

确认 dts 中 I2C 配置是否正确

检查 dts 中 i2c 的 mode 是否配置正确。具体可见 i2c 节点中 i2c-padmux 项的配置，此项需与硬件设计相对应。

```
i2c1@1{
    compatible = "sstar,i2c";
    reg = <0x1F223200 0x200>, <0x1F203c00 0x200>, <0x1F207000 0x200>;
    #address-cells = <1>;
    #size-cells = <0>;
    clocks = <&CLK_miic1>;
    i2c-group = <1>;
    /*
     * padmux: 1 -> PAD_GPI02, PAD_GPI03
     *          2 -> PAD_HDMITX_SCL, PAD_HDMITX_SDA
     *          4 -> PAD_TTL22, PAD_TTL23
     *          5 -> PAD_SD_CLK, PAD_SD_CMD
     */
    i2c-padmux = <2>;
    /*
     * speed: 0 -> HWI2C_HIGH(high speed: 400 KHz)
     *         1 -> HWI2C_NORMAL(normal speed: 300 KHz)
     *         2 -> HWI2C_SLOW(slow speed: 200 KHz)
     *         3 -> HWI2C_VSLOW(very slow: 100 KHz)
     *         4 -> HWI2C_USLOW(ultra slow: 50 KHz)
     *         5 -> HWI2C_UVSLOW(ultra-very slow: 25 KHz)
     */
    i2c-speed = <3>;
    i2c-en-dma = <0>; // 0: disable; 1: enable;
    status = "ok";
}
```

确认 padmux dts 中是否有将 i2c 引脚切对

检查 padmux dtsi 中需要切换成 i2c 的 pin 脚是否有填入，且确保 mode 及复用功能填写正确。如图，此项需与硬件设计相对应。

```
<PAD_HDMITX_SCL PINMUX_FOR_I2C1_MODE_2 MDRV_PUSE_I2C1_SCL >,
<PAD_HDMITX_SDA PINMUX_FOR_I2C1_MODE_2 MDRV_PUSE_I2C1_SDA >,
```

通过 i2c-tools 工具确认 i2c 工作是否正常

可通过开源工具 i2c-tools 中的 i2cget 来读取顺芯 codec 的 chip id，从而确认 i2c 工作是否正常。

7243:

7243 的 i2c 地址与接线有关，可与硬件确认

Table 1 Write Data to Register in I²C Interface Mode

	Chip Address	R/W		Register Address		Data to be written		
start	0010 0 ~AD1 ~AD0	0	ACK	RAM	ACK	DATA	ACK	Stop

REGISTER 0X0E – CHIP ID, DEFAULT 0100 0000

Bit Name	Bit	Description
CHIP_ID	7:0	Chip ID[7:5]: 000 – 93xx 001 – 71xx 010 – 72xx 011 – 73xx 100 – reserve 101 – 81xx 110 – 82xx 111 – 83xx Chip ID[4:0] – Chip Index

因 codec 均挂载 i2c1 上，故 `i2cget -y -f 1 i2cAddr(7 位) 0x0E`，得到 0x40，即说明与 7243 通信正常。

8156:

8156 的 i2c 地址与接线有关，可与硬件确认

Table 2 Write Data to Register in I²C Interface Mode

	Chip Address	R/W		Register Address		Data to be written		
start	0001 00 CE	0	ACK	RAM	ACK	DATA	ACK	Stop

REGISTER 0XFD – CHIP ID1, DEFAULT 10000001

Bit Name	Bit	Description
Chip_number_id1	7:0	Chip ID

REGISTER 0XFE – CHIP ID0, DEFAULT 01010101

Bit Name	Bit	Description
Chip_number_id0	7:0	Chip ID

重复与 7243 同样的操作，读出 chip id，能匹配上，证明主芯片与 codec i2c 通信正常。

确认 dts 中 I2S 设定是否正常

修改 dts 中 sound 节点的 i2s-padmux 与原理图匹配，可与硬件确认。Sound 节点如果没有在 dts 中定义的话，一般都在 infinity2m.dtsi 里面。

```
sound {
    compatible = "sstar, audio";
    reg = <0x1F000000 0x1000000>;
    interrupts=<GIC_SPI INT_IRQ_BACH IRQ_TYPE_LEVEL_HIGH>;
    playback-volume-level=<64>; //0~94
    capture-volume-level=<64>;
    // micin-pregain-level=<1>; //0~3
    micin-pregain-level=<0>; //0~3
    micin-gain-level=<3>; //0~7
    linein-gain-level=<2>; //0~7
    amp-gpio = <PAD_FUART_RX 1>;
    clocks = <&CLK_upll_384m>;
    // playback-dma-buffer=<98304>; //512(ms)*48(kHz)*2(ch)*2(16bits)
    // capture-dma-buffer=<122880>; //640(ms)*48(kHz)*2(ch)*2(16bits)
    digmic-padmux = <2>;
    i2s-padmux = <3>;
    keep-i2s-clk = <0>;
    status = "ok";
};
```

在 padmux 中去掉 I2S pin 脚的相关信息

因顺芯 codec 只要先给了 I2S clock 就会切到硬件模式，为了避免开机就出 I2S 信号，切记不能在 padmux 中填入 I2S 的 pin 脚定义，同时也需要将 I2S 相应的 pin 脚的其他复用功能从 padmux 中拿掉。

Insmod mhal.ko 时加入 insmod 参数

在 insmod mhal.ko 时加入 IS_MHAL_SUPPORT_ES_CODEC=1，来更改 mhal 的行为，开机不出 I2S 信号来适配 codec。

执行 demo，确认两颗 codec 是否工作正常

Demo 源码所在的路径为 sdk/verify/mi_demo/geonosis/codec。根据硬件设计修改以下头文件中的 i2c 地址（7 位）。

```
harvey@ubuntu:~/Code/sdk/verify/mi_demo/geonosis/codec$ ls
codec.c  codec.mk  es7210.c  es7210.h  es7243.c  es7243.h  es8156.c  es8156.h  i2c.c  i2c.h  music.7z  README.txt
```

测试 8156:

```
prog_codec -D 1 -O -i xxx.wav -V 0
```

注意：因主芯片没法提供 mclk，故 xxx.wav 的采样率需大于等于 16K。

异常排查：

1. cat /proc/mi_modules/mi_ao/mi_ao1，查看应用是否有一直在送数据（即 TotalFrmCnt 是否一直在增加）
2. 尝试增加 -V 参数，加大音量(-60 ~ 30)
3. 用示波器测量主芯片端 I2S 波形，确认 BCLK（bit 传输时钟，该时钟频率应为 2（通道数） * 16（位宽） * 采样率），WCLK（左右声道切换时钟，时钟频率为采样率），SDO（能看

到有数据变化即可) 是否有波形, 没有波形则需要确认 I2S 是否配置正确。

4. 测试 codec 端上述波形, 确认 codec 是否接收到 I2S 信号
5. 测量 codec 端的 VRef 电压, 此电压能直观地看出 codec 是否正常工作

测试 7243:

`prog_codec -t 30 -l -o /tmp -d 2 -m 0 -c 2 -s 48000` (若需要和没有挂在 codec 上的 mic 一同使用, 则需要使用 `-d 4 -c 4`)

该指令会在 /tmp 下生成两个音档

异常排查:

1. `cat /proc/mi_modules/mi_ai/mi_ai2`, 查看应用是否有一直在送数据 (即 `ReadFrmCnt` 是否一直在增加)
2. 用示波器测量主芯片端 I2S 波形, 确认 BCLK (bit 传输时钟, 该时钟频率应为 $2 \times (\text{通道数}) \times 16 \times (\text{位宽}) \times \text{采样率}$), WCLK (左右声道切换时钟, 时钟频率为采样率), SDI (能看到有数据变化即可) 是否有波形, 没有波形则需要确认 I2S 是否配置正确。
3. 测试 codec 端上述波形, 确认 codec 是否发出 I2S 信号
4. 测量 codec 端的 VRef 电压, 此电压能直观地看出 codec 是否正常工作

注: 在测量 I2S 信号时, 需要特别关注 SDI/SDO 是否有数据。

注意事项:

因为顺芯 codec 的原因, 建议先对通过 I2C 对顺芯 codec 做了初始化, 再初始化 MI_AI/MI_AO。