AW881XX Android Driver

INFORMATION

HAL File	AudioParamOptions.xml SmartPa_AudioParam.xml SmartPa_ParamUnitDesc.xml
Driver File	aw881xx.c, aw881xx.h, aw881xx_reg.hss aw881xx_monitor.c, aw881xx_monitor.h aw881xx_cali.c, aw881xx_cali.h, aw_data_type.h
Smart PA	aw88164/aw88164a/aw88194/ aw88194a/aw88195
I²C Address	0x34/0x35/0x36/0x37
ADB Debug	yes
Platform	mt6739
Android version	android 9.0

PROJECT CONFIG

在 ProjectXXX.mk 中添加

```
MTK_AUDIO_SPEAKER_PATH = smartpa_awinic_aw881xx
```

AUDIO DEVICE

添加 aw881xx 选项

在 xxx/audio_param/AudioParamOptions.xml 中添加 aw881xx 选项

```
<Param name="MTK_AUDIO_SPEAKER_PATH" value="smartpa_awinic_aw881xx" />
```

添加 aw881xx 参数配置

在 device/mediatek/common/audio_param_smartpa/SmartPa_AudioParam.xml 添加 aw881xx 参数配置

添加 aw881xx：

```
<Param path="smartpa_awinic_aw881xx" param_id="7"/>
```

添加 aw881xx 参数：

```
<ParamUnit param_id="7">
    <Param name="have_dsp" value="1"/>
    <Param name="is_alsa_codec" value="1"/>
    <Param name="chip_delay_us" value="22000"/>
    <Param name="supported_rate_list"
value="8000,11025,12000,16000,22050,24000,32000,44100,48000"/>
    <Param name="spk_lib_path" value=""/>
    <Param name="spk_lib64_path" value=""/>
    <Param name="codec_ctl_name" value="aw_dev_0_prof"/>
    <Param name="is_apll_needed" value="1"/>
    <Param name="i2s_set_stage" value="5"/>
```



```
</ParamUnit>
```

添加 aw881xx 描述

在 device/mediatek/common/audio_param_smartpa/SmartPa_ParamUnitDesc.xml 中添加
<Category name="smartpa_awinic_aw881xx"/>

KERNEL DRIVER

xxx_defconfig

```
#add aw881xx smartpa
CONFIG_SND_SMARTPA_AW881XX=y
```

AW881XX Smart PA Driver

1. 修改 dts

打开 kernel/arch/arm/boot/dts/ *.dts 文件，添加 aw881xx 的配置

如下为单 PA 的配置：

```
/* AWINIC AW881XX Smart PA */
&i2c3 {
    aw881xx_smartpa@34 {
        compatible = "awinic,aw881xx_smartpa";
        reg = <0x34>;
        reset-gpio = <&pio 65 0>;
        irq-gpio = <&pio 5 0>;
        fade-enable = <0>;
       pa-syn-enable = <0>;
        status = "okay";
    };
};
/* AWINIC AW881XX Smart PA End */
```

若为多 PA 项目，则增加 i2c 节点即可，这里以双 PA 为例，注意：不同 i2c 节点 sound-channel 属性需要不同。

```
/* AWINIC AW881XX Smart PA */
&i2c3 {
    aw881xx_smartpa@34 {
        compatible = "awinic,aw881xx_smartpa";
        reg = <0x34>;
        reset-gpio = <&pio 65 0>;
        irq-gpio = <&pio 5 0>;
       sound-channel = <0>;
        fade-enable = <0>;
       pa-syn-enable = <0>;
        status = "okay";
    };
    aw881xx_smartpa@35 {
        compatible = "awinic,aw881xx_smartpa";
```



```
        reg = <0x35>;
        reset-gpio = <&pio 63 0>;
        irq-gpio = <&pio 8 0>;
        sound-channel = <1>;
        fade-enable = <0>;
        pa-syn-enable = <0>;
        status = "okay";
    };
};
/* AWINIC AW881XX Smart PA End */
```

2. 添加驱动文件

在 kernel/sound/soc/codecs 目录下添加 aw881xx 驱动文件 aw881xx.c, aw881xx.h,aw881xx_reg.h,aw881xx_monitor.c,aw881xx_monitor.h,aw881xx_cali.c, aw881xx_cali.h, aw_data_type.h

注意：若在 codec_probe 函数中无法加载到 bin 文件，请修改 aw881xx.h 中如下的宏，增加延时加载时间

```
#define AW_LOAD_BIN_TIME_MS     (1000)
```

3. 更新 Kconfig 和 Makefile

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

```
config SND_SMARTPA_AW881XX
    tristate "SoC Audio for awinic aw881xx series"
    depends on I2C
    help
      This option enables support for aw881xx series Smart PA.
```

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

```
#for AWINIC AW881XX Smart PA
obj-$(CONFIG_SND_SMARTPA_AW881XX) += aw881xx.o aw881xx_monitor.o aw881xx_cali.o
```

4. 添加 aw881xx fw&cfg 文件

1) 在 kernel/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 config 文件 push 到手机的 vendor 目录

ASoc Machine Driver

4G 平台配置

若平台为 4G 平台，在 kernel/sound/soc/mediatek/common_int/mtk-soc-machine.c 中 mt_soc_extspk_dai 添加 aw881xx 的 dai_link 配置，其他默认保持平台不变。

若为单 PA 项目，则添加以下信息，注意 6-0034 对应的总线与地址;

```
struct snd_soc_dai_link_component awinic_codecs[] = {
    {
        .of_node = NULL,
        .dai_name = "aw881xx-aif-6-34",
        .name = "aw881xx_smartpa.6-0034",
    },
};
```

若为多 PA 项目，则在该数组中继续添加设备信息，这里以双 PA 为例，一个 PA 的总线与地址为 6-0034，另一个 PA 的总线与地址为 6-0035

```
struct snd_soc_dai_link_component awinic_codecs[] = {
    {
        .of_node = NULL,
        .dai_name = "aw881xx-aif-6-34",
        .name = "aw881xx_smartpa.6-0034",
    },
    {
        .of_node = NULL,
        .dai_name = "aw881xx-aif-6-35",
        .name = "aw881xx_smartpa.6-0035",
    },

};
```

```
    {
        .name = "Ext_Speaker_Multimedia",
        .stream_name = MT_SOC_SPEAKER_STREAM_NAME,
        .cpu_dai_name = "snd-soc-dummy-dai",
        .platform_name = "snd-soc-dummy",
#ifdef CONFIG_SND_SMARTPA_AW881XX
        .num_codecs = ARRAY_SIZE(awinic_codecs),
        .codecs = awinic_codecs,
#endif
        .ops = &mt_machine_audio_ops,
    },
```

5G 平台配置

若为 5G 平台则在 kernel/ sound/soc/mediatek/mt6853/mt6853-mt6359.c 中添加如下信息

若为单 PA 项目，则添加以下信息，注意 6-0034 对应的总线与地址,

```
struct snd_soc_dai_link_component awinic_codecs[] = {
    {
        .of_node = NULL,
        .dai_name = "aw881xx-aif-6-34",
        .name = "aw881xx_smartpa.6-0034",
    },
};
```

若为多 PA 项目，则在该数组中继续添加设备信息，这里以双 PA 为例，一个 PA 的总线与地址为 6-

0034，另一个 PA 的总线与地址为 6-0035

```
struct snd_soc_dai_link_component awinic_codecs[] = {
    {
        .of_node = NULL,
        .dai_name = "aw881xx-aif-6-34",
        .name = "aw881xx_smartpa.6-0034",
    },
    {
        .of_node = NULL,
        .dai_name = "aw881xx-aif-6-35",
        .name = "aw881xx_smartpa.6-0035",
    },

};
```

```
@@ -779,8 +789,13 @@ static struct snd_soc_dai_link mt6853_mt6359_dai_links[] =
{
    {
        .name = "I2S3",
        .cpu_dai_name = "I2S3",
#ifdef CONFIG_SND_SMARTPA_AW881XX
        .num_codecs = ARRAY_SIZE(awinic_codecs),
        .codecs = awinic_codecs,
#else
        .codec_dai_name = "snd-soc-dummy-dai",
        .codec_name = "snd-soc-dummy",
#endif
        .no_pcm = 1,
        .dpcm_playback = 1,
        .ignore_suspend = 1,

@@ -789,8 +804,13 @@ static struct snd_soc_dai_link mt6853_mt6359_dai_links[] =
{
    {
        .name = "I2S0",
        .cpu_dai_name = "I2S0",
#ifdef CONFIG_SND_SMARTPA_AW881XX
        .num_codecs = ARRAY_SIZE(awinic_codecs),
        .codecs = awinic_codecs,
#else
        .codec_dai_name = "snd-soc-dummy-dai",
        .codec_name = "snd-soc-dummy",
#endif
        .no_pcm = 1,
        .dpcm_capture = 1,
        .ignore_suspend = 1,
```

SPEAKER CALI

AW881XX 需要产线 speaker 校准，在 speaker 的测试符合要求后，需要将值写入到 AP 的 persist 分区中，然后在开机初次配置 dsp 的时候，将校准值写入到 aw881xx 中，完成 speaker 校准。

AW881XX 工作是需要 I2S 信号的，因此产线的 speaker 校准可以在工厂模式测试 speaker 声音的时候操作。

AW881XX 的校准是通过 HAL 层的命令直接控制的，因此需要使用 aw881xx_cali 可执行文件。
参数:（注：不填写 I2C bus 和 I2C Addr 即校准所有 PA;参数中校准 re 时间单位为 ms，最小值为 1000ms）

File	Cmd	Device_name	I2C bus	I2C Addr	Param	Description
aw881xx_cali	start_cali	aw881xx_smartpa	6	34		Re/F0/q 校准
aw881xx_cali	re_cali	aw881xx_smartpa	6	34		Re 校准
aw881xx_cali	f0_cali	aw881xx_smartpa	6	34		F0 获取
aw881xx_cali	fast_start_cali	aw881xx_smartpa	6	34	校准 re 时间	缩短校准 re 时间
aw881xx_cali	fast_re_cali	aw881xx_smartpa	6	34	校准 re 时间	缩短校准 re 时间
aw881xx_cali	store_fixed_re	aw881xx_smartpa	6	34	re 值	存储定点化 Re

操作步骤：
1. 正常播放静音文件；
2. 执行 AW881XX 的校准，参考命令
   ./system/bin/aw881xx_cali start_cali <device name> <i2c bus> <i2c addr>
   示例：
   ./system/bin/aw881xx_cali start_cali aw881xx_smartpa

```
255|msm8996:/ # aw881xx_cali start_cali aw881xx_smartpa
[AWINIC] [INF] main: firmware version: v1.0.6
[AWINIC] [INF] aw881xx_cali_get_re: [6-0034]re=7.477295ohm
[AWINIC] [INF] aw881xx_cali_get_re: [6-0034]fixed_re=7477mohm
[AWINIC] [INF] aw881xx_cali_get_re: [6-0035]re=7.820557ohm
[AWINIC] [INF] aw881xx_cali_get_re: [6-0035]fixed_re=7820mohm
[AWINIC] [INF] aw881xx_multdev_cali_get_f0: [6-0034]f0=823.761536
[AWINIC] [INF] aw881xx_multdev_cali_get_f0: [6-0035]f0=854.489624
```

3. 由 AP 读取 re、f0，校准后的 Re、f0 存储 Node 中，可以用于判别是否符合范围。（为了保证 Re 的精度需求，Node 的 Re 放大了 1000 倍）。

Node	Path
Re	sys/bus/i2c/drivers/<device name>/<i2c bus>-<i2c addr>/re
F0	sys/bus/i2c/drivers/<device name>/<i2c bus>-<i2c addr>/f0

4. AP 判断 re、f0 是否符合要求，如果 re、f0 在设置范围内，表示 speaker 测试正常，需要将 re 的数据写到 AP 的 NVRAM 或者 persist，参考命令

```
./system/bin/aw881xx_cali store_fixed_re <device name> <i2c bus> <i2c addr> <re_ste>
./system/bin/aw881xx_cali store_fixed_re aw881xx_smartpa 6 34 8000
```

注：Re 的值需要存储在手机的 NV 或者 persist，因此，需要在 Driver 中实现函数：

```
aw881xx_set_cali_re_to_phone();
aw881xx_get_cali_re_from_phone();
```

Re 写入验证：

a.查看 re 值是否写入文件中
b.cat dsp_re 节点，确认 dsp 中的值与写入值相同
c.重启手机，播放音乐，再次 cat dsp_re 节点，确认 dsp 中的值与文件中的 re 值

可执行文件使用：

除了直接在命令行调用 aw881xx_cali 可执行文件外，Awinic 也提供了调用 aw881xx_cali 可执行文件的示例代码 aw881xx_call_exe.c,在该文件中,需要设置或获取的变量如下：

变量名	意义	模式
exe_name	可执行文件的名字，默认值为 aw881xx_cali	任何命令均需设置

cmd_type	命令类型，如 start_cali 等	
dev_name	设备名称，默认为 aw881xx_smartpa	
fixed_re	定点化电阻值	store_fixed_re 命令需要设置
time	设置校准 re 的延时时间	fast_start_cali 命令和 fast_re_cali 命令需要设置
aw_re_f0	存储校准得到的 re、f0	

DEBUG INTERFACE

Node

AW881XX Driver 会创建 cali/ update/dsp_rw/dsp/dsp_fw_ver/reg/rw/monitor 8 个设备节点文件，路径是 sys/bus/i2c/driver/aw881xx_smartpa/*-00xx，其中*为 i2c bus number，xx 为 i2c address。
可以使用 adb 配置 cali/ update/dsp_rw/dsp/dsp_fw_ver/reg/rw/monitor 参数调试 aw881xx。

reg

用于读写 AW881XX 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

echo 0x04 0x0241 > reg （向 0x04 寄存器写值 0x0241）

rw

用于读写 AW881XX 的单个寄存器。

节点使用：

读寄存器值：
echo reg_addr > rw （16 进制操作）
cat rw
写寄存器值：
echo reg_addr reg_data > rw （16 进制操作）

参考例程：

echo 0x04 > rw （读取 0x04 寄存器值）
cat rw

echo 0x04 0x0241 > rw （向 0x04 寄存器写值 0x0241）

update

用于读写 AW881XX 的 register 和 dsp 的配置。

节点使用：

更新 register 和 dsp 寄存器值：
echo 1 > update

dsp_rw

用于读写 AW881XX 的 dsp 的单个寄存器。

节点使用：

读寄存器值：
echo reg_addr > dsp_rw （16 进制操作）
cat rw
写寄存器值：
echo reg_addr reg_data > dsp_rw （16 进制操作）

参考例程：

echo 0x8601 > dsp_rw （读取 dsp 的 0x8604 寄存器值）
cat dsp_rw

echo 0x8604 0x4011 > dsp_rw （向 dsp 的 0x8604 寄存器写值 0x4011）

cali

用于配置 AW881XX 的 re 校准。

节点使用：

更新校准 re
echo re > cali
读取校准 re
cat cali

monitor

用于低温低压保护的开关

节点使用：

开关 monitor
echo 1 > monitor
echo 0 > monitor
读取 monitor 状态
cat monitor

dsp_fw_ver

用于查看 dsp fw 版本号

节点使用：

cat dsp_fw_ver

re

校准结束后查看校准的 re 值

节点使用：

cat re

f0

校准结束后查看校准的 f0 值

节点使用：

cat f0

dsp_re

获取 dsp 中的 re 值（不含线损）

节点使用：

cat dsp_re

spk_temp

查看当前喇叭温度

节点使用：

cat spk_temp

monitor_update

更新低温低压 bin 文件

节点使用：

echo 1 > monitor_update

r0

获取实时 re 值

节点使用：

cat r0

Kcontrol

aw_dev_0_switch

PA 开关
tinymix aw_dev_0_switch Enable　PA 允许开启
tinymix aw_dev_0_switch Disable　PA 不允许开启

aw_dev_0_rx_volume

音量配置，每步为 0.5dB，范围：[0~-96dB]
tinymix aw_dev_0_rx_volume 0　　选择 0dB
tinymix aw_dev_0_rx_volume 6　　选择-3dB

aw_dev_0_prof

模式切换(假设 bin 文件中配置了 Music 和 Receive 模式)
tinymix aw_dev_0_prof Music　　切换到 Music 模式
tinymix aw_dev_0_prof Receive　　切换到 Receive 模式

aw881xx_fadein_us

每个步进的淡入时间设置
tinymix aw881xx_fadein_us 500　　将每个步进淡入时间间隔设置为 500us

aw881xx_fadeout_us

每个步进的淡出时间设置
tinymix aw881xx_fadeout_us 500　　将每个步进淡出时间间隔设置为 500us

