

AW87XXX Android Driver(MTK)

版本: V3.0

时间: 2020 年 10 月

修订记录

日期	版本	描述	作者
2020-03-05	V1.0	驱动兼容	张鹏彪
2020-07-31	V2.0	驱动兼容	姜拴雷
2020-09-30	V3.0	加入动态降噪相关节点	姜拴雷

目录

AW87XXX ANDROID DRIVER	4
1. INFORMATION	4
2. PROJECT CONFIG	4
3. KERNEL DRIVER.....	4
3.1 AW87XXX SMART K PA DRIVER	4
3.2 MT6357 CODEC DRIVER.....	6
4. DEBUG INTERFACE.....	7
4.1 HWEN	7
4.2 REG	7
4.3 UPDATE.....	7
4.4 MODE	7
4.5 VMAX.....	8
4.6 VBAT.....	8
4.7 MONITOR	8
4.8 VMAX_TIME	8
4.9 ACTFLAG.....	8
4.10 AFTASK	8
4.11 AUTO_THREAD.....	9
4.12 ACTGAIN	9

AW87XXX Android Driver

1. Information

Driver File	aw87xxx.c, aw87xxx.h, aw87339.h, aw87359.h, aw87519.h, aw87549.h, aw87559.h, aw87xxx_monitor.c, aw87xxx_monitor.h, aw_bin_parse.c, aw_bin_parse.h
Support i2c	aw87329, aw87339, aw87349, aw87359, aw87389, aw87509, aw87519, aw87529, aw87539, aw87369, aw87559
I ² C Address	请参考对应的 datasheet
ADB Debug	Yes
Platform	mt6739

2. Project Config

```
#add aw87xxx smartpa
CONFIG_SND_SOC_AW87XXX=y
```

3. Kernel Driver

3.1 AW87XXX Smart K PA Driver

3.1.1 dts

打开 kernel-4.4/arch/arm/boot/dts/*.dtsi 文件，添加 aw87xxx 的配置

注意：由于 aw87359 和 aw87389 是没有复位引脚的，故 aw87359 和 aw87389 不需要配置 reset-gpio；
pa-channel = < 0 > 为左声道，pa-channel = < 1 > 为右声道；gain-reg-addr 为增益寄存器地址，
需根据 datasheet 判定。

```
&i2c_x { /*x 表示对应的总线号*/
    /* AWINIC AW87XXX Smart K PA */
    aw87xxx_pa_58@58 {
        compatible = "awinic,aw87xxx pa 58";
        reg = <0x58>;
        reset-gpio = <&pio 63 0>;
        pa-channel = < 0 >;
        /* actflag-gain-array: off music voice fm rcv */
        actflag-gain-array = < 0x00 0x08 0x08 0x08 0x00 >;
        /*actflag-auto-run;*/
        gain-reg-addr = <0x06>;
        monitor-flag = <1>;
        monitor-timer-val = <3000>;
        monitor-timer-count-max = <5>;
        status = "okay";
    };

    aw87xxx_pa_59@59 {
        compatible = "awinic,aw87xxx pa 59";
        reg = <0x59>;
        reset-gpio = <&pio 65 0>;
    };
}
```

```

pa-channel = < 1 >;
/* actflag-gain-array: off music voice fm rcv */
actflag-gain-array = < 0x00 0x08 0x08 0x08 0x00 >;
/*actflag-auto-run;*/
gain-reg-addr = <0x05>;
monitor-flag = <1>;
monitor-timer-val = <3000>;
monitor-timer-count-max = <5>;
status = "okay";
};
/* AWINIC AW87XXX Smart K PA End */
/* AWINIC AW87XXX Smart K PA End */
};

```

3.1.2 Driver

在 kernel-4.4/sound/soc/mediatek 目录下添加 aw87xxx.c, aw87xxx_monitor.c, aw_bin_parse.c, aw87xxx.h, aw87xxx_monitor.h, aw_bin_parse.h, aw87579.h, aw87569.h, aw87559.h, aw87549.h, aw87519.h, aw87359.h, aw87339.h 文件。

3.1.3 Kconfig && Makefile

1) 在 kernel-4.4/sound/soc/mediatek/Kconfig 中添加

```

config SND_SOC_AW87XXX
    tristate "SoC Audio for awinic AW87XXX Smart K PA"
    depends on I2C
    help
        This option enables support for AW87XXX Smart K PA.

```

2) 在 kernel-4.4/sound/soc/mediatek/Makefile 中添加

```

#for AWINIC AW889X Smart K PA
obj-$(CONFIG_SND_SOC_AW87XXX) += awinic/aw87xxx.o
awinic/aw87xxx_monitor.o awinic/aw_bin_parse.o

```

3.1.4 AW87XXX Config Bin File

1) 在 kernel-4.4/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 将场景 bin 文件和 vmax bin 文件 push 到手机中的 /system/vendor/firmware, bin 文件可以通过各自产品的 UI 生成。

注意:

场景 bin 文件命名的格式为: aw87xxx_pid_num_scene_channel.bin, 其中 num 为芯片的 id, scene 为该 bin 文件对应的场景, channel 依据 pa-channel 生成, 表示声道编号, 如下面 push 的 bin 文件其

对应的芯片 id 号为 0x59, 场景为 music, channel 为 0:

adb push aw87xxx_pid_59_music_0.bin /system/vendor/firmware
vmax bin 文件的命名格式为: aw87xxx_vma_channel.bin, 其中 channel 依据 pa-channel 生成, 表示声道编号, 如下面 push 的 bin 文件 channel 为 0:

adb push aw87xxx_vmax_0.bin /system/vendor/firmware

chip id 与 Smart K PA 的对应关系如下图:

Chip id	Smart K PA
0x39	aw87329, aw87339, aw87349
0x59	aw87359, aw87509, aw87519, aw87529, aw87539, aw87389
0x5A	Aw87559
0x69	aw87369

3.2 MT6357 Codec Driver

在 kernel-4.4 /sound/soc/mediate/codecs/mt6357/mtk-soc-codec-6357.c 中添加

```
enum {
    AW87XXX_OFF_MODE 0,
    AW87XXX_MUSIC_MODE 1,
    AW87XXX_VOICE_MODE 2,
    AW87XXX_FM_MODE 3,
    AW87XXX_RCV_MODE 4,
    AW87XXX_MODE_MAX 5,
};

enum {
    AW87XXX_LEFT_CHANNEL 0,
    AW87XXX_RIRHT_CHANNEL 1,
};

extern unsigned char aw87xxx_show_current_mode(int32_t channel);
extern int aw87xxx_audio_scene_load(uint8_t mode, int32_t channel);

if (enable) {
    AudDrv_GPIO_EXTAMP_Select(false, 3);
    usleep_range(1 * 1000, 2 * 1000);
    #if defined(CONFIG_MTK_LEGACY)
    #elif (defined CONFIG SND SOC AW87XXX)
        aw87xxx_audio_scene_load(AW87XXX_MUSIC_MODE, AW87XXX_LEFT_CHANNEL);
    #else
        AudDrv_GPIO_EXTAMP_Select(true, 3);
    #endif
    usleep_range(5 * 1000, 10 * 1000)
} else {
    #if defined(CONFIG_MTK_LEGACY)
    #elif (defined CONFIG SND SOC AW87XXX)
        aw87xxx_audio_scene_load(AW87XXX_OFF_MODE, AW87XXX_LEFT_CHANNEL);
    #else
        AudDrv_GPIO_EXTAMP_Select(false, 3);
    #endif
    udelay(500);
}
```

4. Debug Interface

AW87XXX Driver 会根据 2 个 pa 创建两个设备文件，路径是 sys/bus/i2c/driver/aw87xxx_pa/*-00xx，在每个设备文件下各创建 hwen/reg/update/mode/vmax/vbat/monitor/vmax_time8 个设备节点，其中*为 i2c bus number，xx 为 i2c address。

4.1 hwen

节点名字	hwen
功能描述	用于控制 AW87xxx 的硬件关闭
使用方法	cat hwen (获取 AW87xxx 硬件状态) echo 1 > hwen (AW87xxx 硬件使能) echo 0 > hwen (AW87xxx 硬件关闭)

4.2 reg

节点名字	reg
功能描述	用于读写 aw87xxx 的所有寄存器
使用方法	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 (获取所有可读寄存器上的值) echo 0x01 0x07 > reg (向 0x01 寄存器写入 0x07 的值)

4.3 update

节点名字	update
功能描述	用于更新 AW87XXX 参数配置文件
使用方法	echo 1 > update (AW87XXX 参数更新)

4.4 mode

节点名字	mode
功能描述	用于配置 AW87XXX 的工作模式
使用方法	cat mode (获取不同模式对应的数字) echo 0 > mode (AW87XXX 关闭) echo 1 > mode (AW87XXX 使能，工作在 music 模式) echo 2 > mode (AW87XXX 使能，工作在 voice 模式) echo 3 > mode (AW87XXX 使能，工作在 fm 模式) echo 4 > mode (AW87XXX 使能，工作在 rcv 模式)

4.5 vmax

节点名字	vmax
功能描述	用于设置和获取 vmax 值（获取的 vmax 为算法中计算出的 vmax 值，与设置的值不相同）
使用方法	cat vmax （获取当前 vmax 的值） echo N > vmax （发送计算后的 vmax 值）
参考例程	echo 0xffff95f7e > vmax （将 0xffff95f7e 值发送给 vmax）

4.6 vbat

节点名字	vbat
功能描述	用于设置和获取当前用户输入的电量值
使用方法	cat vbat （获取当前用户输入的 vbat 值） echo capacity > vbat （设置 vbat 电量值）
参考例程	echo 50 > vbat （设置当前电量为 50%）

4.7 monitor

节点名字	monitor
功能描述	用于设置 aw87xxx 的保护功能
使用方法	cat monitor （获取当前电量的保护状态） echo 0 > monitor （aw87xxx 关闭保护） echo 1 > monitor （aw87xxx 开启保护）

4.8 vmax_time

节点名字	vmax_time
功能描述	用于设置 vmax 的上报时间
使用方法	cat vmax_time （获取 vmax_time 的值） echo N > vmax_time （设置 vmax_time 上报时间）

4.9 actflag

节点名字	actflag
功能描述	用于查看 active_flag 实时数值
使用方法	cat actflag

4.10 aftask

节点名字	aftask
功能描述	用于查看并手动开启关闭 active_flag 线程
使用方法	cat aftask echo 1 > aftask echo 0 > aftask

4.11 auto_thread

节点名字	auto_thread
功能描述	用于查看并设定 active_flag 线程自启动开关
使用方法	cat auto_thread echo 1 > auto_thread echo 0 > auto_thread

4.12 actgain

节点名字	actgain
功能描述	用于查看并输入 active_flag gain 数据
使用方法	cat actgain echo 0x00 0x08 0x08 0x08 0x00 > actgain