

AW87XXX Android Driver(MTK)

版本: V1.0

时间: 2020 年 3 月 5 日

修订记录

日期	版本	描述	作者
2020-03-05	V1.0	驱动兼容	张鹏彪

Awinic Driver

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.....	5
4. DEBUG INTERFACE.....	6
4.1 HWEN	6
4.2 REG	6
4.3 UPDATE.....	6
4.4 MODE	6
4.5 VMAX.....	7

AW87XXX Android Driver

1. Information

Driver File	aw87xxx.c, aw87xxx.h, aw87339.h, aw87359.h, aw87519.h, aw87xxx_monitor.c, aw87xxx_monitor.h
Smart K PA	aw87329, aw87339, aw87349, aw87359, aw87509, aw87519, aw87529, aw87539, aw87369
I ² C Address	0x58/0x59/0x5A/0x5B
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 是没有复位引脚的, 故 aw87359 不需要配置 reset-gpio

```
&i2c_x {                                     /*x 表示对应的总线号*/
    /* AWINIC AW87XXX Smart K PA */
    aw87xxx@58 {
        compatible = "awinic,aw87xxx_pa";
        reg = <0x58>;
        reset-gpio = <&pio 64 0>;
        status = "okay";
    };
    /* 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, aw87xxx.h, aw87xxx_monitor.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-3.18/sound/soc/mediatek/Makefile 中添加

```
#for AWINIC AW889X Smart K PA
obj-$(CONFIG_SND_SOC_AW87XXX) += aw87xxx.o aw87xxx_monitor.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.bin, 其中 num 为芯片的 id, scene 为该 bin 文件对应的场景，如下面 push 的 bin 文件其对应的芯片 id 号为 0x59, 场景为 music:

```
adb push aw87xxx_pid_59_music.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
0x69	aw87369

3.2 MT6357 Codec Driver

在 kernel-4.4 /sound/soc/mediate/codec/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,
};

extern unsigned char aw87xxx_show_current_mode(void);
extern int aw87xxx_audio_scene_load(uint32_t mode);

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);
    #endif
}
```

```
#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);
    #else
        AudDrv_GPIO_EXTAMP_Select(false, 3);
    #endif
    udelay(500);
}
```

4. Debug Interface

AW87XXX Driver 会创建 hwen/reg/update/mode/vmax 5 个设备节点，路径是 sys/bus/i2c/driver/aw87xxx_pa/*-00xx，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	用于获取 vmax 值 (获取的 vmax 为算法中计算出的 vmax 值)
使用方法	cat vmax (获取当前 vmax 的值)