AW87XXX Android Driver(MTK)

Version : V2.8

Date : Nov, 2021

REVISION RECORD

Date	Version	Description	Author
2021-03	V2.0	Compatible with qcom and mtk platforms	Zhongbo Zhao
2021-06	V2.1	Compatible with AW87418 products	Zhongbo Zhao
2021-06	V2.2	Compatible with AW87319 products	Zhongbo Zhao
2021-06	V2.3	Added the ESD detection function	Zhongbo Zhao
2021-07	V2.4	Added 4G and 5G platform function call cutting scenarios	Zhongbo Zhao
2021-07	V2.5	Optimization section description	Zhongbo Zhao
2021-09	V2.6	Compatible with AW87390 products	Huidong Zhou
2021-10	V2.6.1	Modify profile string description	Huidong Zhou
2021-11	V2.7	Modify the profile supported description of acf	Zhongbo Zhao
2021-11	V2.8	Compatible with AW87418 products	Zhongbo Zhao

CANTENTS

AW87XXX Android Driver

1. Information

Driver file	aw87xxx.c, aw87xxx.h, aw_device.c, aw_device.h, aw_monitor.c, aw_monitor.h, aw_dsp.c, aw_dsp.h, aw_acf_bin.c, aw_acf_bin.h, aw_log.h, aw_bin_parse.c, aw87xxx_pid_39_reg.h, aw87xxx_pid_59_3x9_reg.h, aw87xxx_pid_59_5x9_reg.h, aw87xxx_pid_5a_reg.h, aw87xxx_pid_18_reg.h, aw87xxx_pid_9b_reg.h, aw87xxx_pid_76_reg.h
Support product	aw87319, aw87329, aw87339, aw87349, aw87359, aw87389, aw87509, aw87519, aw87529, aw87539, aw87549, aw87559, aw87569, aw87579, aw81509 aw87390, aw87418
Support I²C	aw87418:0x5C other:0x58, 0x59, 0x5A, 0x5B
ADB debug	Yes

2. Project Config

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

3. Kernel Driver

3.1 AW87XXX Smart K PA Driver

3.1.1 Configuration Dtsi

For the device tree-based I2C device driver, add the aw87XXX device tree configuration to the kernel.

Note: Since AW87359, AW87389,AW87390 and AW87549 have no reset pin, reset-gpio does not need to be configured.

Dev_index is the number of PA, which can be set to 0,1,2,3, etc. This parameter is optional. By default, PA is numbered according to the registration sequence of PA, and the driver uses ***dev_index*** as the default parameter for parsing.

The following table shows the CHIPID and reset-pin information for each product.

PRODUCT	CHIPID	RESET_PIN
AW87319	0x9B	Y
AW87329/AW87339/AW87349	0x39	Y
AW87519/AW87529	0x59	Y
AW87359/AW87389	0x59	N
AW87549	0x5A	N
AW87559/AW87569/AW87579/AW81509	0x5A	Y
AW87390	0x76	N
AW87418	0x18	Y

Add aw87xxx configuration in the *kernel/arch/arm/boot/dts/*.dtsi.*

Single PA configuration:

```
&i2c_x {     /* x indicates the bus number of I2C */
```



```
        /* AWINIC AW87XXX Smart K PA */
        aw87xxx_pa@58 {
                    compatible = "awinic,aw87xxx_pa";
                    reg = <0x58>;
                    reset-gpio = <&pio 1 0>; /* Products with reset pins need to
be configured */
                    dev_index = < 0 >;
                    status = "okay";
        };
        /* AWINIC AW87XXX Smart K PA End */
    };
```

If you need to use the ESD detection function, please add "***esd-enable = "true";***" to the dtsi configuration.(***true: enable; false: disable***).

Double PA configuration:

```
    &i2c_x {     /* x indicates the bus number of I2C */
        /* AWINIC AW87XXX Smart K PA */
        aw87xxx_pa@58 {
                    compatible = "awinic,aw87xxx_pa";
                    reg = <0x58>;
                    reset-gpio = <&pio 1 0>; /* Products with reset pins need to
be configured */
                    dev_index = < 0 >;
                    status = "okay";
        };
        aw87xxx_pa@59 {
                    compatible = "awinic,aw87xxx_pa";
                    reg = <0x59>;
                    reset-gpio = <&pio 7 0>; /* Products with reset pins need to
be configured */
                   dev_index = < 1 >;
                   status = "okay";
        };
       /* AWINIC AW87XXX Smart K PA End */
    };
```

If you need to use the ESD detection function, please add "***esd-enable = "true";***" to the dtsi configuration.(***true: enable; false: disable***).

For the multi-PA configuration, you can refer to the preceding double PA configuration to add device nodes.

3.1.2 Driver

Add the project file of AW87XXX in the *kernel/sound/soc/awinic*.

SOURCE FILE	aw87xxx.c, aw_device.c, aw_monitor.c, aw_dsp.c, aw_acf_bin.c, aw_bin_parse.c
HEAD FILE	aw87xxx.h, aw_device.h, aw_monitor.h, aw_dsp.h, aw_acf_bin.h, aw_log.h, aw87xxx_pid_39_reg.h, aw87xxx_pid_59_3x9_reg.h, aw87xxx_pid_59_5x9_reg.h, aw87xxx_pid_5a_reg.h, aw87xxx_pid_18_reg.h, aw87xxx_pid_9b_reg.h, aw87xxx_pid_76_reg.h

3.1.3 Kconfig && Makefile

Add the Kconfig file of AW87XXX in the *kernel/sound/soc/mediatek/Kconfig*.

```
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.
```

Add the Makefile file of AW87XXX in the *kernel/sound/soc/mediatek/Makefile*.

```
#for AWINIC AW87XXX Smart K PA
obj-$(CONFIG_SND_SOC_AW87XXX) += awinic/aw87xxx.o  awinic/aw_device.o  awinic/aw_monitor.o
awinic/aw_bin_parse.o  awinic/aw_dsp.o  awinic/aw_acf_bin.o
```

3.1.4 AW87XXX Config Bin File

Add the bin file's path directory in the *kernel/drivers/base/firmware_class.c*, The path directory is determined by the system. The general path directory is */vendor/firmware* .

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

Push aw87xxx_acf.bin to /vendor/firmware directory of machine with adb tool.

Note: The ***aw87xxx_acf.bin*** file in the config directory in the software port package directory contains the default mono bin file and the default stereo configuration of the same product, and contains the default ***Music***, ***Receiver*** and ***Off*** profile configuration and the default monitor configuration(But with the exception of AW87319, the ***Off*** profile configuration is not required).

The default ***aw87xxx_acf.bin*** synthesis tool version in the software migration package is Awinic_ACF_Tool v0.0.7.

3.1.5 AW87XXX Config update time

In the driver source file aw87xxx.h, you can set the initialization firmware delay load time, The way to do this is to modify the macro definition, By default, the driver registers 5000ms after loading the firmware.

```
#define AWINIC_CFG_UPDATE_DELAY
#define AWINIC_CFG_UPDATE_DELAY_TIME    (5000)
```

When the macro AWINIC_CFG_UPDATE_DELAY is defined, the firmware is loaded after the AWINIC_CFG_UPDATE_DELAY_TIME time delay.AWINIC_CFG_UPDATE_DELAY If not defined, the firmware is directly loaded without delay.

You can set this parameter based on the client file system loading time to ensure that the firmware parameters can be properly loaded.

3.2 Codec Driver

Awinic offers two solutions to control the PA at the platform end, which customers can choose to use according to their actual needs.

3.2.1 Scheme 1: Kcontrol controls the PA switch

1） Kcontrol Register

When the Kcontrol was used to switch the profile or get real-time monitor adjustment parameters vmax, the registration of aw87xxx_codec needs to be added to the codec code of the platform.Take mtk-soc-codec-6357.c as an example.

The path is *kernel/sound/soc/mediatek/codec/mt6357/mtk-soc-codec-6357.c*.

Add external reference declaration for aw87xxx_codec registration function to *mtk-soc-codec-6357.c*.

```
#ifdef CONFIG_SND_SOC_AW87XXX
extern int aw87xxx_add_codec_controls(void *codec);
#endif /*CONFIG_SND_SOC_AWINIC_AW87XXX*/
```

Add the aw87xxx_codec registration call to the ***mt6357_codec_probe***.

```
#ifdef CONFIG_SND_SOC_AW87XXX
    ret = aw87xxx_add_codec_controls(codec);
    if (ret < 0) {
        pr_err("%s: aw87xxx_add_codec_controls failed, ret= %d\n",
           __func__, ret);
           return ret;
        };
#endif
```

2） Conguartion XML

Awinic provides customers with defauls profiles of ***Music , Receiver , Off.*** If you need other profiles, you can add profiles through aw87xxx_acf.bin configuration.Customers can synthesize profiles parameters in aw87xxx_acf.bin as required. Profiles in XML can also be configured based on profiles contained in ***aw87xxx_acf.bin***.

audio_device.xml is a mtk platform audio path management file that can turn on, turn off, and you can set different values for Kcontrols.

Here's the ***speaker_output*** path as an example

```
<!--speaker output-->
 <path name="speaker_output" value="turnon">
      <kctl name="aw87xxx_profile_switch_0" value="Music" />
      <kctl name="aw87xxx_profile_switch_1" value="Music" />
      …
 </path>
 <path name="speaker_output" value="turnoff">
      <kctl name="aw87xxx_profile_switch_0" value="Off" />
      <kctl name="aw87xxx_profile_switch_1" value="Off" />
      …
 </path>
```

3） Instructions on the use of Kcontrol

Scenario switching interface：

Awinic provides customers with a Kcontrol for switch profile. The interface name is ***aw87xxx_profile_switch_x*** (x is ***dev_index*** which was configured in dtsi or as the device registration sequence.). As a result, one Kontrol will be created for each registered device.

usage method：

tinymix aw87xxx_profile_switch_0 Displays the profiles which dev0 was successfully loaded and the current profile
tinymix aw87xxx_profile_switch_0 Music Switch the profile of dev0 to ***Music***
tinymix aw87xxx_profile_switch_1 Off Switch the profile of dev1 to ***Off***

Low voltage protection interface:
Awinic provides customers Kcontrol that can obtain real-time low-voltage protection adjustment parameters. It can be used with or without dsp. The Kcontrol's name is ***aw87xxx_vmax_get_x*** (x is ***dev_index*** of each PA).

usage method:
tinymix aw87xxx_vmax_get_0 Get the real-time power protection adjustment parameters of dev0.

For Examples:

```
k39tv1_bsp:/ # tinymix aw87xxx_vmax_get_0
aw87xxx_vmax_get_0: -7272660 (dsrange -2147483648->0)
k39tv1_bsp:/ # tinymix aw87xxx_vmax_get_1
aw87xxx_vmax_get_1: -7272660 (dsrange -2147483648->0)
[name:aw_monitor&][1-0058]aw_monitor_get_battery_capacity: The percentage is 1
[name:aw_monitor&][1-0058]aw_monitor_no_dsp_get_vmax: get_battery_capacity is[1]
[name:aw_monitor&][1-0058]aw_search_vmax_from_table: read setting vmax=0xff91072c, step[0]: vbat_min=0,vbat_max=40
[name:aw87xxx&][1-0058]aw87xxxx_vmax_get: get vmax = [0xff91072c]
[name:aw_monitor&][1-0059]aw_monitor_get_battery_capacity: The percentage is 1
[name:aw_monitor&][1-0059]aw_monitor_no_dsp_get_vmax: get_battery_capacity is[1]
[name:aw_monitor&][1-0059]aw_search_vmax_from_table: read setting vmax=0xff91072c, step[0]: vbat_min=0,vbat_max=40
[name:aw87xxx&][1-0059]aw87xxxx_vmax_get: get vmax = [0xff91072c]
```

3.2.2 Scheme 2: Add controls to the platform for the aw87xxx loading profiles

1) 4G platform porting scheme

Here's the *mtk-soc-codec-6357.c* file as an example, the path is *kernel/sound/soc/mediatek/codec/mt6357/mtk-soc-codec-6357.c*. Add the aw87xxx_profile load function to its profile control function.
Add aw87xxx profile switching external function declaration and variable definition in *mtk-soc-codec-6357.c*.

```
@@ -80,6 +80,17 @@ static void setDlMtkifSrc(bool enable);
 #ifndef ANALOG_HPTRIM
 static int SetDcCompenSation(bool enable);
 #endif
+#ifdef CONFIG_SND_SOC_AW87XXX
+extern int aw87xxx_set_profile(int dev_index, char *profile);
+
+static char *aw_profile[] = {"Music", "Off"};
+enum aw87xxx_dev_index {
+      AW_DEV_0 = 0,
+};
+#endif
 static void Voice_Amp_Change(bool enable);
```

Note: Awinic provides customers with defauls profiles of ***Music , Receiver , Off.*** If you need other profiles, you can add profiles through aw87xxx_acf.bin configuration. The profiles string in the array variable name of ***aw_profile*** needs consistent with the configuration in awinic_ACF_tool. The default parameter configuration allows switching profiles to ***Music, Receiver and Off***.

Add aw87xxx profile loading function to the specified scene control function:

```
@@ -3296,6 +3347,9 @@ static void Speaker_Amp_Change(bool enable)
       Ana_Set_Reg(AUDDEC_ANA_CON6, 0x0201, 0xffff);
       /* Switch LOL MUX to audio DAC */
       Ana_Set_Reg(AUDDEC_ANA_CON4, 0x011b, 0xffff);
```



```
+#ifdef CONFIG_SND_SOC_AW87XXX
+      aw87xxx_set_profile(AW_DEV_0, aw_profile[0]);
+#endif
       /* disable Pull-down HPL/R to AVSS28_AUD */
       if (mIsNeedPullDown)
           hp_pull_down(false);
@@ -3333,6 +3387,10 @@ static void Speaker_Amp_Change(bool enable)
           Ana_Set_Reg(AUDDEC_ANA_CON12, 0x0, 0x1055);
           /* Disable NCP */
           Ana_Set_Reg(AUDNCP_CLKDIV_CON3, 0x1, 0x1);
+
+#ifdef CONFIG_SND_SOC_AW87XXX
+      aw87xxx_set_profile(AW_DEV_0, aw_profile[1]);
+#endif
           TurnOffDacPower();
       }
   }
```

Note: the above added function calls take single PA and single profile as an example. If you need to realize multiple PAs or multiple profile control, please refer to the above code to add corresponding variables or function calls.

2） 5G platform widget control PA addition scheme

```
   @@ -17,6 +17,13 @@
 #include "../../codecs/mt6359.h"
 #include "../common/mtk-sp-spk-amp.h"

+#ifdef CONFIG_SND_SOC_AW87XXX
+extern int aw87xxx_set_profile(int dev_index, char *profile);
+static char *aw_profile[] = {"Music", "Off"};
+enum aw87xxx_dev_index {
+     AW_DEV_0 = 0,
+};
+#endif

 /*
  * if need additional control for the ext spk amp that is connected
@@ -92,9 +99,25 @@ static int mt6853_mt6359_spk_amp_event(s
   switch (event) {
   case SND_SOC_DAPM_POST_PMU:
       /* spk amp on control */
+#ifdef CONFIG_SND_SOC_AW87XXX
+      ret = aw87xxx_set_profile(AW_DEV_0, aw_profile[0]);
+       if (ret < 0) {
+          pr_err("[Awinic] %s: set profile[%s] failed",
+                 __func__, aw_profile[0]);
+          return ret;
+       }
+#endif
       break;
   case SND_SOC_DAPM_PRE_PMD:
       /* spk amp off control */
+#ifdef CONFIG_SND_SOC_AW87XXX
+       ret = aw87xxx_set_profile(AW_DEV_0, aw_profile[1]);
+       if (ret < 0) {
+          pr_err("[Awinic] %s: set profile[%s] failed",
+                 __func__, aw_profile[1]);
+          return ret;
```



```
+       }
+#endif
        break;
    default:
        break;
```

Note：Awinic provides customers with defauls profiles of ***Music , Receiver , Off.*** If you need other profiles, you can add profiles through aw87xxx_acf.bin configuration**.** The profiles string in the array variable name of aw_profile needs consistent with the configuration in ***awinic_ACF_tool***. The default parameter configuration allows switching profiles to ***Music, Receiver and Off***.

The above added function calls take single PA and single profile as an example. If you need to realize multiple PAs or multiple profile control, please refer to the above code to add corresponding variables or function calls.

4. Debug Interface

4.1 attributes nodes

AW87XXX driver will create different device nodes. The path is *sys/bus/i2c/driver/aw87xxx_pa/*-00xx*.

The driver will create four device nodes under each device,there are ***reg, profile, hwen*** and ***esd_enable***, where ******* is I2C bus number and ***xx*** is i2c address.

1） reg

Node name	**reg**
Function description	Read and write all register value of aw87xxx
Usage method	Read register value：cat reg Write register：echo reg_addr reg_data > reg （Hexadecimal operation）
Reference routine	cat reg （Get all the values of the register with read permission） echo 0x01 0x07 > reg （write the value of 0x07 to the register of 0x01）

2） profile

Node name	**profile**
Function description	Used to switch profiles
Usage method	Used to switch profiles：cat profile Set profile：echo profile_name > profile
Reference routine	cat profile （View the current profile and switchable profiles） echo “Music” > profile （Load ***Music*** profile） echo “Off” > profile （Load ***Off*** profile (PA power down)）

1） hwen

Node name	**hwen**
Function description	Used to control the hardware power on or power down
Usage method	cat hwen （Get aw87xxx hardware status） echo 1 > hwen （Hardware power on） echo 0 > hwen （Hardware power down）

2） **esd_enable**

Node name	**esd_enable**
Function description	ESD function switch
Usage method	Read：cat esd_enable Write：echo is_enable > esd_enable （String operation）
Reference routine	cat esd_enable （Get status of ESD function） echo true > esd_enable （Enable the ESD function） echo false > esd_enable （Disable the ESD function）

5. Low battery protection algorithm

Note: The following links are for customers who need low-power protection algorithm. If they do not need it, they may not pay attention to it:

According to whether the platform contains open dsp, two schemes are designed: 1. No dsp low-power protection algorithm; 2. With dsp low power protection algorithm.

5.1 Configuration

For no_dsp and with_dsp needs to be configured with config. Please refer to “aw87xxx_ monitor_ bin_ guide. pdf ", the difference is for no_dsp config configured monitor_ switch/monitor_ count/monitor_ time is invalid. You can configure it according to the instructions.

5.2 No DSP low power protection algorithm

If the platform does not contain open dsp, please refer to the algorithm migration document of “awinic_skt_mtk_porting.pdf”.Through the registered kcontrol interface ***aw87xxx_ vmax_ get_ x*** (x is ***dev_index*** of PA) to obtain the real-time low-power protection adjustment value.

Note: the no dsp low-power protection algorithm is compatible with the version before v2.0.0, and the real-time low-power adjustment value can be obtained through the Vmax node. Therefore, before using the V2.0.0 version of the algorithm, the user needs to have read permission on Vmax node

5.2.1 Debug Interface

Aw87xxx driver for no dsp platform will create two device nodes to low power protection under the device, there are vmax and vbat. The path is *sys/bus/i2c/driver/aw87xxx_pa/*-00xx*. where ***** is I2C bus number and ***xx*** is i2c address.

1）**vmax**

Node name	**vmax**
Function description	Used to found real-time power vmax value in monitor_bin
Usage method	cat vmax （Get the value of the current vmax）

1）**vbat**

Node name	**vbat**
Function description	It is used to debug and set the current power and get the real-time power value
Usage method	cat vbat （Get real-time power value）

	echo capacity > vbat	（Set vbat charge value）
	echo 0 > vbat	（Cancel the commissioning power input before）
Reference routine	echo 50 > vbat	（Set the current power to 50%）

5.3 With DSP low power protection algorithm

If the platform has an open dsp, please go to aw_dsp.h and add macro definition related to mtk platform dsp.

```
/*#define AW_MTK_OPEN_DSP_PLATFORM*/
```

5.3.1 Debug Interface

Aw87xxx driver for with_dsp platform will create six device nodes to low power protection under the device, there are ***vmax , vbat, monitor_switch, switch_count, monitor_time*** and ***rx***. The path is *sys/bus/i2c/driver/aw87xxx_pa/*-00xx*. where ***** is I2C bus number and ***xx*** is i2c address

1） vmax

Node name	**vmax**	
Function description	It is used to set vmax to dsp and get the current vmax value of dsp (The gets vmax is the vmax value calculated in the algorithm, which is different from the set value)	
Usage method	cat vmax	（Get the value of the current vmax）
	echo N > vmax	（Send calculated vmax value）
Reference routine	echo 0xfff95f7e > vmax	（Send 0xfff95f7e value to dsp）

2） vbat

Node name	**vbat**	
Function description	It is used to set the current power and get the real-time power value	
Usage method	cat vbat	（Get real-time power value）
	echo capacity > vbat	（Set vbat charge value）
	echo 0 > vbat	（Cancel the commissioning power input before）
Reference routine	echo 50 > vbat	（Set the current power to 50%）

3） monitor_switch

Node name	**monitor_switch**	
Function description	It is used to set aw87xxx to enable the automatic protection function with dsp	
Usage method	cat monitor_switch	（Get the protection enable status of the current power）
	echo 0 > monitor_switch	（Turn off automatic protection）
	echo 1 > monitor_switch	（Turn on automatic protection）

4） monitor_count

Node name	**monitor_count**
Function description	It is used to set time for AW87XXX monitor to get power value before setting vmax to the dsp

Usage method	cat monitor_count （Get the current power acquisition times of the system） echo 5 > monitor_count （Set the average number of power acquisition to 5）

5） monitor_time

Node name	**monitor_time**
Function description	Used to set the time interval for getting power value
Usage method	cat monitor_time （Get the monitor loop interval (ms)） echo N > monitor_time （Set the monitor loop interval (ms)）

6） rx

Node name	**rx**
Function description	Used to set and set the status of dsp RX module
Usage method	cat rx （Get the status of RX module of dsp） echo 1 > rx （Set RX module enable of dsp） echo 0 > rx （Set RX module of dsp not enabled）

