: none"> <li>FULL 为 20</li> <li>Empty 为 12</li> </ul>
sprd,rx_watermark	设置 I2S rx 时 FIFO 的 FULL 和 Empty <ul style="list-style-type: none"> <li>FULL 为 20</li> <li>Empty 为 12</li> </ul>
status	disable，表示默认情况下 I2S0 不使能

### 3.2.3 sprd\_fe\_dai DT

在 Sound card 的 FE/BE 架构里，VBC DAI 位于 DAI-Link 的 BE 部分，因此需要一个 FE 部分的 CPU DAI。它主要实现 DMA 的相关配置以及 MCDT 等相关数据通道的配置。相关的 DT 节点信息比较简单，在文件 sharkl5Pro.dtsi 里，如下：

```
sprd_fe_dai: sprd-fe-dai {
compatible = "sprd,fe-dai";
#sound-dai-cells = <1>;
```

```
};
```

表3-7 sprd\_fe\_dai DT 参数

参数	取值及含义
compatible	驱动代码里匹配 sprd_fe_dai 设备的名称
#sound-dai-cells	1 表示 sprd_fe 的 dai 有多个，需要使用一个参数来指定

## 3.3 Platform 级 DT

在展锐平台上，Platform 有两类：

- 一类是面向 DAL-Link 的 FE，主要实现了 DMA 相关的设置以及数据流向的控制。有三个 Platform 设备节点：sprd\_pcm、sprd\_compr 和 sprd\_pcm\_iis。
- 一类用于 DAI-Link 的 BE 部分，实现与 FE 部分的连接。只有一个设备节点 sprd\_route\_pcm。

它们的 DT 节点信息都存在文件 sharkl5Pro.dtsi 里。

### 3.3.1 sprd\_pcm platform DT

通常使用的 Platform 是 sprd\_pcm 节点，其 DT 信息如下：

```
sprd_pcm: sprd-pcm-audio {
    compatible = "sprd,sharkl5-pcm-platform",
        "sprd,sharkl5pro-pcm-platform";
    #sound-dai-cells = <0>;

    /* agcp dma_ap */
    dmas = <&agcp_dma 1 &agcp_dma 2
        &agcp_dma 3 &agcp_dma 4
        &agcp_dma 5 &agcp_dma 6
        &agcp_dma 7 &agcp_dma 8
        &agcp_dma 9 &agcp_dma 10
        &agcp_dma 11 &agcp_dma 12
        &agcp_dma 13 &agcp_dma 14
        &agcp_dma 15 &agcp_dma 16
        &agcp_dma 9 &agcp_dma 9>;
    dma-names =
        "normal_p_l", "normal_p_r",
        "normal_c_l", "normal_c_r",
        "normal23_p_l", "normal23_p_r",
        "normal23_c_l", "normal23_c_r",
        "dspcap_c", "a2dppcm_p",
```

```
"voice_c", "fast_p",
"loop_c", "loop_p",
"voip_c", "voip_p",
/* dspfmcap_c and dspbtscocap_c same as dspcap_c */
"dspfmcap_c", "dspbtscocap_c";
};
```

表3-8 sprd\_pcm platform DT 参数

参数	取值及含义
compatible	驱动代码里匹配 platform sprd_pcm 设备的名称
#sound-dai-cells	0 表示该 Platform 的 dai 只有 1 个，不需要使用参数来指定
dmass/names	表示 PCM device 使用 DMA 对应的 channel ID

### 3.3.2 sprd\_compr platform DT

sprd\_compr platform 节点信息与 sprd\_pcm 的基本相同，该节点当前主要用于 offload playback 模式，DT 信息如下：

```
sprd_compr: sprd-compr-audio {
compatible = "sprd,sharkl5-compress-platform",
"sprd,sharkl5pro-compress-platform";
#sound-dai-cells = <0>;
dmass = <&agcp_dma 17 &agcp_dma 18>;
dma-names = "compr_dma_stage0", "compr_dma_stage1";
};
```

表3-9 sprd\_compr platform DT 参数

参数	取值及含义
compatible	驱动代码里匹配 platform sprd_compr 设备的名称
#sound-dai-cells	0 表示该 Platform 的 dai 只有 1 个，不需要使用参数来指定
dmass/names	表示 PCM device 使用 DMA 对应的 channel ID

### 3.3.3 sprd\_pcm\_iis platform DT

I2S 模式下所使用的 platform 节点，DT 信息如下：

```
sprd_pcm_iis: sprd-pcm-iis {
compatible = "sprd,sharkl5-pcm-platform";
#sound-dai-cells = <0>;
```



```

dmas = <&ap_dma 5 &ap_dma 6>;
dma-names = "iis0_tx", "iis0_rx";
sprd,dma-2stage-usecase = <2>;
sprd,node-count-2stage-level-1 = <1>;
sprd,syscon-pmu-apb = <&pmu_apb_regs>;
sprd,dma-2stage-level-1-int-source = <1>;
};

```

表3-10 sprd\_pcm\_iis platform DT 参数

参数	取值及含义
compatible	驱动代码里匹配 platform sprd_pcm_iis 设备的名称
#sound-dai-cells	0 表示该 Platform 的 dai 只有 1 个，不需要使用参数来指定
dmas/names	表示 PCM device 使用 DMA 对应的 channel ID
sprd,dma-2stage-usecase	表示是否使用两级 DMA，1/2/4 分别表示 Normal/Deep/Capture 三种模式下使用两级 DMA 处理
sprd,node-count-2stage-level-1	表示是否使用 PingPang buf <ul style="list-style-type: none"> <li>1 表示不使用</li> <li>2 表示使用</li> </ul>
sprd,syscon-pmu-apb	指定 pmu apb 的节点
sprd,dma-2stage-level-1-int-source	设定两级 DMA 使用时 Interrupt 的来源 <ul style="list-style-type: none"> <li>1 表示来自于 AMR</li> <li>2 表示中断源来源于 AP</li> </ul>

### 3.3.4 sprd\_route\_pcm platform DT

在 Sound Card 的 FE/BE 架构里，FE 的 Platform 提供对外的 PCM 接口，主要实现数据的 DMA 传输。BE 的 Platform 部分，则需要实现与 FE 的 Routing 连接。其节点信息如下：

```

sprd_route_pcm: sprd-routing-pcm {
compatible = "sprd,pcm-routing";
#sound-dai-cells = <0>;
};

```

表3-11 sprd\_route\_pcm platform DT 参数

参数	取值及含义
compatible	驱动代码里匹配 sprd_route_pcm platform 设备的名称
#sound-dai-cells	0 表示该 Platform 的 dai 只有一个，无需使用参数来指定

## 3.4 声卡 DT

UMS512 平台有两个声卡，Sprdphone card 和 I2S card，大部分的 Audio 应用需要使用 Sprdphone 声卡，只有 BT 在特定模式下，才会使用 I2S 声卡。

### 3.4.1 Sprdphone 声卡 DT

该声卡的主要节点信息在 Audio 文件 sprd-sound-fe-be.dtsi 里，与 SOC 相关的差异类的节点信息放在 ums512.dts 文件里。

该声卡 DT 节点主要的部分是在进行 sound-dai 的设置。因为使用了 FE/BE 的架构，因此，

- DAI-Link 的 FE 节点使用情况如下：
  - platform dai: sprd\_pcm/sprd\_compr
  - cpu dai: sprd\_fe\_dai
  - codec dai: none
- DAI-Link 的 BE 节点使用情况如下：
  - platform dai: sprd\_route\_pcm
  - cpu dai: vbc
  - codec dai: sprd\_audio\_codec/外部 codec/外部 smartpa

#### 3.4.1.1 Audio 文件里 Sprdphone 声卡 DT

Sprdphone 声卡是展锐平台主要使用的声卡，涉及的 DAI-Link 比较多，而且使用了 FE/BE 的连接方式，其 DT 节点信息如下：

```
sound_vbc_v4_sprd_codec: sound@0 {
compatible = "sprd,vbc-v4-codec-sc2730";
status = "disabled";
sprd-audio-card,name = "sprdphone-sc2730";
sprd-audio-card,widgets =
"Headphone", "inter HP PA",
"Speaker", "inter Spk PA",
"Speaker", "inter Ear PA";
sprd-audio-card,routing =
"HPMIC Pin", "HP Mic Jack",
"MIC Pin", "Mic Jack",
"MIC2 Pin", "Aux Mic Jack",
"inter HP PA", "HP Pin",
"inter Spk PA", "SPK Pin",
"inter Ear PA", "EAR Pin";
sprd-audio-card,headset = <&sprd_headset>;
sprd,syscon-agcp-ahb = <&audcp_ahb_regs>;

/* fe dai-links */
```

```
/* fe normal ap01 */
sprd-audio-card,dai-link@0 {
link-name = "FE_NORMAL_AP01";
stream-name = "FE_ST_NORMAL_AP01";
ignore-suspend = <0>;
dynamic = <1>;
trigger = <1 1>;
dpcm-playback = <1>;
dpcm-capture = <1>;
be-id = <0>;
plat {
sound-dai = <&sprd_pcm>;
};
cpu {
sound-dai = <&sprd_fe_dai 0>;
};
codec {
sound-dai = <0 0>;
};
}

... ..

/* be dai-links */
/* 0: be normal_ap01 codec */
sprd-audio-card,dai-link@18 {
ignore-suspend = <0>;
link-name = "BE_NORMAL_AP01_CODEC";
stream-name = "BE_ST_NORMAL_AP01_CODEC";
no-pcm = <1>;
dpcm-playback = <1>;
dpcm-capture = <1>;
be-id = <0>;
plat {
sound-dai = <&sprd_route_pcm>;
};
cpu {
sound-dai = <&vbc_v4 0>;
};
codec {
sound-dai = <&sprd_audio_codec_ana 0>;
```

```
};
};
... ..
}
```

表3-12 Audio 文件里 Sprdphone 声卡 DT 参数

参数	取值及含义
compatible	驱动代码里匹配 Sprdphone sound card 设备的名称
status	disable 表示默认情况下该声卡不使能
sprd-audio-card,name	表示该声卡的名称
sprd-audio-card,widgets	表示 output 的几个 widget
sprd-audio-card,routing	表示输入/输出的 path routing, sink ← source
sprd-audio-card,headset	表示耳机设备的节点
sprd,syscon-agcp-ahb	用于指定 agcp ahb 的节点
sprd-audio-card,dai-link@n	表示 dai-link 节点, 有 FE/BE 之分
link-name	节点名称
stream-name	对应的 stream 名称
dpcm-playback	该 stream 是否包含 pcm playback device
dpcm-capture	该 stream 是否包含 pcm capture device
ignore-suspend	<ul style="list-style-type: none"> <li>为 0 表示 suspend 时 dai-link 需要下电</li> <li>为 1 表示保持上电状态</li> </ul>
be-id	目前所有的 dai-link 都设置为 0
no-pcm	用于区分 FE/BE, BE 需设置为 1, 表示没有对外的 PCM 接口
dynamic	FE 需要设置为 1, 表示需要动态加载到某个 BE 进行连接
trigger	只用于 FE, 表示 playback/capture 两种 stream, FE/BE 在 trigger 时的顺序, <ul style="list-style-type: none"> <li>0 表示先 FE 再 BE</li> <li>1 表示先 BE 再 FE</li> <li>2 表示共用一个 bespoke trigger</li> </ul>
plat/sound-dai	指定 platform 所使用的 dai 节点
cpu/sound-dai	指定 cpu 所使用的 dai 节点, 具体的 dai 可通过参数指定
codec/sound-dai	指定 codec 所使用的 dai 节点, 具体的 dai 可通过参数指定

### 3.4.1.2 SOC 文件里 Sprdphone 声卡 DT

SOC 文件里该声卡的节点信息，主要是包含了各个平台间差异的部分，公共部分都是使用 Audio 文件里的节点信息。节点信息里的字段基本都相同，这里不再解释。只是 SOC 文件里 Sprdphone 声卡的 widget/routing 信息会覆盖 Audio 文件里的 Sprdphone 声卡 DT 信息，并且 SOC 文件里 Sprdphone 声卡会增加一些平台特有的 DAI-Link 节点信息。

UMS512 项目就增加了使用 UCP1301 设备的相关节点信息，如下：

```
&sound_vbc_v4_sprd_codec {
status = "okay";
sprd-audio-card,routing =
"HPMIC Pin", "HP Mic Jack",
"MIC Pin", "Mic Jack",
"MIC2 Pin", "Aux Mic Jack",
"inter HP PA", "HP Pin",
"inter Spk PA", "SPK Pin",
"UCP1301 SPK ON", "SPK Pin",
"inter Spk PA", "UCP1301 SPK",
"DMIC Pin", "DMic Jack",
"DMIC1 Pin", "DMic1 Jack",
"inter Ear PA", "EAR Pin";

/* 35: ucp1301 normal ap01 */
sprd-audio-card,dai-link@53 {
ignore-suspend = <1>;
link-name = "BE_NORMAL_AP01_UCPSPK";
stream-name = "BE_ST_NORMAL_AP01_UCPSPK";
no-pcm = <1>;
dpcm-playback = <1>;
be-id = <0>;
dai-format = "i2s";
plat {
sound-dai = <&sprd_route_pcm>;
};
cpu {
sound-dai = <&vbc_v4 3>;
};
codec {
sound-dai = <&ucp1301_spk 0>;
};
};
```

```

/*
 * node 54 and node 55 (index 36 and 37) are reserved for
 * "ucp1301-spkr2" and "ucp1301-rcv"
 * nodes from 56 to 67 (index from 38 to 49) are used by HIFI
 */
};

```

### 3.4.2 I2S 声卡 DT

I2S 声卡的 DT 节点信息相对比较简单，基本都在 Sprdphone 声卡 DT 节点里使用过。对于 Dai-link 项，Sprdphone 声卡节点里使用了 FE/BE 的方式，I2S 声卡却没有使用。如下：

```

sound_sprd_ap_alliis: sound@1 {
compatible = "sprd,i2s-null-codec";
status = "disabled";
sprd-audio-card,name = "all-i2s";
/* dai-links */
sprd-audio-card,dai-link@0 {
plat {
sound-dai = <&sprd_pcm_iis>;
};
cpu {
sound-dai = <&i2s0>;
};
codec {
sound-dai = <0 0>;
};
};
}

```

表3-13 I2S 声卡 DT 参数

参数	取值及含义
compatible	驱动代码里匹配 I2S sound card 设备的名称
status	disable 表示默认情况下该声卡不使能
sprd-audio-card,name	表示该声卡的名称
sprd-audio-card,dai-link@0	表示 dai-link 节点，该声卡只有一个
plat/sound-dai	指定 platform 使用 sprd_pcm_iis 节点里的 dai
cpu/sound-dai	指定 cpu 使用 i2s0 节点里的 dai
codec/sound-dai	指定 codec 不使用 dai

## 3.5 其他 DT

### 3.5.1 Headset DT

展锐平台上耳机设备是 Audio Codec 的一部分，DT 设备节点有两部分，common 部分在 sprd-sound-fe-be.dts Audio 文件里，另一部分跟 SOC 差异相关的配置，在 ums512.dts 文件里。

- Common 部分 DT 信息如下：

```
sprd_headset: sprd-headset {
compatible = "sprd,headset";
status = "disabled";
/* 0: normal-open jack; 1: normal-close jack */
sprd,jack-type = <0>;
io-channels = <&pmic_adc 20>;
io-channel-names = "headmic_in_little";
gpios = <&pmic_eic 6 GPIO_ACTIVE_HIGH>;
/*
* 10ms debounce time is recommended from
* asic from headset button
*/
sprd,debounce-interval = <10>;
gpio-names = "aud_int_all";
sprd,3pole-adc-threshold = <2950>;
sprd,adc-gnd = <328>;
sprd,half-adc-gnd = <100>;
sprd,coefficient = <884520>;
sprd,irq-threshold-button = <4>;
nvmem-cells = <&headset_adc_fir>, <&headset_adc_sec>;
nvmem-cell-names = "hp_adc_fir_calib", "hp_adc_sec_calib";

sprd,headset-button@0 {
/* media */
adc-min = <0>;
adc-max = <480>;
code = <226>;
};
sprd,headset-button@1 {
/* volume up */
adc-min = <481>;
adc-max = <990>;
code = <115>;
};
```

```
sprd,headset-button@2 {
/* volume down */
adc-min = <991>;
adc-max = <2500>;
code = <114>;
};
};
```

表3-14 Headset DT Common 部分参数

参数	取值及含义
compatible	驱动代码里匹配 sprd_headset 设备的名称
status	disable, 表示默认情况下该耳机设备不使能
sprd,jack-type	表示 jack 的类型 <ul style="list-style-type: none"> <li>0: normal-open jack</li> <li>1: normal-close jack</li> </ul>
io-channels/names	对应耳机 ADC channel 的节点信息
gpios/names	耳机 Detect 的 GPIO 节点信息
sprd,debounce-interval	设置 debounce time 为 10ms
sprd,3pole-adc-threshold	用于判断为 3pole 耳机类型的一个阈值
sprd,adc-gnd	用于判断为不同耳机类型的一个电压值
sprd,half-adc-gnd	用于判断为不同耳机类型的另一个电压值
sprd,coefficient	一个用于获取 real 电压的换算系数
sprd,irq-threshold-button	设置 button IRQ 的门槛值, 对应寄存器 ANA_HDT0 相应比特位的值
nvmem-cells/names	指定校准相关的两个节点, 用于读取 nvmem 里面的数值
sprd,headset-button	三个按键节点, 分别是 media/volume up/volume down
adc-min/adc-max	ADC 的取值范围, 用于相应按键的检测
code	对应三个按键上报时的键值

- SOC 文件里, 主要是增加了 TypeC 耳机节点的支持, 相关的 DT 信息如下:

```
&sprd_headset {
    status = "okay";
    extcon = <&pmic_typec>;
    mic-gpios = <&ap_gpio 16 GPIO_ACTIVE_HIGH>;
    lr-gpios = <&ap_gpio 15 GPIO_ACTIVE_HIGH>;
    sprd,3pole-adc-threshold = <112>;
```



```
};
```

表3-15 Headset DT SOC 部分参数

参数	取值及含义
status	okay 表示 SOC 使能 headset 这个设备节点
extcon	type C 耳机的支持节点信息
mic-gpios	type C 耳机 mic 对应的 GPIO 为 16
lr-gpios	type C 耳机 output 对应的 GPIO 为 15
sprd,3pole-adc-threshold	用于判断耳机类型的一个电压值

### 3.5.2 Driver MISC 模块 DT

这部分主要包含 Audio 驱动的 SIPC、MCDT、AUDMEM、AGCP、PIPE、AUDIO DUMP、AUDIOCP DVFS 等模块的 DT，相关节点信息也放在 sharkl5Pro.dtsi 文件里。因为这些模块基本是展锐平台固有的驱动模块，与芯片相关，客户需求涉及的改动较少，这里不做详细介绍。

Unisoc Confidential For hiar

# 4

## SOC 级 DT

SOC 级 DT 是指类似 ums512.dts 文件，会包含很多 dtsi 文件。该文件主要实现的是一些设备节点的使能，另外包含一些节点的 SOC 级配置，比如上文介绍的 Audio Codec、Sprdphone sound card 以及 Headset。

使能的节点如下：

- Codec 级的 sprd\_audio\_codec\_ana。
- CPU 级的 i2s0。
- 声卡级的 sound\_vbc\_v4\_sprd\_codec 和 sound\_sprd\_ap\_alliis。
- 其他的有 sprd\_headset。

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

```
&i2s0 {  
    status = "okay";  
};
```

```
&sound_vbc_v4_sprd_codec {  
    status = "okay";  
};
```

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

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

# 5

## Kernel 里常用的 DT 解析接口

- 对 register 的解析

```
platform_device *pdev;
struct resource *res;
res = platform_get_resource(pdev, IORESOURCE_MEM, 0);
i2s->memphys = (unsigned int *)res->start;
i2s->reg_size = (unsigned int)resource_size(res);
i2s->membase = devm_ioremap_resource(&pdev->dev, res);
```

- 对 pin map 的解析

```
platform_device *pdev;
struct pinctrl *pctrl;
pctrl = devm_pinctrl_get(&pdev->dev);
state = pinctrl_lookup_state(pctrl, buf);
pinctrl_select_state(pctrl, state);
```

- 对 nvmem 的解析

```
platform_device *pdev;
struct nvmem_cell *cell;
cell = nvmem_cell_get(&pdev->dev, cell_name);
buf = nvmem_cell_read(cell, &len);
nvmem_cell_put(cell);
```

- 对 clock 的解析

```
struct i2s_priv *i2s;
struct clk *clk_parent;
clk_parent = devm_clk_get(i2s->dev, "clk_twpll_128m");
clk_set_parent(i2s->i2s_clk, clk_parent);
source_clk = clk_get_rate(clk_parent);
clk_set_rate(i2s->i2s_clk, source_clk);
clk_put(clk_parent);
```

- 对 DMA channel 的使用

```
platform_device *pdev;
struct dma_chan *dma_chn_request;
dma_chn_request = dma_request_slave_channel(pdev->dev, used_dma_channel_name[i]);
```