



D1-H Linux DMIC 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 DMIC 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	2
2.3.1 Device Tree 配置介绍	2
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	10
4.1 模块声卡/设备查看	10
4.2 模块音频控件及通路配置说明	10
4.2.1 模块音频控件说明	10
4.3 模块音频通路配置说明	11
4.4 模块功能验证说明	11
4.4.1 DMIC 8 通道录音	11
5 FAQ	13

插图

3-1 Device Driver	7
3-2 Sound	7
3-3 Advanced	8
3-4 ALSA	8
3-5 Allwinner	9
3-6 module	9
4-1 DMIC_CAP_WAV	12



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 DMIC 接口的具体使用方法，能够更快地基于 AW SUNXI 平台完成对 DMIC 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

- DMIC：数字麦克风阵列，用作语音采集输入；
- 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 DMIC 接口模块的基础介绍

2.1 模块功能规格介绍

AW SUNXI 平台 DMIC 接口模块功能及规格:

- 只支持 record 录音功能;
- 支持多种采样率格式 (8KHz, 11.025KHz, 16KHz, 22.05KHz, 24KHz, 32KHz, 44.1KHz, 48KHz);
- 最高可支持至 8 通道;
- 支持 64 OSR (过采样率) 以及 128 OSR;
- 支持 16bit/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:

```
/tina/lichee/linux-5.4/sound/soc/  
├── sunxi                               // Sunxi平台  
│   ├── sunxi-dmic.c                   // Sunxi平台DMIC接口代码  
│   ├── sunxi-dmic.h                   // Sunxi平台DMIC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  
├── codecs                             // 内核解码器存放路径  
└── dmic.c                             // DMIC解码器驱动
```

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 DMIC 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)

```
dmic:dmic@2031000{
    #sound-dai-cells = <0>;
    compatible = "allwinner,sunxi-dmic";
    reg = <0x0 0x02031000 0x0 0x50>;
    clocks = <&ccu CLK_PLL_AUDI00>,
            <&ccu CLK_DMIC>,
            <&ccu CLK_BUS_DMIC>;
    clock-names = "pll_audio", "dmic", "dmic_bus";
    resets = <&ccu RST_BUS_DMIC>;
    dmas = <&dma 8>;
    dma-names = "rx";
    interrupts-extended = <&plic0 40 IRQ_TYPE_LEVEL_HIGH>;
    clk_parent = <0x1>;
    capture_cma = <256>;
    data_vol = <0xB0>;
    dmic_rxsync_en = <0x0>;
    rx_chmap = <0x76543210>;
    device_type = "dmic";
    status = "disabled";
};

dmic_codec:sound@2031050{
    #sound-dai-cells = <0>;
    compatible = "dmic-codec";
    reg = <0x0 0x02031050 0x0 0x4>;
    num-channels = <8>;
    status = "disabled";
};

sounddmic:sounddmic@2031060 {
    reg = <0x0 0x02031060 0x0 0x4>;
    compatible = "sunxi,simple-audio-card";
    simple-audio-card,name = "snddmic";
    simple-audio-card,capture_only;
    status = "disabled";
    /* simple-audio-card,format = "i2s"; */
    simple-audio-card,cpu {
        sound-dai = <&dmic>;
    };
    simple-audio-card,codec {
        sound-dai = <&dmic_codec>;
    };
};
```

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)
```

```
dmic_pins_a: dmich@0 {
    /* DMIC_PIN: CLK, DATA0, DATA1, DATA2 */
    pins = "PE17", "PB11", "PB10", "PD17";
    function = "dmich";
    drive-strength = <20>;
    bias-disable;
};

dmich_pins_b: dmich@1 {
    pins = "PE17", "PB11", "PB10", "PD17";
    function = "io_disabled";
    drive-strength = <20>;
    bias-disable;
};

&dmich {
    pinctrl-names = "default","sleep";
    pinctrl-0 = <&dmich_pins_a>;
    pinctrl-1 = <&dmich_pins_b>;
    status = "okay";
};

&dmich_codec {
    status = "okay";
};

&sounddmich {
    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 `dmic`、`dmic_codec`、`snddmic` 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：

```
&dmic {  
    ...  
    status = "okay";  
};  
  
&dmic_codec {  
    status = "okay";  
};  
  
&sounddmic {  
    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：

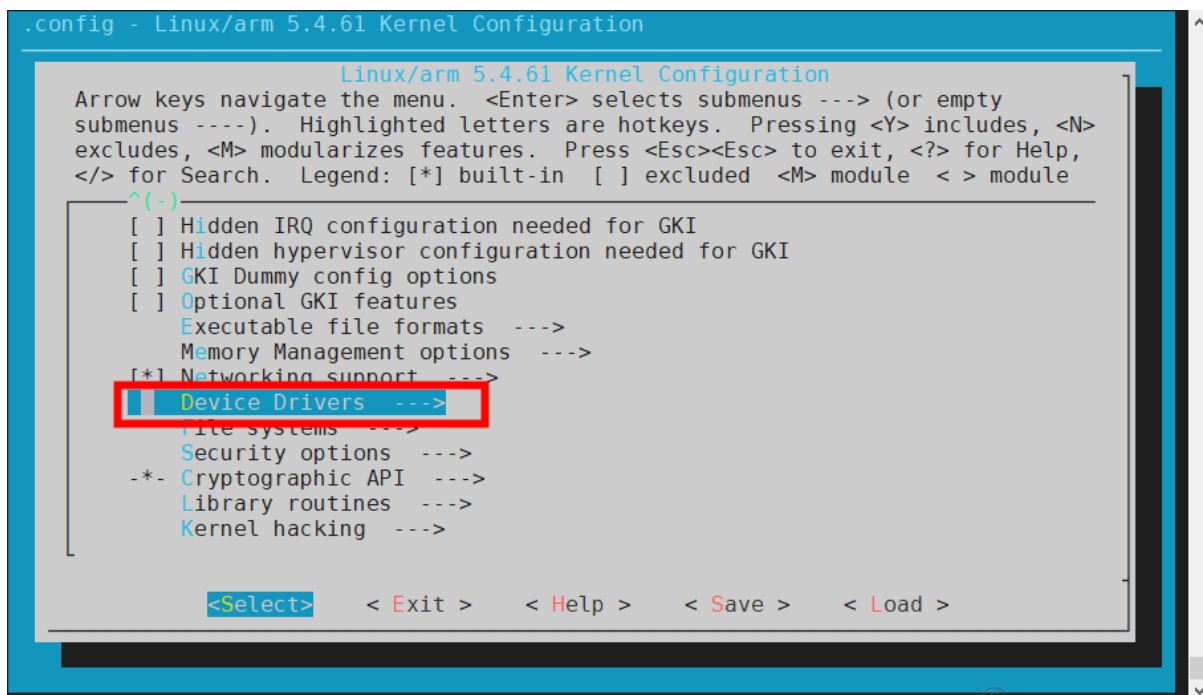


图 3-1: Device Driver

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

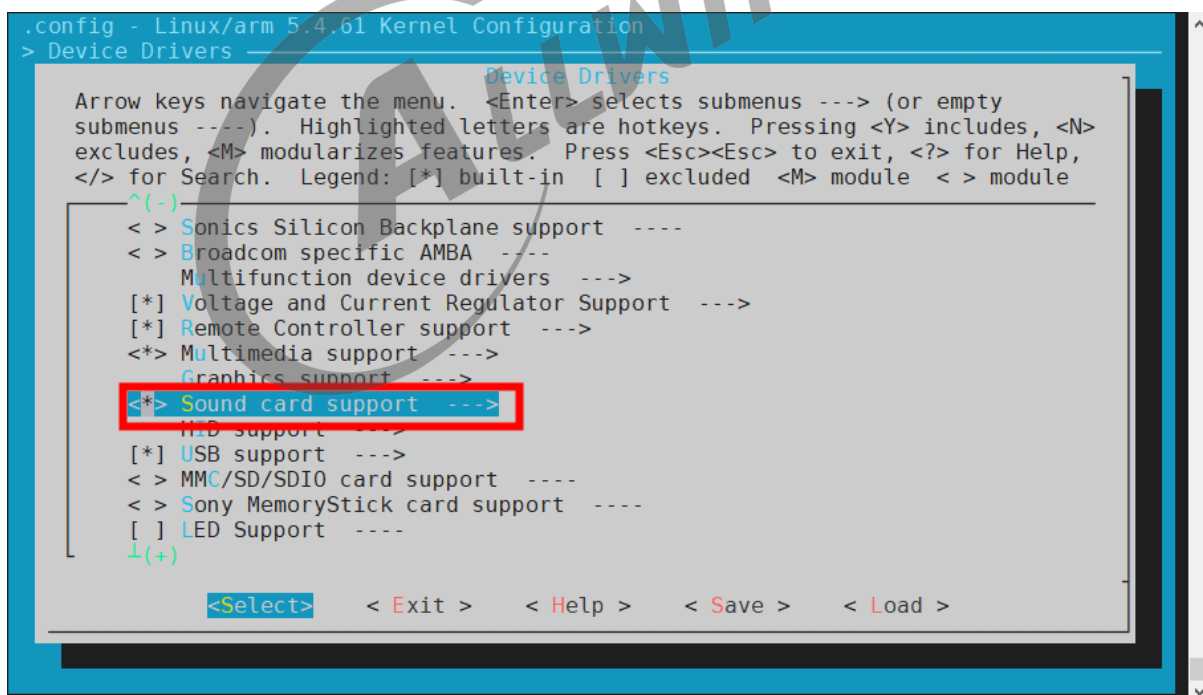


图 3-2: Sound

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

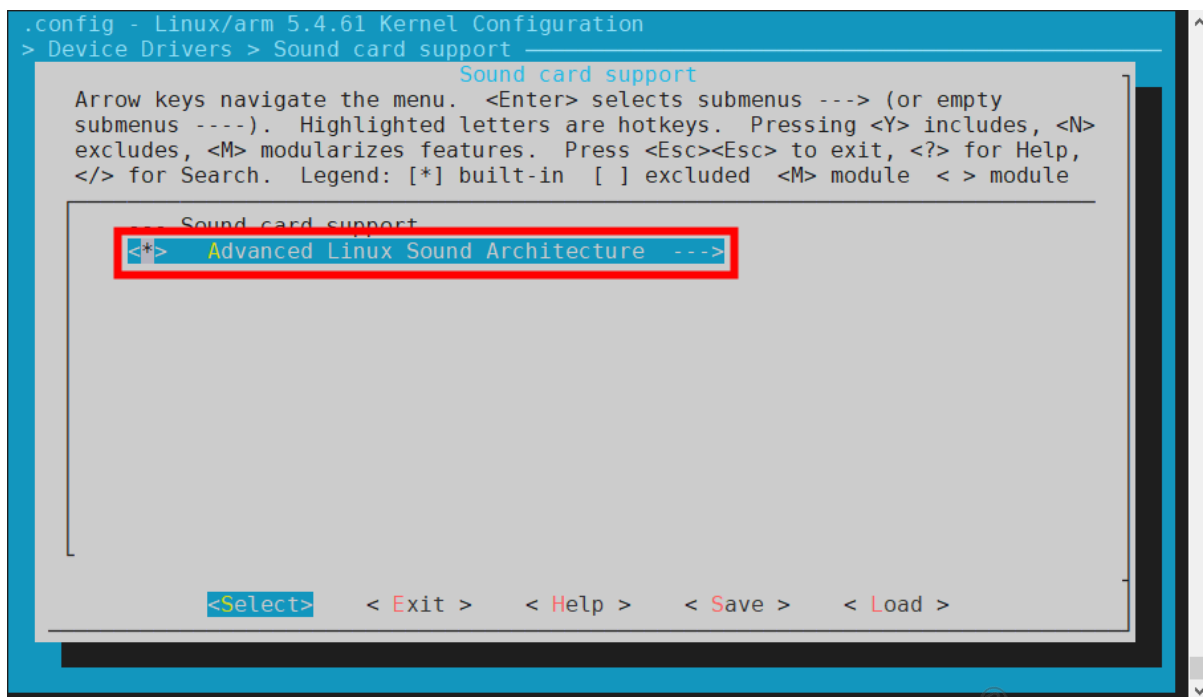


图 3-3: Advanced

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

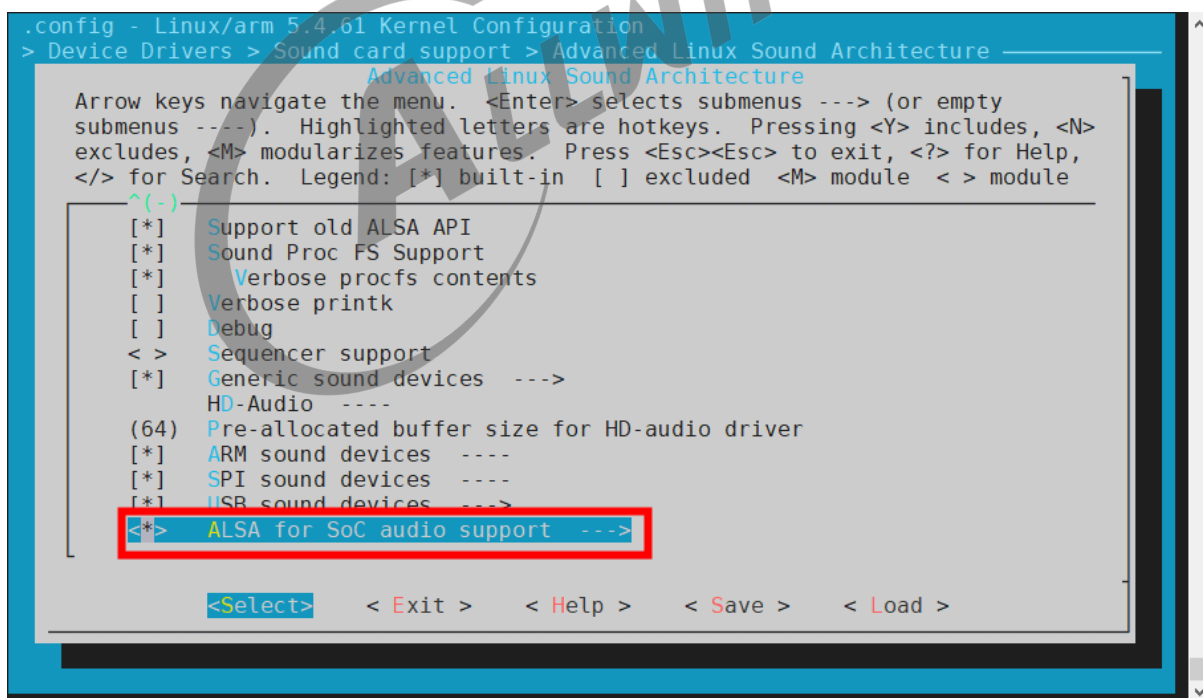


图 3-4: ALSA

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

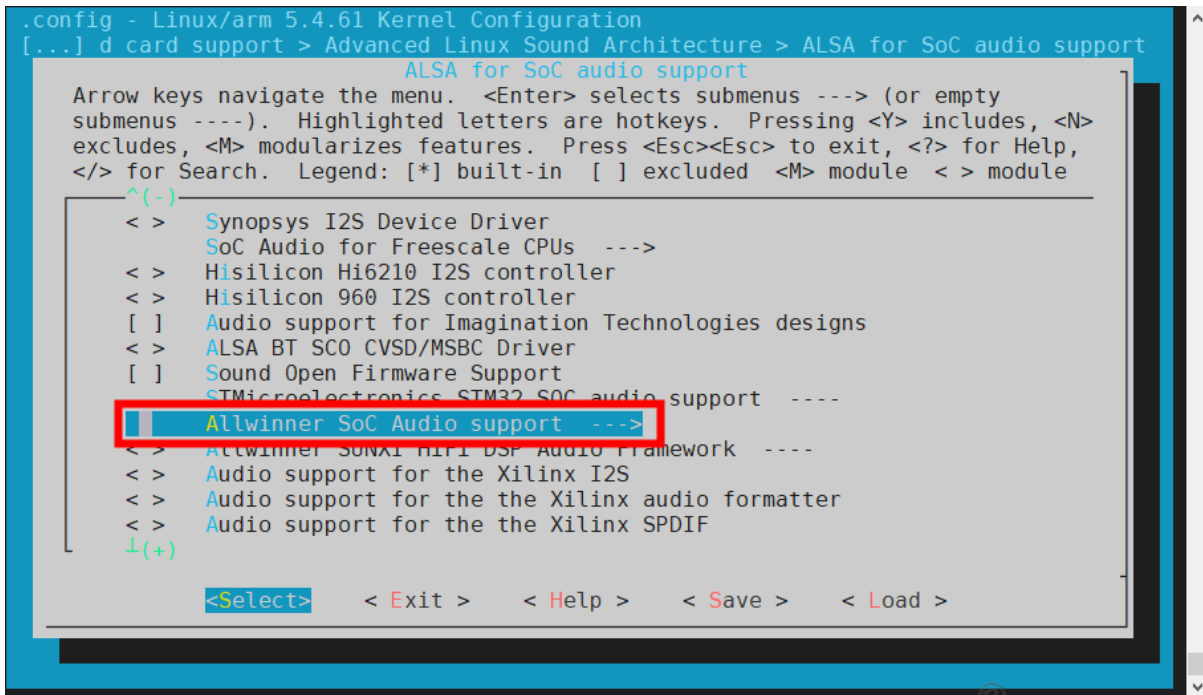


图 3-5: Allwinner

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：

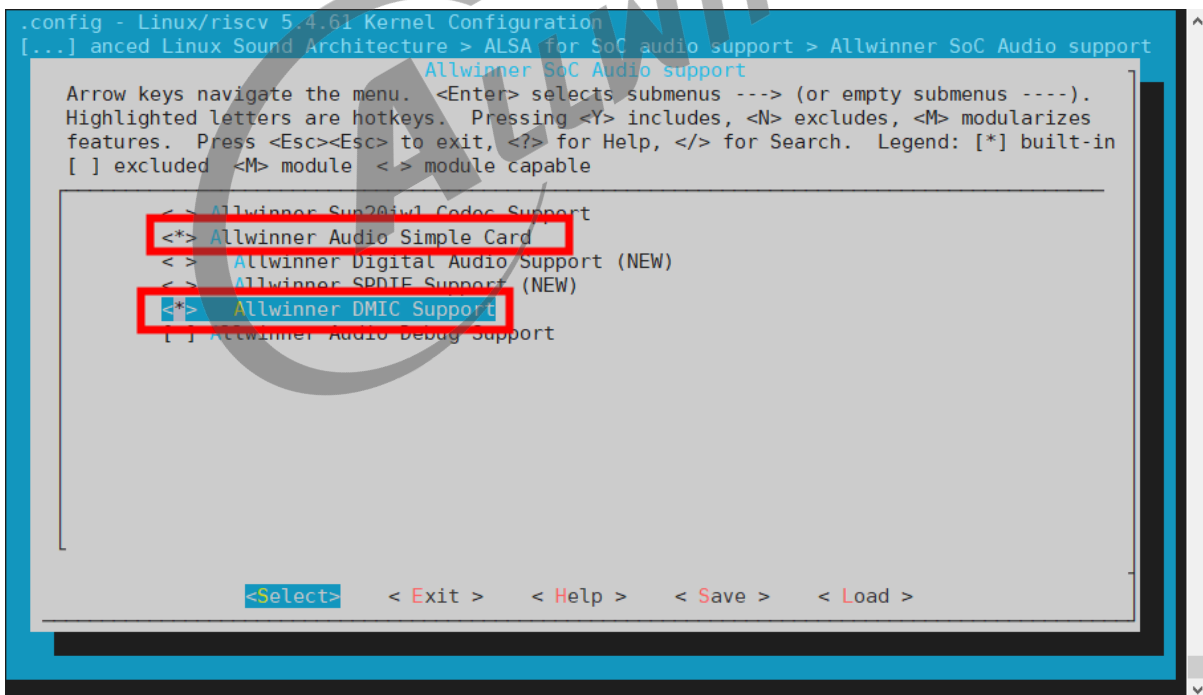


图 3-6: module

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 dm1c 声卡及设备，具体查看及确认操作示例如下所示：

```
/ # cat /proc/asound/cards
0 [snddm1c      ]: snddm1c - snddm1c           // dm1c模块声卡
                  snddm1c

/ #
/ # ls -l /proc/asound/snddm1c/
total 0
-r--r--r--    1 root    root          0 Jan  1 00:48 id      // dm1c模块声卡ID名称
dr-xr-xr-x    3 root    root          0 Jan  1 00:48 pcm0c  // dm1c模块声卡录音设备
/ #
```

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 dm1c 模块声卡音频控件列表及音频路由：

```
/ #
0 [snddm1c      ]: snddm1c - snddm1c
                  snddm1c

/ #
/ # tinymix -D 0           // 查看默认声卡序号的声卡列表状态
Mixer name: 'snddm1c'
Number of controls: 2
ctl      type    num      name                               value
/ #
```

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：（当前 DM1C 声卡无任何音频控件，故无相应控件说明）

音频控件序号	音频控件名称	配置可选值	音频控件说明
/	/	/	/

4.3 模块音频通路配置说明

由于 DMIC 模块属于纯直通型输入通道，且音频驱动并未透出任何通路控件可操作，所以无需做任何的音频通路配置，直接对音频声卡及其设备录音操作即可。

4.4 模块功能验证说明

4.4.1 DMIC 8 通道录音

具体功能验证操作演示如下：

```
/ # cat proc/asound/cards
0 [snddmic      ]: snddmic - snddmic           // dmic声卡序号为0
                                snddmic
/ #
/ # tinycap /tmp/test_dmic_8ch.wav -D 0 -d 0 -c 8 -b 16 -r 48000           // 用声卡序号为0的声卡
    即DMIC声卡进行8声道、16bit、48KHz的音频保存为/tmp/test_dmic_8ch.wav
Capturing sample: 8 ch, 48000 hz, 16 bit                                // 正在录音中

^CCaptured 585728 frames                                               // Ctrl + C退出录音程
    序，录音完成
/ #
```

最后通过 adb 工具（adb pull）等方式将刚录制的 WAV 音频文件拉出来通过音频解析软件（Audition/ocenaudio）播放查看、确认即可，具体正常八通道录音音频文件解析示例如下图所示：

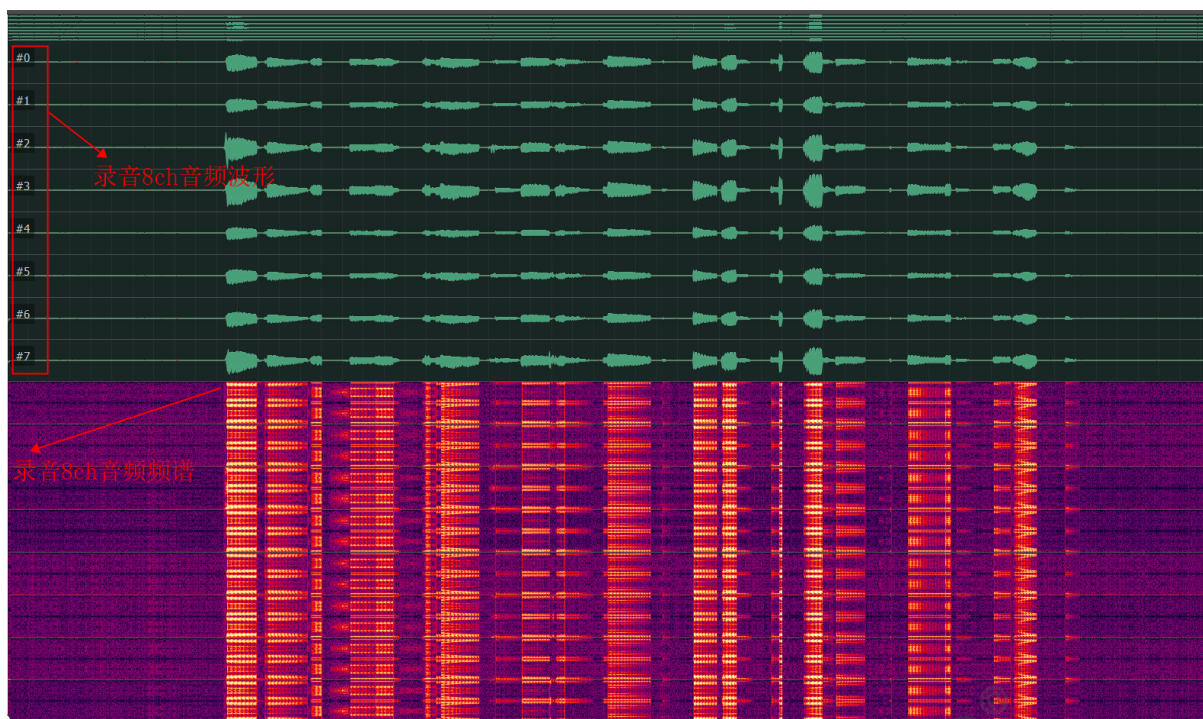


图 4-1: DMIC_CAP_WAV

5 FAQ

- 按要求进行 menuconfig 配置，并且在 board.dts 将 dmic 打开，但无声卡生成。
- 查看 dmic 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。