

AW889x Android Driver (Qcom)

INFORMATION

HAL File	Mixer_paths_xxx.xml
Driver File	aw889x.c, aw889x.h, aw889x_reg.h aw889x_reg_table.h
Smart PA	aw889x
I ² C Address	0x34/0x35/0x36/0x37
ADB Debug	yes
Platform	Msm8953

PROJECT CONFIG

```
#add aw889x smartpa  
CONFIG_SND_SMARTPA_AW889X=y
```

AUDIO DEVICE

Mixer_paths_xxx.xml 是一个高通音频 path 管理文件，可以对相应的 control 进行 turn on, turn off, setting。在项目对应的 mixer_paths_xxx.xml 文件中添加 aw889x 的 kcontrol 控制。

在初始化列表中添加 SmartPa 控件：

```
<!-- These are the initial mixer settings -->  
<ctl name="Voice Rx Device Mute" id="0" value="0" />  
<ctl name="Voice Rx Device Mute" id="1" value="-1" />  
.....  
<ctl name="aw889x_speaker_switch" value="Off" />  
.....
```

在对应的 speaker path 中添加 SmartPa 控件：

```
<path name="speaker">  
<ctl name="aw889x_speaker_switch" value="On">  
</path>
```

KERNEL DRIVER

AW889x Smart PA Driver

1. 修改 dts

打开 kernel-3.18/arch/arm/boot/dts/ *.dts 文件，添加 aw889x 的配置

```
/* AWINIC AW889X Smart PA */
&i2c_x {                                     /*x 表示对应的总线号*/
    /*aw889x_smartpa@34*/
    i2c_smartpa@34 {
        compatible = "awinic,i2c_smartpa";
        reg = <0x34>;
        reset-gpio = <&tlmm 67 0>;
        /*irq-gpio = <&tlmm 132 0x01>;*/
        /*interrupt-parent = <&tlmm>;*/
        /*interrupts = <132 8>;*/
        status = "okay";
    };
};
/* AWINIC AW889X Smart PA End */
```

2. 添加驱动文件

在 kernel-3.18/sound/soc/codecs/aw 目录下添加 aw889x 驱动文件 aw889x.c, aw889x.h, aw889x_reg.h,aw889x_reg_table.h

3. 更新 Kconfig 和 Makefile

1) 在 kernel-3.18/sound/soc/codecs/Kconfig 中添加

```
config SND_SMARTPA_AW889X
    tristate "SoC Audio for awinic aw889x series"
    depends on I2C
    help
        This option enables support for aw889x series Smart PA.
```

2) 在 kernel-3.18/sound/soc/codecs/Makefile 中添加

```
#for AWINIC AW889X Smart PA
obj-$(CONFIG_SND_SMARTPA_AW889X) += awinic/aw889x.o
```

4. 添加 aw889x fw&cfg 文件

1) 在 kernel-3.18/drivers/base/firmware_class.c 中添加 bin 文件目录，目录由系统决定，一般目录为 /system/vendor/firmware 或 /system/etc/firmware

```
static const char * const fw_path[] = {
    fw_path_para,
    "/system/vendor/firmware",
    "/system/etc/firmware",
    "/lib/firmware/updates/" UTS_RELEASE,
    "/lib/firmware/updates",
    "/lib/firmware/" UTS_RELEASE,
    "/lib/firmware"
};
```

2) 使用 adb 将 ram 文件 push 到手机中

```
adb push aw889x_fw.bin /system/vendor/firmware/  
adb push aw889x_cfg.bin /system/vendor/firmware/
```

ASoc Machine Driver （以 Quinary i2s 为例）

1. snd_soc_dai_link

在 kernel-3.18/sound/soc/msm/msm8952.c 中添加

```
#ifdef CONFIG_SND_SMARTPA_AW889X  
{  
.name = LPASS_BE_QUIN_MI2S_RX,  
.stream_name = "Quinary MI2S Playback",  
.cpu_dai_name = "msm-dai-q6-mi2s.5",  
.platform_name = "msm-pcm-routing",  
.codec_name = " aw889x_smartpa ",  
.codec_dai_name = " aw889x-aif ",  
.dai_fmt = SND_SOC_DAIFMT_I2S | SND_SOC_DAIFMT_NB_NF |  
SND_SOC_DAIFMT_CBS_CFS,  
.dpcm_playback = 1,  
.no_pcm = 1,  
.be_id = MSM_BACKEND_DAI_QUINARY_MI2S_RX,  
.be_hw_params_fixup = msm_mi2s_rx_be_hw_params_fixup,  
.ops = &msm8952_quin_mi2s_be_ops,  
.ignore_pmdown_time = 1, /* dai link has playback support */  
.ignore_suspend = 1,  
},  
#else  
{  
.name = LPASS_BE_QUIN_MI2S_RX,  
.stream_name = "Quinary MI2S Playback",  
.cpu_dai_name = "msm-dai-q6-mi2s.5",  
.platform_name = "msm-pcm-routing",  
.codec_dai_name = "msm_hdmi_dba_codec_rx_dai",  
.codec_name = "msm-hdmi-dba-codec-rx",  
.no_pcm = 1,  
.dpcm_playback = 1,  
.be_id = MSM_BACKEND_DAI_QUINARY_MI2S_RX,  
.be_hw_params_fixup = msm_mi2s_rx_be_hw_params_fixup,  
.ops = &msm8952_quin_mi2s_be_ops,  
.ignore_pmdown_time = 1, /* dai link has playback support */  
}
```

```
.ignore_suspend = 1,  
},  
#endif
```

DEBUG INTERFACE

AW889x Driver 会创建 reg/dsp 2 个设备节点文件，路径是 sys/bus/i2c/driver/i2c_smartpa/*-00xx，其中*为 i2c bus number，xx 为 i2c address。

可以使用 adb 配置 reg/dsp 参数，调试 AW889x 音效。

也可以使用 sys/kernel/debug/asoc/msm89xx/codec:aw889x_smartpa/codec_reg 节点查看 smartpa 相关寄存器状态。

reg

用于读写 AW889x 的所有寄存器。

节点使用：

读寄存器值：cat reg

写寄存器值：echo "reg_addr reg_data" > reg （16 进制操作）

参考例程：

cat reg

```
reg:0x00=0x0310  
reg:0x01=0x5100  
reg:0x02=0x4000  
reg:0x03=0x3fff  
reg:0x04=0x0241  
reg:0x05=0x0ce8  
reg:0x06=0x0030  
reg:0x07=0x00a0  
reg:0x08=0xa00f  
reg:0x09=0x0401  
reg:0x0a=0x6464  
reg:0x20=0x0007  
reg:0x21=0x2101  
reg:0x22=0x0004  
reg:0x23=0x1f1f  
reg:0x60=0x1aa2  
reg:0x61=0xd6fe  
reg:0x62=0x307f  
reg:0x63=0x0002  
reg:0x64=0x0000  
reg:0x65=0x0422  
reg:0x66=0x0154  
reg:0x67=0x0500
```

echo "0x04 0x0241" > reg

（向 0x04 寄存器写值 0x0284）

dsp

用于读写 dsp 中的 firmware 和 config。

节点使用:

读取 dsp: cat dsp

更新 dsp: echo 1 > dsp (更新 aw889x_fw.bin/aw889x_fw.bin)

参考例程:

cat dsp

1. 无音乐

```
root@n391b:/sys/bus/i2c/drivers/i2c-smartpa/3-0034 # cat dsp
aw889x plls=0, no iis signal
root@n391b:/sys/bus/i2c/drivers/i2c-smartpa/3-0034 #
```

2. 有音乐

```
aw889x dsp working
aw889x dsp firmware:
0x30,0xb5,0x0d,0x46,0x31,0x49,0x32,0x4c,
0xb1,0xb0,0x89,0x6b,0x21,0x60,0xc9,0x07,
0x04,0xd0,0x08,0x22,0x10,0xa9,0x00,0xf0,
0x87,0xf8,0x10,0xa8,0x21,0x68,0x89,0x07,
0x07,0xd5,0x01,0x46,0x2b,0x48,0x08,0x23,
0x6a,0x46,0x00,0x68,0x00,0xf0,0x82,0xf8,
0x68,0x46,0x21,0x68,0x49,0x07,0x07,0xd5,
0x01,0x46,0x27,0x48,0x08,0x23,0x08,0xaa,
0x00,0x68,0x00,0xf0,0x77,0xf8,0x08,0xa8,
0x21,0x68,0x09,0x07,0x07,0xd5,0x01,0x46,
0x22,0x48,0x08,0x23,0x6a,0x46,0x00,0x68,
0x00,0xf0,0x6c,0xf8,0x68,0x46,0x20,0x49,
0x08,0x23,0x0a,0x68,0x18,0xa9,0x00,0xf0,
0x6b,0xf8,0x21,0x68,0x18,0xa8,0x09,0x06,
0x03,0xd5,0x20,0xa9,0x00,0xf0,0x6a,0xf8,
0x20,0xa8,0x21,0x68,0xc9,0x05,0x03,0xd5,
0x28,0xa9,0x00,0xf0,0x69,0xf8,0x28,0xa8,
0x21,0x68,0xc9,0x06,0x03,0xd5,0x08,0xa9,
0x00,0xf0,0x2b,0xf8,0x08,0xa8,0x13,0x49,
0x08,0x23,0x0a,0x68,0x29,0x46,0x00,0xf0,
0x61,0xf8,0x31,0xb0,0x30,0xbd,0x09,0x48,
0x09,0x49,0x80,0x6b,0x00,0xbf,0x08,0x60,
0x00,0xbf,0x00,0xbf,0x00,0xbf,0x00,0xbf,
0x00,0xbf,0x00,0xbf,0x00,0xbf,0x70,0x47,
0x02,0x48,0x03,0x49,0x80,0x6b,0x08,0x60,
0x70,0x47,0x00,0x00,0x40,0x07,0x00,0x20,
0x24,0x00,0x00,0x20,0x08,0x00,0x00,0x20,
0x0c,0x00,0x00,0x20,0x10,0x00,0x00,0x20,
0x04,0x00,0x00,0x20,0x18,0x00,0x00,0x20,
```

Codec_reg

Cat codec_reg

xxxx: 表示该寄存器不可读

```
at codec_reg
00: 0310
01: 5100
02: 5391
03: 3fff
04: 0241
05: 0ce8
06: 0030
07: 00a0
08: a00f
09: 0401
0a: 6464
0b: XXXX
0c: XXXX
0d: XXXX
0e: XXXX
0f: XXXX
10: XXXX
11: XXXX
12: XXXX
13: XXXX
14: XXXX
15: XXXX
16: XXXX
17: XXXX
18: XXXX
19: XXXX
```

echo "reg_addr reg_data" > codec_reg 设置寄存器值

如: echo "0x05 0x01ce8" > codec_reg