

AW882XX Android Driver(Qcom)

Version:

V3.3

Date:

June, 2024

contents

AW882XX ANDROID DRIVER (QCOM)	3
1. INFORMATION	3
2. RX DRIVER PORTING	3
2.1 AW882XX DRIVER PORTING	3
2.2 PLATFORM DRIVER CONFIGURATION	6
2.3 PLATFORM ROUTES CONFIGURATION	8
2.4 RX DRIVER PORTING VERIFICATION	9
3. ALGORITHM-PORTING	10
3.1 ALGORITHM AUTHENTICATION FUNCTION	10
4. TX DRIVER PORTING	10
4.1 PLATFORM ROUTES CONFIGURATION	10
4.2 HAL LAYER PORTING	13
4.3 TX DRIVER PORTING VERIFICATION	17
5. CALIBRATION	17
5.1 CALIBRATION PURPOSES	17
5.2 CALIBRATION ADAPTATION	17
5.3 CALIBRATION METHOD	18
5.4 VERIFICATION OF CALIBRATION RESULTS	21
5.5 CALIBRATION EXAMPLE CODE	21
6. DEBUG INTERFACE	21
6.1 NODE	21
6.2 KCONTROL	23
7. APPENDIX	24
7.1 ABOUT CALIBRATION	24
7.2 SOLUTION OF TX_SWITCH INVALID (APR COMMUNICATION FAILED)	24
7.3 KO COMPILE CONFIGURATION	25
7.4 PLATFORM I2C BUS DYNAMIC CHANGES	30

AW882XX Android Driver (Qcom)

1. INFORMATION

Driver File	aw882xx_calib.c, aw882xx_calib.h, aw882xx_monitor.c, aw882xx_monitor.h, aw882xx.c, aw882xx.h, aw882xx_device.c, aw882xx_device.h, aw882xx_dsp.c, aw882xx_dsp.h, aw882xx_init.c, aw882xx_init.h, aw882xx_log.h, aw882xx_spin.c, aw882xx_spin.h, aw882xx_data_type.h, aw882xx_bin_parse.c, aw882xx_bin_parse.h
Smart PA(with IV)	aw88257, aw88258, aw88261, aw88261s, aw88262, aw88263, aw88263h, aw88263s, aw88264, aw88265, aw88266, aw88266s, aw88270, aw88274, aw88299, aw88461, aw88271
Smart PA(no IV)	aw88298, aw88298g, aw88266a, aw88230, aw88082, aw88252 (No calibration function, please ignore the integration in Chapter 3/4/5)
I ² C Address	0x30/0x31/0x32/0x33/0x34/0x35/0x36/0x37

2. RX DRIVER PORTING

2.1 AW882XX DRIVER PORTING

2.1.1 DTS Configuration

Single PA Configuration

```
diff --git a/arch/arm/boot/dts/qcom/apq8096-dragonboard.dtsi
b/arch/arm/boot/dts/qcom/apq8096-dragonboard.dtsi
index f22db2e..a340a32 100644
--- a/arch/arm/boot/dts/qcom/apq8096-dragonboard.dtsi
+++ b/arch/arm/boot/dts/qcom/apq8096-dragonboard.dtsi
@@ -549,6 +549,8 @@
     i2c_x {                                /*x means the i2c bus number*/
+
+        /* AWINIC AW882XX mono Smart PA */
+        aw882xx_smartpa@34 {
+            compatible = "awinic,aw882xx_smartpa";
+            reg = <0x34>;
+            /* You cannot configure reset-gpio for aw88230, aw88257, aw88261,
aw88082 or aw88265*/
+            reset-gpio = <&tlmm 84 0>;
+            irq-gpio = <&tlmm 136 0x2008>;
+            sync-load = <1>; /* Firmware loading mode, qcom platform uses
synchronous loading*/
+            aw-tx-topo-id = <0x1000ff00>;
+            aw-rx-topo-id = <0x1000ff01>;
+            /*Set this parameter based on the port id of the I2S TX in use*/
+            aw-tx-port-id = <0x1007>;
```

```
+      /*Set this parameter based on the port id of the I2S RX in use*/
+      aw-rx-port-id = <0x1006>;
+      aw-re-min = <4000>;          /*Minimum calibration value (mohms)*/
+      aw-re-max= <30000>;         /*Maximum calibration value (mohms)*/
+      /*You cannot configure aw-cali-mode for products with IV*/
+      aw-cali-mode = "none";
+      status = "okay";
+  };
+  /* AWINIC AW882XX mono Smart PA End */
+  /*re means resistance of speaker */
```

Multiple PA Configuration

```
diff --git a/arch/arm/boot/dts/qcom/apq8096-dragonboard.dtsi
b/arch/arm/boot/dts/qcom/apq8096-dragonboard.dtsi
index f22db2e..a340a32 100644
--- a/arch/arm/boot/dts/qcom/apq8096-dragonboard.dtsi
+++ b/arch/arm/boot/dts/qcom/apq8096-dragonboard.dtsi
@@ -549,6 +549,8 @@
     i2c_x {                                /*x means the i2c bus number*/
+
+      /* AWINIC AW882XX Smart PA */
+      aw882xx_smartpa@34 {
+        compatible = "awinic,aw882xx_smartpa";
+        reg = <0x34>;
+        reset-gpio = <&tlmm 84 0>;
+        irq-gpio = <&tlmm 136 0x2008>;
+        sync-load = <1>;
+        sound-channel = <0>;                /*0:pri_l 1:pri_r 2:sec_l 3:sec_r*/
+        aw-tx-topo-id = <0x1000ff00>;
+        aw-rx-topo-id = <0x1000ff01>;
+        aw-tx-port-id = <0x1007>;
+        aw-rx-port-id = <0x1006>;
+        aw-re-min = <4000>;
+        aw-re-max= <30000>;
+        /*aw-cali-mode = "none";*/
+        status = "okay";
+    };
+
+    aw882xx_smartpa@35 {
+        compatible = "awinic,aw882xx_smartpa";
+        reg = <0x35>;
+        reset-gpio = <&tlmm 82 0>;
+        irq-gpio = <&tlmm 143 0x2008>;
+        sync-load = <1>;
+        sound-channel = <1>;                /*0:pri_l 1:pri_r 2:sec_l 3:sec_r*/
+        aw-tx-topo-id = <0x1000ff00>;
+        aw-rx-topo-id = <0x1000ff01>;
+        aw-tx-port-id = <0x1007>;
+        aw-rx-port-id = <0x1006>;
+        aw-re-min = <4000>;
+        aw-re-max= <30000>;
+        /*aw-cali-mode = "none";*/
+        status = "okay";
+    };
+
+    /* AWINIC AW882XX Smart PA End */
```

2.1.2 Driver Configuration

Qualcomm driver has two compilation modes: KO and build-in kernel. Select the corresponding compilation configuration scheme according to the platform loading mode.

KO Compile Configuration

See Appendix **KO COMPILE CONFIGURATION.**

Kernel Compile Configuration

Defconfig Configuration

```
#add aw882xx smartpa
CONFIG_SND_SMARTPA_AW882XX=y
```

Create aw882xx directory in the kernel codecs directory and add driver files

```
aw882xx_calib.c, aw882xx_calib.h, aw882xx_monitor.c, aw882xx_monitor.h,
aw882xx.c, aw882xx.h, aw882xx_device.c, aw882xx_device.h, aw882xx_dsp.c,
aw882xx_dsp.h, aw882xx_init.c, aw882xx_init.h, aw882xx_log.h,
aw882xx_spin.c, aw882xx_spin.h, aw882xx_data_type.h, aw882xx_bin_parse.c,
aw882xx_bin_parse.h
```

Kconfig Configuration

```
config SND_SMARTPA_AW882XX
    tristate "SoC Audio for awinic aw882xxseries"
    depends on I2C
    help
        This option enables support for aw882xxseries Smart PA.
```

Makefile Configuration

```
#for AWINIC AW882XX Smart PA
obj-$(CONFIG_SND_SMARTPA_AW882XX) += aw882xx/aw882xx.o
aw882xx/aw882xx_monitor.o aw882xx/aw882xx_init.o aw882xx/aw882xx_dsp.o
aw882xx/aw882xx_device.o aw882xx/aw882xx_calib.o
aw882xx/aw882xx_bin_parse.o aw882xx/aw882xx_spin.o
```

2.1.3 Add Communication Function With ADSP

Add Afe Communication Interfaces

Please refer to patch to modify q6afe.c:

```
AW882XX_Driver_QCOM_v1.10.0
├── ap
│   └── kernel
│       ├── 0001-Project-AW882XX_COPP.patch
│       ├── 0001-Project-AW882XX_COPP_V2.patch
│       ├── 0001-Project-AW882XX_driver.patch
│       └── 0001-Project-AW882XX_driver_params_v2.patch
```

/*Driver root directory*/

/*q6afe.c V3 patch*/

/*q6afe.c V2 patch*/

Add COPP Control Interface (Default Not Included)

Please refer to patch to add COPP control interface:

```
AW882XX_Driver_QCOM_v1.10.0                                /*Driver root directory*/
├── ap
│   └── kernel
│       ├── 0001-Project-AW882XX_COPP.patch                /*COPP interface V3 patch*/
│       ├── 0001-Project-AW882XX_COPP_V2.patch            /*COPP interface V2 patch*/
│       ├── 0001-Project-AW882XX_driver.patch
│       └── 0001-Project-AW882XX_driver_params_v2.patch
```

2.1.4 Bin Configuration

PA needs to configure register parameters to work normally. The configuration steps of PA bin file are as follows:

Compile Configuration

Add the bin file compilation option at the corresponding location of the platform

```
PRODUCT_COPY_FILES += \
hardware/qcom/audio/configs/xxx/aw882xx_acf.bin:$(TARGET_COPY_OUT_VENDOR)/firmware/aw882xx_acf.
bin
```

/*xxx means Platform path*/

Path Configuration

Add the directory of bin file in the firmware_class.c, the general directory is vendor / firmware

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

(PS: In the debugging stage, you can directly push the bin file to /vendor/ firmware/)

Bin Selection

Select the bin file in the config directory of the driver package based on the platform signal output format, PA quantity, and product name.

2.2 PLATFORM DRIVER CONFIGURATION

2.2.1 DAI_LINK Configuration

Different platform dai_link configuration is as follows:

Kernel Versions Prior To 5.4

1) Add awinic_codecs array

Single PA Configuration

```
diff --git a/sound/soc/msm/msm8996.c b/sound/soc/msm/msm8996.c
index ab04888..b5e4bf5 100644
--- a/sound/soc/msm/msm8996.c      /*Take the MSM8996 platform for example*/
+++ b/sound/soc/msm/msm8996.c
+struct snd_soc_dai_link_component awinic_codecs[] = {
+ {
+     .of_node = NULL,
+     .dai_name = "aw882xx-aif-6-34",      /*I2C BUS:6, I2C Address:34*/
+
+     .name = "aw882xx_smartpa.6-0034",
+ },
+}
```

Multiple PA Configuration

```
diff --git a/sound/soc/msm/msm8996.c b/sound/soc/msm/msm8996.c
index ab04888..b5e4bf5 100644
--- a/sound/soc/msm/msm8996.c
+++ b/sound/soc/msm/msm8996.c
+struct snd_soc_dai_link_component awinic_codecs[] = {
+ {
+     .of_node = NULL,
+     .dai_name = "aw882xx-aif-6-34",
+     .name = "aw882xx_smartpa.6-0034",
+ },
+ {
+     .of_node = NULL,
+     .dai_name = "aw882xx-aif-6-35",
+     .name = "aw882xx_smartpa.6-0035",
+ },
+}
```

2) Modify DAI interface

```
@@ -3775,8 +3777,10 @@ static struct snd_soc_dai_link
msm8996_common_be_dai_links[] = {
{
    .name = LPASS_BE_QUAT_MI2S_RX,
    /* Take the Quaternary MI2S interface for example*/
    .stream_name = "Quaternary MI2S Playback",
    .cpu_dai_name = "msm-dai-q6-mi2s.3",
    .platform_name = "msm-pcm-routing",
+#ifdef CONFIG_SND_SMARTPA_AW882XX
+    .num_codecs = ARRAY_SIZE(awinic_codecs),
+    .codecs = awinic_codecs,
+#else
    .codec_name = "msm-stub-codec.1",
    .codec_dai_name = "msm-stub-rx",
+#endif
    .no_pcm = 1,
    .dpcm_playback = 1,
    .be_id = MSM_BACKEND_DAI_QUATERNARY_MI2S_RX,
    .be_hw_params_fixup = msm_quat_mi2s_rx_be_hw_params_fixup,
```

```
.ops = &msm8996_quat_mi2s_be_ops,
.ignore_suspend = 1,
},
```

Kernel 5.4 And Later

Single PA Configuration

```
/*Take the PRI MI2S RX interface as an example*/
SND_SOC_DAILINK_DEFS(pri_mi2s_rx,
    DAILINK_COMP_ARRAY(COMP_CPU("msm-dai-q6-mi2s.0")),
    /*I2C BUS:6, I2C Address:34*/
    DAILINK_COMP_ARRAY(COMP_CODEC("aw882xx_smartpa.6-0034", "aw882xx-aif-6-34")),
    DAILINK_COMP_ARRAY(COMP_PLATFORM("msm-pcm-routing")));
```

Multiple PA Configuration

```
SND_SOC_DAILINK_DEFS(pri_mi2s_rx,
    DAILINK_COMP_ARRAY(COMP_CPU("msm-dai-q6-mi2s.0")),
    DAILINK_COMP_ARRAY(COMP_CODEC("aw882xx_smartpa.6-0034", "aw882xx-aif-6-34")),
    COMP_CODEC("aw882xx_smartpa.6-0035", "aw882xx-aif-6-35")),
    DAILINK_COMP_ARRAY(COMP_PLATFORM("msm-pcm-routing")));
```

2.3 PLATFORM ROUTES CONFIGURATION

2.3.1 XML Configuration

Mixer_paths_xxx.xml Configuration

1) Add the initialization controller

```
@@ -334,6 +335,18 @@
    <ctl name="SLIMBUS_0_RX Audio Mixer MultiMedia4" value="0" />
    <ctl name="SLIMBUS_6_RX Port Mixer AUX_PCM_UL_TX" value="0" />
    <ctl name="HDMI Mixer MultiMedia4" value="0" />
+    <!-- quat start -->
+    <!-- Take the QUAT MI2S RX interface for example -->
+    <ctl name="QUAT_MI2S_RX Audio Mixer MultiMedia1" value="0" />
+    <ctl name="QUAT_MI2S_RX Audio Mixer MultiMedia2" value="0" />
+    <ctl name="QUAT_MI2S_RX Audio Mixer MultiMedia3" value="0" />
+    <ctl name="QUAT_MI2S_RX Audio Mixer MultiMedia4" value="0" />
+    <ctl name="QUAT_MI2S_RX Audio Mixer MultiMedia5" value="0" />
+    <ctl name="QUAT_MI2S_RX Audio Mixer MultiMedia6" value="0" />
+    <ctl name="QUAT_MI2S_RX Audio Mixer MultiMedia7" value="0" />
+    <ctl name="QUAT_MI2S_RX Audio Mixer MultiMedia8" value="0" />
+    <ctl name="QUAT_MI2S_RX Audio Mixer MultiMedia9" value="0" />
+    <ctl name="QUAT_MI2S_RX Audio Mixer MultiMedia10" value="0" />
    <!-- HFP start -->
    <ctl name="HFP_PRI_AUX_UL_HL Switch" value="0" />
    <ctl name="SLIMBUS_0_RX Port Mixer AUX_PCM_UL_TX" value="0" />
```

2) Modify all usecase output interfaces, using the deep-buffer-playback speaker as an example

```
@@ -608,7 +621,8 @@
    </path>

    <path name="deep-buffer-playback speaker">
-        <ctl name="SLIMBUS_0_RX Audio Mixer MultiMedia1" value="1" />
```



```
+ <!--ctl name="SLIMBUS_0_RX Audio Mixer MultiMedial" value="1" /-->
+ <ctl name="QUAT_MI2S_RX Audio Mixer MultiMedial" value="1" />
</path>
```

3) Invalidate the content in path name="speaker"

```
@@ -1884,7 +1901,7 @@
</path>

<path name="speaker">
- <path name="speaker-lineout" />
+ <!--path name="speaker-lineout" /-->
</path>
```

Audio_platform_info_xxx.xml Configuration

All devices in use need to be added. Take SND_DEVICE_OUT_SPEAKER as an example

```
diff --git a/configs/msm8996/audio_platform_info.xml
b/configs/msm8996/audio_platform_info.xml
index e8fecec..22c40c2 100755
--- a/configs/msm8996/audio_platform_info.xml
+++ b/configs/msm8996/audio_platform_info.xml
@@ -47,6 +47,8 @@
<param key="input_mic_max_count" value="4"/>
</config_params>
<backend_names>
+ <device name="SND_DEVICE_OUT_SPEAKER" interface="QUAT_MI2S_RX">
  <device name="SND_DEVICE_OUT_HEADPHONES" backend="headphones"
interface="SLIMBUS_6_RX"/>
  <device name="SND_DEVICE_OUT_LINE" backend="headphones"
interface="SLIMBUS_6_RX"/>
  <device name="SND_DEVICE_OUT_ANC_HEADSET" backend="headphones"
interface="SLIMBUS_6_RX"/>

/*Take the QUAT MI2S RX interface as an example*/
```

2.4 RX DRIVER PORTING VERIFICATION

The above operations have completed the rx driver integration. Confirm that is effective through the following driver log:

I2C communication succeeded:

```
[Awinic][6-0034]aw882xx_dai_drv_append_suffix: dai name [aw882xx-aif-6-34]
[Awinic][6-0034]aw882xx_dai_drv_append_suffix: pstream_name name [Speaker_Playback-6-34]
[Awinic][6-0034]aw882xx_dai_drv_append_suffix: cstream_name name [Speaker_Capture-6-34]
[Awinic][6-0034]aw882xx_i2c_probe: dev_cnt 1
```

Sound card registration succeeded:

```
[Awinic][6-0034]aw882xx_codec_probe: enter
[Awinic][6-0034]aw882xx_add_codec_controls: enter
[Awinic][6-0034]aw882xx_request_firmware: load [aw882xx_acf.bin] , file size: [2016]
[Awinic]aw_dev_parse_check_acf_by_hdr: project name [A2113]
[Awinic]aw_dev_parse_check_acf_by_hdr: custom name [Awinic]
```

PA Bin loaded successfully:

```
[Awinic] [6-0034] aw_monitor_parse_vol_data_v0_1_1: ==parse vol end ==
[Awinic] [6-0034] aw_dev_parse_skt_type: enter
[Awinic] [6-0034] aw_dev_parse_skt_type: get dsp data prof cnt is 0
[Awinic] [6-0034] aw_dev_parse_get_vaild_prof: get vaild profile:2
[Awinic] [6-0034] aw_dev_parse_acf: parse cfg success
[Awinic] [6-0034] aw_dev_soft_reset: soft reset done
[Awinic] [6-0034] aw_dev_reg_fw_update: amppd_st=0x0000
```

Play music and PA makes sound:

```
[Awinic] [6-0034] aw_dev_set_intmask: done
[Awinic] [6-0034] aw_monitor_start: enter
[Awinic] [6-0034] aw_check_bop_status: enter
[Awinic] [6-0034] aw_check_bop_status: check done! bop status is 0
[Awinic] [6-0034] aw_device_start: done
[Awinic] [6-0034] aw882xx_start_pa: start success
```

3. ALGORITHM-PORTING

Please refer to the algorithm integration document provided by Awinic for integration.

3.1 ALGORITHM AUTHENTICATION FUNCTION

If there is regular white noise playback during PA operation (audio source 10s ->white noise 3s->audio source 10s ->white noise 3s->.....)

The above phenomenon indicates that the awinic algorithm authentication has failed. Please enable the corresponding function in the driver to recompile.

```
diff --git a/aw882xx_dsp.h b/aw882xx_dsp.h
index 998fa55..bce2a75 100644
--- a/aw882xx_dsp.h
+++ b/aw882xx_dsp.h
@@ -19,7 +19,7 @@
-/*#define AW_ALGO_AUTH_DSP*/
+#define AW_ALGO_AUTH_DSP

/*factor form 12bit(4096) to 1000*/
#define AW_DSP_RE_TO_SHOW_RE(re) (((re) * (1000)) >> (12))
```

4. TX DRIVER PORTING

4.1 PLATFORM ROUTES CONFIGURATION

4.1.1 TX Routes Configuration

Add QUAT MI2S TX interface. The following uses QUAT MI2S as an example

```
diff --git a/sound/soc/msm/qdsp6v2/msm-pcm-routing-v2.c
b/sound/soc/msm/qdsp6v2/msm-pcm-routing-v2.c
index ed7d4ed..1cabd90 100644
--- a/sound/soc/msm/qdsp6v2/msm-pcm-routing-v2.c
+++ b/sound/soc/msm/qdsp6v2/msm-pcm-routing-v2.c
```

```

@@ -10225,7 +10225,9 @@ static const char * const
slim0_rx_vi_fb_tx_rch_mux_text[] = {
    static const char * const mi2s_rx_vi_fb_tx_mux_text[] = {
        "ZERO", "SENARY_TX"
    };
-
+static const char * const quat_mi2s_rx_vi_fb_tx_mux_text[] = {
+    "ZERO", "QUAT_MI2S_TX"
+};
static const int const slim0_rx_vi_fb_tx_lch_value[] = {
    MSM_BACKEND_DAI_MAX, MSM_BACKEND_DAI_SLIMBUS_4_TX
};
@@ -10238,6 +10240,10 @@ static const int const mi2s_rx_vi_fb_tx_value[] = {
    MSM_BACKEND_DAI_MAX, MSM_BACKEND_DAI_SENARY_MI2S_TX
};

+static const int const quat_mi2s_rx_vi_fb_tx_value[] = {
+    MSM_BACKEND_DAI_MAX, MSM_BACKEND_DAI_QUATERNARY_MI2S_TX
+};
+
static const struct soc_enum slim0_rx_vi_fb_lch_mux_enum =
    SOC_VALUE_ENUM_DOUBLE(0, MSM_BACKEND_DAI_SLIMBUS_0_RX, 0, 0,
        ARRAY_SIZE(slim0_rx_vi_fb_tx_lch_mux_text),
@@ -10253,6 +10259,11 @@ static const struct soc_enum mi2s_rx_vi_fb_mux_enum =
    ARRAY_SIZE(mi2s_rx_vi_fb_tx_mux_text),
    mi2s_rx_vi_fb_tx_mux_text, mi2s_rx_vi_fb_tx_value);

+static const struct soc_enum quat_mi2s_rx_vi_fb_mux_enum =
+    SOC_VALUE_ENUM_DOUBLE(0, MSM_BACKEND_DAI_QUATERNARY_MI2S_RX, 0, 0,
+    ARRAY_SIZE(quat_mi2s_rx_vi_fb_tx_mux_text),
+    quat_mi2s_rx_vi_fb_tx_mux_text, quat_mi2s_rx_vi_fb_tx_value);
+
static const struct snd_kcontrol_new slim0_rx_vi_fb_lch_mux =
    SOC_DAPM_ENUM_EXT("SLIM0_RX_VI_FB_LCH_MUX",
        slim0_rx_vi_fb_lch_mux_enum, spkr_prot_get_vi_lch_port,
@@ -10268,6 +10279,11 @@ static const struct snd_kcontrol_new mi2s_rx_vi_fb_mux
=
    mi2s_rx_vi_fb_mux_enum, spkr_prot_get_vi_lch_port,
    spkr_prot_put_vi_lch_port);

+static const struct snd_kcontrol_new quat_mi2s_rx_vi_fb_mux =
+    SOC_DAPM_ENUM_EXT("QUAT_MI2S_RX_VI_FB_MUX",
+    quat_mi2s_rx_vi_fb_mux_enum, spkr_prot_get_vi_lch_port,
+    spkr_prot_put_vi_lch_port);
+
static const struct snd_soc_dapm_widget msm_qdsp6_widgets[] = {
    /* Frontend AIF */
    /* Widget name equals to Front-End DAI name<Need confirmation>,
@@ -11242,7 +11258,8 @@ static const struct snd_soc_dapm_widget
msm_qdsp6_widgets[] = {
        &slim0_rx_vi_fb_rch_mux),
    SND_SOC_DAPM_MUX("PRI_MI2S_RX_VI_FB_MUX", SND_SOC_NOPM, 0, 0,
        &mi2s_rx_vi_fb_mux),
-
+    SND_SOC_DAPM_MUX("QUAT_MI2S_RX_VI_FB_MUX", SND_SOC_NOPM, 0, 0,
+    &quat_mi2s_rx_vi_fb_mux),
    SND_SOC_DAPM_MUX("VOC_EXT_EC_MUX", SND_SOC_NOPM, 0, 0,
        &voc_ext_ec_mux),
    SND_SOC_DAPM_MUX("AUDIO_REF_EC_UL1_MUX", SND_SOC_NOPM, 0, 0,

```

```
@@ -13450,9 +13467,11 @@ static const struct snd_soc_dapm_route intercon[] = {
    {"SLIM0_RX_VI_FB_LCH_MUX", "SLIM4_TX", "SLIMBUS_4_TX"},
    {"SLIM0_RX_VI_FB_RCH_MUX", "SLIM4_TX", "SLIMBUS_4_TX"},
    {"PRI_MI2S_RX_VI_FB_MUX", "SENARY_TX", "SENARY_TX"},
+   {"QUAT_MI2S_RX_VI_FB_MUX", "QUAT_MI2S_TX", "QUAT_MI2S_TX"},
    {"SLIMBUS_0_RX", NULL, "SLIM0_RX_VI_FB_LCH_MUX"},
    {"SLIMBUS_0_RX", NULL, "SLIM0_RX_VI_FB_RCH_MUX"},
    {"PRI_MI2S_RX", NULL, "PRI_MI2S_RX_VI_FB_MUX"},
+   {"QUAT_MI2S_RX", NULL, "QUAT_MI2S_RX_VI_FB_MUX"},
    {"PRI_TDM_TX_0", NULL, "BE_IN"},
    {"PRI_TDM_TX_1", NULL, "BE_IN"},
    {"PRI_TDM_TX_2", NULL, "BE_IN"},
}
```

4.1.2 TX channel 配置

Set the number of channels for the TX module to 2(2 is required for both single PA and Multiple PA). The following example is for reference.

```
diff --git a/kernel/msm-4.4/sound/soc/msm/sdm660-common.c b/kernel/msm-4.4/sound/soc/msm/sdm660-common.c
index c30c42cf51..ae81c66530 100755
--- a/kernel/msm-4.4/sound/soc/msm/sdm660-common.c
+++ b/kernel/msm-4.4/sound/soc/msm/sdm660-common.c
@@ -236,7 +236,7 @@ static struct dev_config mi2s_rx_cfg[] = {

static struct dev_config mi2s_tx_cfg[] = {
    #ifdef CONFIG_SND_I2S_PRIMARY
-   [PRIM_MI2S] = {SAMPLING_RATE_48KHZ, SNDRV_PCM_FORMAT_S16_LE, 1},
+   [PRIM_MI2S] = {SAMPLING_RATE_48KHZ, SNDRV_PCM_FORMAT_S16_LE, 2},
    #else
    [PRIM_MI2S] = {SAMPLING_RATE_48KHZ, SNDRV_PCM_FORMAT_S16_LE, 1},
    #endif
}
```

4.1.3 FE DAI Configuration

Example for FE DAI configuration modification, can be used as a reference

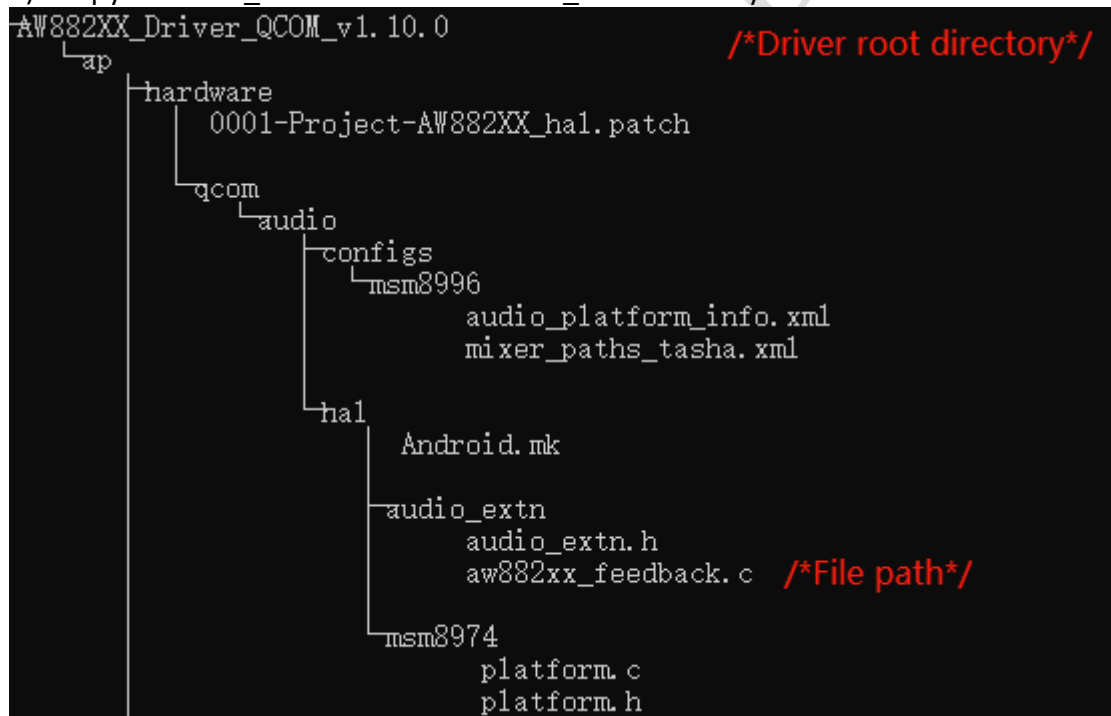
```
diff --git a/asoc/msm8996.c b/asoc/msm8996.c
index 71e01b9..9949b1f 100644
--- a/sound/soc/msm/msm8996.c
+++ b/sound/soc/msm/msm8996.c
@@ -2961,6 +3492,36 @@ static struct snd_soc_dai_link
msm8996_common_dai_links[] = {
    .codec_name = "snd-soc-dummy",
    .be_id = MSM_FRONTEND_DAI_VOICE2,
},
+ {
+   .name = "Quaternary MI2S RX_Hostless",
+   .stream_name = "Quaternary MI2S_RX Hostless Playback",
+   .cpu_dai_name = "QUAT_MI2S_RX_HOSTLESS",
+   .platform_name = "msm-pcm-hostless",
+   .dynamic = 1,
+   .dpcm_playback = 1,
+   .trigger = {SND_SOC_DPCM_TRIGGER_POST,
+               SND_SOC_DPCM_TRIGGER_POST},
+   .no_host_mode = SND_SOC_DAI_LINK_NO_HOST,
+   .ignore_suspend = 1,
}
```

```
+ .ignore_pmdown_time = 1,
+ .codec_dai_name = "snd-soc-dummy-dai",
+ .codec_name = "snd-soc-dummy",
+ },
+ {
+ .name = "Quaternary MI2S_TX_Hostless",
+ .stream_name = "Quaternary MI2S_TX Hostless Capture",
+ .cpu_dai_name = "QUAT_MI2S_TX_HOSTLESS",
+ .platform_name = "msm-pcm-hostless",
+ .dynamic = 1,
+ .dpcm_capture = 1,
+ .trigger = {SND_SOC_DPCM_TRIGGER_POST,
+           SND_SOC_DPCM_TRIGGER_POST},
+ .no_host_mode = SND_SOC_DAI_LINK_NO_HOST,
+ .ignore_suspend = 1,
+ .ignore_pmdown_time = 1,
+ .codec_dai_name = "snd-soc-dummy-dai",
+ .codec_name = "snd-soc-dummy",
+ },
+ },
};
```

4.2 HAL LAYER PORTING

4.2.1 HAL Code Porting

1) Copy aw882xx_feedback.c file to audio_extn directory:



2) Modify the Android.mk file

```
diff --git a/hal/Android.mk b/hal/Android.mk
index 94c21bd..5a963fa 100644
--- a/hal/Android.mk
+++ b/hal/Android.mk
```

```
@@ -79,6 +79,10 @@ LOCAL_C_INCLUDES +=
$(TARGET_OUT_INTERMEDIATES)/KERNEL_OBJ/usr/include
LOCAL_C_INCLUDES +=
$(TARGET_OUT_INTERMEDIATES)/KERNEL_OBJ/usr/techpack/audio/include
LOCAL_ADDITIONAL_DEPENDENCIES += $(TARGET_OUT_INTERMEDIATES)/KERNEL_OBJ/usr

+#awinic add
/*Only for Android R or later, you need to add*/
+#LOCAL_CFLAGS += -DANDROID_R
+LOCAL_CFLAGS += -DAWINIC_SMARTPA_ENABLE
+LOCAL_SRC_FILES += audio_extn/aw882xx_feedback.c
+
ifeq ($(strip $(AUDIO_FEATURE_ENABLED_DLKM)),true)
    LOCAL_HEADER_LIBRARIES += audio_kernel_headers
    LOCAL_C_INCLUDES += $(TARGET_OUT_INTERMEDIATES)/vendor/qcom/opensource/audio-
kernel/include
```

3) Modify audio_extn.h file, add Awinic function interface

```
diff --git a/hal/audio_extn/audio_extn.h b/hal/audio_extn/audio_extn.h
index d3e7a5f..6785c10 100644
--- a/hal/audio_extn/audio_extn.h
+++ b/hal/audio_extn/audio_extn.h
@@ -1106,4 +1105,13 @@ void audio_extn_ffv_check_and_append_ec_ref_dev(char
*device_name);
    snd_device_t audio_extn_ffv_get_capture_snd_device();
    void audio_extn_ffv_append_ec_ref_dev_name(char *device_name);
    #endif
+
+int audio_extn_aw882xx_start_feedback(struct audio_device *adev, snd_device_t
snd_device);
+void audio_extn_aw882xx_stop_feedback(struct audio_device *adev, snd_device_t
snd_device);
+
#endif /* AUDIO_EXTN_H */
```

4) Modify the supported device types based on customer requirements

```
diff --git a/platform.c b/platform.c
index 1b6b5e0..7f48c4b 100644
--- a/platform.c
+++ b/platform.c
@@ -7506,16 +7506,25 @@ platform_can_enable_spkr_prot_on_device(snd_device_t snd_device)
{
    bool ret = false;

+    ALOGV("%s [Awinic] %d\n", __func__, snd_device);
    if (snd_device == SND_DEVICE_OUT_SPEAKER ||
        snd_device == SND_DEVICE_OUT_SPEAKER_REVERSE ||
+    snd_device == SND_DEVICE_OUT_SPEAKER_WSA ||
        snd_device == SND_DEVICE_OUT_SPEAKER_VBAT ||
        snd_device == SND_DEVICE_OUT_VOICE_SPEAKER_VBAT ||
        snd_device == SND_DEVICE_OUT_VOICE_SPEAKER_2_VBAT ||
+    snd_device == SND_DEVICE_OUT_VOICE_SPEAKER_STEREO ||
        snd_device == SND_DEVICE_OUT_VOICE_SPEAKER ||
-        snd_device == SND_DEVICE_OUT_VOICE_SPEAKER_2) {
+    snd_device == SND_DEVICE_OUT_VOICE_SPEAKER_2 ||
```

```
+     snd_device == SND_DEVICE_OUT_VOICE_SPEAKER_WSA ||
+     snd_device == SND_DEVICE_OUT_VOICE_SPEAKER_2_WSA ||
+     snd_device == SND_DEVICE_OUT_VOIP_SPEAKER) {
+         ret = true;
+     }

+     if (ret) {
+         ALOGV("%s [Awinic] snd_device_out id %d is supported iv feedback\n",
+             __func__, snd_device);
+     } else {
+         ALOGV("%s [Awinic] unsupport snd_device_out id: %d\n",
+             __func__, snd_device);
+     }
+     return ret;
+ }
```

- 5) Added calls to audio_extn_aw882xx_start_feedback and audio_extn_aw882xx_stop_feedback functions

```
diff --git a/hal/audio_hw.c b/hal/audio_hw.c
index 3e3f72f..6574566 100644
--- a/hal/audio_hw.c
+++ b/hal/audio_hw.c
@@ -1096,7 +1096,8 @@ int enable_snd_device(struct audio_device *adev,
    }
    audio_extn_dev_arbi_acquire(snd_device);
    audio_route_apply_and_update_path(adev->audio_route, device_name);

-
+ /*Awinic Add*/
+ audio_extn_aw882xx_start_feedback( adev, snd_device );
    if (SND_DEVICE_OUT_HEADPHONES == snd_device &&
        !adev->native_playback_enabled &&
        audio_is_true_native_stream_active(adev)) {
@@ -1157,6 +1158,7 @@ int disable_snd_device(struct audio_device *adev,
        disable_snd_device(adev, new_snd_devices[i]);
    }
    } else {
+     audio_extn_aw882xx_stop_feedback( adev, snd_device);
        audio_route_reset_and_update_path(adev->audio_route, device_name);
    }
}
```

4.2.2 XML Configuration

mixer_paths_xxx.xml Configuration

- 1) Add IV feedback related control controls to the initialization list

```
@@ -179,6 +179,7 @@
    <ctl name="IIR1_INP2_MUX" value="ZERO" />
    <ctl name="SLIM0_RX_VI_FB_LCH_MUX" value="ZERO" />
    <ctl name="SLIM0_RX_VI_FB_RCH_MUX" value="ZERO" />
+   <ctl name="QUAT_MI2S_RX_VI_FB_MUX" value="ZERO" />
    <ctl name="VI_FEED_TX_Channels" value="Two" />
    <ctl name="AIF4_VI_Mixer_SPKR_VI_1" value="0" />
    <ctl name="AIF4_VI_Mixer_SPKR_VI_2" value="0" />
```

2) Change path name="spkr-vi-record" to the QUAT MI2S interface

```
@@ -1740,7 +1756,8 @@
    </path>

    <path name="spkr-vi-record">
-        <ctl name="SLIM0_RX_VI_FB_LCH_MUX" value="SLIM4_TX" />
+        <ctl name="QUAT_MI2S_RX_VI_FB_MUX" value="QUAT_MI2S_TX" />
+        <!--ctl name="SLIM0_RX_VI_FB_LCH_MUX" value="SLIM4_TX" /-->
    </path>
```

audio_platform_info_xxx.xml Configuration

1) Add QUAT MI2S configuration

```
diff --git a/configs/msm8996/audio_platform_info.xml
b/configs/msm8996/audio_platform_info.xml
index e8fecec..22c40c2 100755
--- a/configs/msm8996/audio_platform_info.xml
+++ b/configs/msm8996/audio_platform_info.xml
@@ -47,6 +47,8 @@
    <param key="input_mic_max_count" value="4"/>
</config_params>
<backend_names>
    <device name="SND_DEVICE_OUT_SPEAKER" interface="QUAT_MI2S_RX">
+    <device name="SND_DEVICE_IN_CAPTURE_VI_FEEDBACK"
interface="QUAT_MI2S_TX">
    <device name="SND_DEVICE_OUT_HEADPHONES" backend="headphones"
interface="SLIMBUS_6_RX"/>
    <device name="SND_DEVICE_OUT_LINE" backend="headphones"
interface="SLIMBUS_6_RX"/>
    <device name="SND_DEVICE_OUT_ANC_HEADSET" backend="headphones"
interface="SLIMBUS_6_RX"/>
```

2) Perform the following operations to configure the pcm id of USECASE_AUDIO_SPKR_CALIB_TX in xml

a. Confirm the pcm id as follows:

```
msm8996:/ #
msm8996:/ # cat /proc/asound/pcm /*executive command*/
00-00: MultiMedia1 (*) : : playback 1 : capture 1
00-01: MultiMedia2 (*) : : playback 1 : capture 1
00-02: VoiceMMModel (*) : : playback 1 : capture 1
00-03: VoIP (*) : : playback 1 : capture 1
00-04: MultiMedia3 (*) : : playback 1 : capture 1
00-05: SLIMBUS_0 Hostless (*) : : playback 1 : capture 1
00-06: Tertiary MI2S_TX Hostless Capture (*) : : capture 1
00-07: AFE-PROXY RX msm-stub-rx-7 : : playback 1
00-08: AFE-PROXY TX msm-stub-tx-8 : : capture 1
00-09: (Compress1) : : playback 1 : capture 1
00-10: AUXPCM Hostless (*) : : playback 1 : capture 1

00-38: (Compress8) : : playback 1
00-39: (Compress9) : : playback 1
00-40: CS-Voice (*) : : playback 1 : capture 1
00-41: Voice2 (*) : : playback 1 : capture 1
00-42: Quaternary MI2S_RX Hostless Playback (*) : : playback 1
00-43: Quaternary MI2S_TX Hostless Capture (*) : : capture 1 /*The pcm id is 43*/
00-44: Slimbus4 Capture tasha_vifedback-44 : : capture 1
```


b. Configure USECASE AUDIO SPKR CALIB TX:

```
diff --git a/audio_platform_info.xml b/audio_platform_info.xml
index 8923f07..a284d73 100644
--- a/audio_platform_info.xml
+++ b/audio_platform_info.xml
@@ -34,6 +34,11 @@
     <device name="SND_DEVICE_IN_UNPROCESSED_QAUD_MIC" acdb_id="146"/>
     <device name="SND_DEVICE_IN_UNPROCESSED_HEADSET_MIC" acdb_id="147"/>
 </acdb_ids>
 <pcm_ids>
   <usecase name="USECASE_AUDIO_RECORD_LOW_LATENCY" type="in" id="19" />
+  <!--Configure the actual pcm id-->
+  <usecase name="USECASE_AUDIO_SPKR_CALIB_TX" type="in" id="43" />
   <usecase name="USECASE_AUDIO_PLAYBACK_UCLL" type="out" id="19" />
 </pcm_ids>
 <bit_width_configs>
   <device name="SND_DEVICE_OUT_SPEAKER" bit_width="24"/>
 </bit_width_configs>
```

4.3 TX DRIVER PORTING VERIFICATION

Play music, locat captures logs, and verify the configuration is correct:

E.g. Pcm tx start success

```
12-05 11:26:55.604 464 2290 D audio_hw_primary: enable_audio_route: apply mixer and update
path: spkr-vi-record
12-05 11:26:55.604 464 2290 D audio_route: Apply path: spkr-vi-record
12-05 11:26:55.605 1048 1584 W UsageStatsService: Event reported without a package name
12-05 11:26:55.605 464 2290 D audio_hw_awinic_feedback:
audio_extn_aw882xx_start_feedback:[Awinic] the pcm id uc_info->id = 48, pcm_tx_id = 43
12-05 11:26:55.631 2275 2275 I m.android.emai: The ClassLoaderContext is a special shared
library.
```

(PS: pcm_tx_id=43 corresponding to stream_name = "Quaternary MI2S_TX Hostless Capture")

5. CALIBRATION

5.1 CALIBRATION PURPOSES

To meet speaker protection requirements, the AW882XX driver supports speaker calibration on the production line and writes the re value of a qualified speaker to the persist partition of the phone. When loading the chip configuration at boot, the driver will write the calibration value read in the persist partition into the protection algorithm to achieve the function of horn protection.

5.2 CALIBRATION ADAPTATION

5.2.1 Calibration File saving path adaptation

The path to save the calibration value file is defined in driver (aw882xx_calib.c) as follows:

```
#ifdef AW_CALI_STORE_EXAMPLE
/*write cali to persist file example*/
#define AWINIC_CALI_FILE "/mnt/vendor/persist/factory/audio/aw_cali.bin" /*Save path*/
#define AW_INT_DEC_DIGIT 10
```

Please confirm whether there is the same path in the mobile phone. If there is no corresponding path, the calibration will fail:

```
msm8996:/ # cd mnt/vendor/persist/factory/audio/
msm8996:/mnt/vendor/persist/factory/audio # pwd
/mnt/vendor/persist/factory/audio
msm8996:/mnt/vendor/persist/factory/audio # ls
aw_cali.bin
msm8996:/mnt/vendor/persist/factory/audio #
```

5.3 CALIBRATION METHOD

AW882XX driver provides misc, class and attr calibration methods.

5.3.1 Class Calibration

1) Node function

Nodes	Function
/sys/class/smartpa/cali_time	1.Set calibration time 2.Show current calibration time
/sys/class/smartpa/re25_calib	1.Calibrate re 2.Save calibration re value
/sys/class/smartpa/f0_calib	Calibrate f0
/sys/class/smartpa/re_show	Show re
/sys/class/smartpa/f0_show	Show f0
/sys/class/smartpa/re_range	Show the range of calibration re value

2) Calibration steps

- Play mute music:
- Start the calibration:

```
msm8996:/sys/class/smartpa #
msm8996:/sys/class/smartpa # cat re25_calib
pri_l:6428 m0hms pri_r:6280 m0hms
msm8996:/sys/class/smartpa #
```

- F0 calibration:

```
msm8996:/sys/class/smartpa #
msm8996:/sys/class/smartpa # cat f0_calib
pri_l:832 pri_r:980
msm8996:/sys/class/smartpa #
```

5.3.2 Misc Calibration

The misc calibration of AW882XX driver is realized through executable file. Follow the steps below:

1) Get the executable file

```
AW882XX_Driver_QCOM_v1.10.0      /*Driver root directory*/
├── ap
│   └── smartpa_cali
│       ├── aw882xx_cali_32      /*32bit system executable file*/
│       ├── aw882xx_cali_64      /*64bit system executable file*/
│       └── example_source_code
│           ├── aw882xx_cali_attr_multi_mode.c
│           ├── aw882xx_cali_attr_single_dev_mode.c
│           └── aw882xx_cali_class_example.c
```

2) Configure the executable file

```
C:\Users\AW882XX_Driver_MTK_v1.10.0\ap\smartpa_cali>adb push aw882xx_cali_32 /system/bin/
aw882xx_cali_32: 1 file pushed. 0.6 MB/s (34504 bytes in 0.052s)

C:\Users\AW882XX_Driver_MTK_v1.10.0\ap\smartpa_cali>adb shell chmod 0777 /system/bin/aw882xx_cali_32
```

3) Command introduction

```
Calibration executables version: v0.3.10

-----
./aw882xx_cali [dev_name] cmd [optional params]      /*command format*/
-----
./aw882xx_cali [dev_name] cali [cali_re_time(ms)]      /*calibration*/
-----
./aw882xx_cali [dev_name] cali_re [cali_re_time(ms)]      /*re calibration*/
-----
./aw882xx_cali [dev_name] cali_f0 [noise]      /*f0 calibration*/
-----
./aw882xx_cali [dev_name] get_spkr_status      /*get realtime re\te\f0*/
-----
./aw882xx_cali [dev_name] get_spkr_st      /*get realtime re\te*/
-----
./aw882xx_cali [dev_name] set_cali_re re_value1 [re_value2] /*set cali_re value*/
-----
./aw882xx_cali [dev_name] cali_f0_q [noise]      /*cali f0\q*/
-----
./aw882xx_cali [dev_name] cali_all [cali_re_time(ms)]      /*cali re\f0\q*/
-----
./aw882xx_cali [dev_name] get_re_range      /*get range of cali re value*/
-----
./aw882xx_cali dev_name set_params params_file      /*set params files to dsp*/
-----
```

Parameter explanation (Note: [] means this option can be left blank)

dev_name	This parameter is used to calibrate a single device. The dev_name used by each device corresponds to the sound-channel configured in the DTS. 0: aw882xx_smartpa_l 1: aw882xx_smartpa_r 2: aw882xx_smartpa_sec_l 3: aw882xx_smartpa_sec_r If this parameter is not specified, all devices are calibrated by default.
noise	Fill in the noise parameter to play white noise when calibrating F0; If the customer chooses to play white noise by itself, you do not need to set this parameter.
cali_re_time	The time used to set the calibration RE value, in ms, shall not be less than 1000ms;

	If this parameter is not specified, the default calibration time is 3000ms.
re_value1	The calibration resistance value (mohm) needs to be set to the system.
params_file	Path corresponding to algorithm parameters.

4) Calibration steps

- Play mute music;
- Start the calibration:

```
msm8996:/ # aw882xx_cali_32 start_cali
[aw882xx_smartpa_1]cali_RE = 6429
[aw882xx_smartpa_r]cali_RE = 6346
[aw882xx_smartpa_1]cali_f0 = 836
[aw882xx_smartpa_r]cali_f0 = 979
```

5.3.3 Attr Calibration

1) Node path

6-0034: “6” means I2C bus, “0034” meand I2C address:

```
msm8996:/sys/bus/i2c/drivers/aw882xx_smartpa/6-0034 #
msm8996:/sys/bus/i2c/drivers/aw882xx_smartpa/6-0034 # ls /*Node path*/
algo_ver cali_f0 dbg_prof dsp_re modalias name print_dbg reg uevent
awrw cali_re driver f0_show monitor phase_sync re_range rw
cali cali_time drv_ver fade_step monitor_update power re_show subsystem
```

2) Node function

Node	Function
cali_time	1.Set calibration time; 2.Show current calibration time.
cali	Turn on re and f0 calibration
cali_re	1.Calibrate re; 2.Save calibration re value.
cali_f0	Calibrate f0;
re_show	Show re.
f0_show	Show f0.
re_range	Show the range of calibration re value.

3) Calibration steps

- Play mute music;
- Start the calibration:

```
msm8996:/sys/bus/i2c/drivers/aw882xx_smartpa/6-0034 #
msm8996:/sys/bus/i2c/drivers/aw882xx_smartpa/6-0034 # cat cali_re
pri_l:6371 mOhms pri_r:6380 mOhms
msm8996:/sys/bus/i2c/drivers/aw882xx_smartpa/6-0034 #
```

c. F0 calibration:

```
msm8996:/sys/bus/i2c/drivers/aw882xx_smartpa/6-0034 #
msm8996:/sys/bus/i2c/drivers/aw882xx_smartpa/6-0034 # cat cali_f0
pri_l:830 pri_r:980
msm8996:/sys/bus/i2c/drivers/aw882xx_smartpa/6-0034 #
```

5.4 VERIFICATION OF CALIBRATION RESULTS

- 1) Calibrate several times to confirm whether the calibration re is within the effective range and the value will change slightly. (re effective range confirmed with hardware colleagues),

Calibration twice in attr mode, for example:

```
msm8996:/sys/bus/i2c/drivers/aw882xx_smartpa/6-0034 # cat cali_re
pri_l:7177 mOhms pri_r:6742 mOhms
msm8996:/sys/bus/i2c/drivers/aw882xx_smartpa/6-0034 # cat cali_re
pri_l:7157 mOhms pri_r:6737 mOhms
msm8996:/sys/bus/i2c/drivers/aw882xx_smartpa/6-0034 #
```

- 2) Check whether the re value is written to the file:

```
msm8996:/ # cat mnt/vendor/persist/factory/audio/aw_cali.bin
7157 6737
msm8996:/ #
```

- 3) When playing music, check the dsp_re node, confirm that it is the same as the above calibration value:

```
msm8996:/ # cat sys/bus/i2c/drivers/aw882xx_smartpa/6-0034/dsp_re
7156
msm8996:/ # cat sys/bus/i2c/drivers/aw882xx_smartpa/6-0037/dsp_re
6736
msm8996:/ #
```

(PS: A deviation of 1 in the read re value is normal)

- 4) Restart the phone, play music and check the DSP again_ Re node, confirm DSP_ The value in the re node is the same as the re value in the file.

5.5 CALIBRATION EXAMPLE CODE

AW882XX driver provides reference codes for calling attr and class calibration nodes, as shown below:

```
AW882XX_Driver_QCOM_v1.10.0
├── ap
│   └── smartpa_cali
│       ├── aw882xx_cali_32
│       ├── aw882xx_cali_64
│       └── example_source_code
│           ├── aw882xx_cali_attr_multi_mode.c
│           ├── aw882xx_cali_attr_single_dev_mode.c
│           └── aw882xx_cali_class_example.c
```

*/*Driver root directory*/*

*/*Attr calibration example*/*

*/*Class calibration example*/*

6. DEBUG INTERFACE

6.1 NODE

AW882XX driver will create multiple device node files with different functions. The path is sys/bus/i2c/drivers/aw882xx_smartpa/*-00xx, where * is the i2c bus number and xx is the i2c address. You can use adb to read and write nodes to debug AW882XX driver.

reg

Node name	reg
Description	read and write all register value of aw882xx
Instructions	read register value: cat reg write register value: echo reg_addr reg_data > reg (Hexadecimal operation)
Example	cat reg (get all the values of the register with read permission) echo 0x04 0x0241 > reg (write the value of 0x0241 to the register of 0x04)

rw

Node name	rw
Description	read and write single register value of aw882xx
Instructions	read register value: echo reg_addr > rw (Hexadecimal operation) cat rw write register value: echo reg_addr reg_data > rw (Hexadecimal operation)
Example	echo 0x04 > rw (read the value of the 0x04 register) cat rw echo 0x04 0x0241 > rw (write the value of 0x0241 to the register of 0x04)

driver_ver

Node name	driver_ver
Description	get driver version number
Instructions	get version number: cat driver_ver

dsp_re

Node name	dsp_re
Description	set or get the re value set in the algorithm
Instructions	get re: cat dsp_re set re: echo 7000 > dsp_re

fade_step

Node name	fade_step
Description	set step of fade in and fade out
Instructions	set step: echo step > fade_step get step: cat fade_step
Example	echo 6 > fade_step (set the step to 6) cat fade_step (get the current step of fade in and fade out)

dbg_prof

Node name	dbg_prof
-----------	----------

Description	switch scene function control node
Instructions	switch scene function enable: echo 1 > dbg_prof switch scene function disable: echo 0 > dbg_prof get dbg_prof status: cat dbg_prof

phase_sync

Node name	phase_sync
Description	Phase synchronization function control node
Instructions	Phase synchronization function enable: echo 1 > phase_sync Phase synchronization function disable: echo 0 > phase_sync get phase_sync status: cat phase_sync

print_dbg

Node name	print_dbg
Description	i2c data printing control node
Instructions	i2c data printing enable: echo 1 > print_dbg i2c data printing disable: echo 0 > print_dbg get print_dbg status: cat print_dbg

algo_ver

Node name	algo_ver
Description	get the version of algo
Instructions	get the version of algo: cat algo_ver

monitor

Node name	monitor
Description	switch monitor function
Instructions	turn on the monitor function: echo 1 > monitor turn off the monitor function: echo 0 > monitor get monitor status: cat monitor

monitor_update

Node name	monitor_update
Description	update monitor.bin file
Instructions	update monitor.bin file: echo 1 > monitor_update

6.2 KCONTROL

Kcontrol	function	example
aw_dev_0_prof	mode selection	tinymix aw_dev_0_prof Music select Music mode tinymix aw_dev_0_prof Receiver

		select Receiver mode	
aw_dev_0_switch	switch chip	tinymix aw_dev_0_switch Enable	turn on the chip
		tinymix aw_dev_0_switch Disable	turn off the chip
aw_dev_0_monitor	switch monitor	tinymix aw_dev_0_monitor Enable	Enable monitor
		tinymix aw_dev_0_monitor Disable	Disable monitor
aw882xx_rx_switch	switch mec	tinymix aw882xx_rx_switch 0	turn off mec
		tinymix aw882xx_rx_switch 1	turn on mec
aw882xx_tx_switch	switch tx	tinymix aw882xx_tx_switch 0	turn off tx
		tinymix aw882xx_tx_switch 1	turn on tx
aw882xx_spin_switch	set spin angle	tinymix aw882xx_spin_switch spin_90	rotate 90 degrees
		tinymix aw882xx_spin_switch spin_180	rotate 180 degrees
aw882xx_fadein_us	set stepping time of fade in	tinymix aw882xx_fadein_us 500	set stepping time of fade in :500
aw882xx_fadeout_us	set stepping time of fade out	tinymix aw882xx_fadeout_us 500	set stepping time of fade out: 500

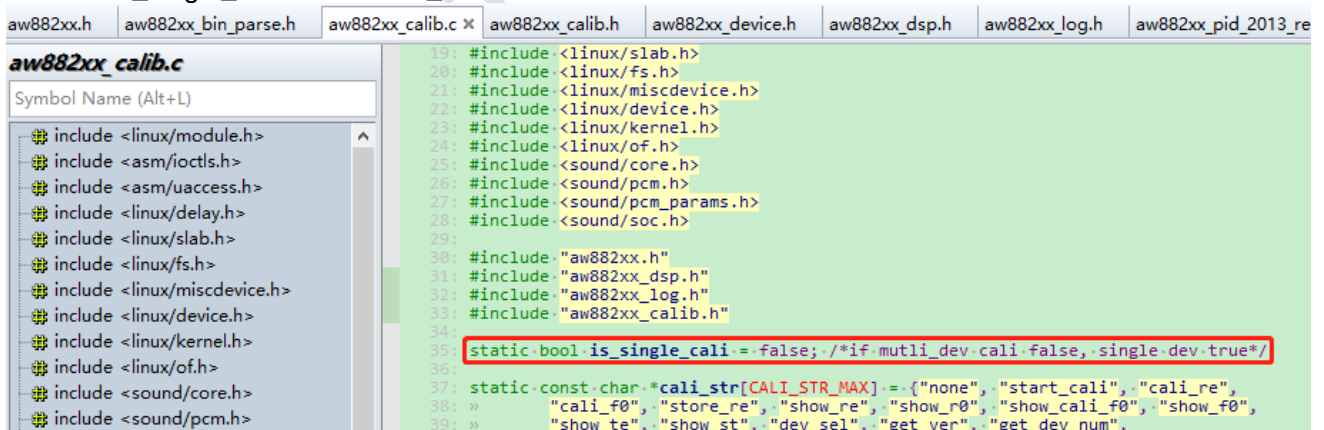
7. APPENDIX

7.1 ABOUT CALIBRATION

Awinic provides misc ioctl node calibration, device attr node calibration and class node calibration methods

Name	Description
/dev/aw882xx_smartpa	Support multi-PA calibration at the same time
/sys/class/smartpa	Support multi-PA calibration at the same time
/sys/bus/i2c/driver/aw882xx_smartpa/x-xx/	Support single PA or multiple PA calibration at the same time

The code defaults to misc/class multi-PA calibration at the same time. The device attr node can support both a single PA calibration and multiple PA calibrations. FAE or clients can configure the global variable is_single_cali in aw882xx_calib.c as needed.



If there is only a single PA, all nodes are single PA calibration.

7.2 SOLUTION OF TX_SWITCH INVALID (APR COMMUNICATION FAILED)

If the tx_switch control is invalid. Please check whether the instance id sent by the driver (adsp communication version v3) is consistent with the instance id setting of the tx module in acdb.

In the following example, the instance id sent by the driver is 0x0, and the instance id of the tx module in acdb is 0x8000. so no return data from the dsp after the kernel sends a message to the dsp, which eventually causes the tx_switch control to fail.

The instance id sent by the driver is 0x0.

```
int aw_send_afe_cal_apr(uint32_t param_id, void *buf, int cmd_size, bool write)
{
    int32_t result = 0, port_id = AFE_PORT_ID_AWDSP_RX;
    int32_t module_id = AFE_MODULE_ID_AWDSP_RX;
    uint32_t port_index = 0;
    uint32_t payload_size = 0;
    size_t len;
    struct rtac_cal_block_data *aw_cal = &(this_afe.aw_cal);
    struct mem_mapping_hdr mem_hdr;
    struct param_hdr_v3 param_hdr;

    pr_debug("%s: enter\n", __func__);

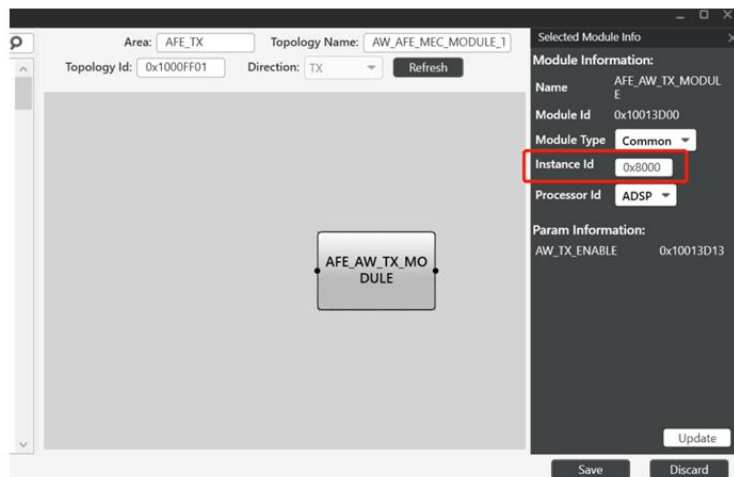
    if (param_id == AFE_PARAM_ID_AWDSP_TX_SET_ENABLE) {
        port_id = AFE_PORT_ID_AWDSP_TX;
        module_id = AFE_MODULE_ID_AWDSP_TX;
    }

    if (aw_cal->map_data.dma_buf == 0) {
        goto err;
    }

    /* Pack message header with data */
    param_hdr.module_id = module_id;
    param_hdr.instance_id = INSTANCE_ID_0;
    param_hdr.param_size = cmd_size;

    if (write) {
        param_hdr.param_id = param_id;
    }
}
```

The instance id of the tx module in acdb is 0x8000.



For the above problem, change the instance id of the tx module in acdb to 0x0000, and the apr communication can be successful.

7.3 KO COMPILE CONFIGURATION

7.3.1 Add Source Code

Create the aw882xx folder under the vendor/qcom/opensource/audio-kernel/asoc/codecs path and add the driver source code.

7.3.2 Add Android.mk And Kbuild

Create Android.mk and kbuild files in aw882xx path. The following is an example of SDM450 configuration. For details, please refer to other modules of the platform Android.mk and kbuild.

Android.mk Configuration

```
# Android makefile for audio kernel modules

# Assume no targets will be supported

AUDIO_CHIPSET := audio
# Build/Package only in case of supported target
ifeq ($(call is-board-platform-in-list,msm8953 msm8937 sdm710 qcs605),true)

LOCAL_PATH := $(call my-dir)

# This makefile is only for DLKM
ifeq ($(findstring vendor,$(LOCAL_PATH)),)

ifeq ($(findstring opensource,$(LOCAL_PATH)),)
    AUDIO_BLD_DIR := $(ANDROID_BUILD_TOP)/vendor/qcom/opensource/audio-kernel
endif # opensource

DLKM_DIR := $(TOP)/device/qcom/common/dlkm

# Build audio.ko as $(AUDIO_CHIPSET)_audio.ko
#####
# This is set once per LOCAL_PATH, not per (kernel) module
KBUILD_OPTIONS := AUDIO_ROOT=$(AUDIO_BLD_DIR)

# We are actually building audio.ko here, as per the
# requirement we are specifying <chipset>_audio.ko as LOCAL_MODULE.
# This means we need to rename the module to <chipset>_audio.ko
# after audio.ko is built.
KBUILD_OPTIONS += MODNAME=aw882xx_dlkm
KBUILD_OPTIONS += BOARD_PLATFORM=$(TARGET_BOARD_PLATFORM)
KBUILD_OPTIONS += $(AUDIO_SELECT)

#####
include $(CLEAR_VARS)
LOCAL_MODULE := $(AUDIO_CHIPSET)_aw882xx.ko
LOCAL_MODULE_KBUILD_NAME := aw882xx_dlkm.ko
LOCAL_MODULE_TAGS := optional
LOCAL_MODULE_DEBUG_ENABLE := true
LOCAL_MODULE_PATH := $(KERNEL_MODULES_OUT)
include $(DLKM_DIR)/AndroidKernelModule.mk
#####
endif # DLKM check
endif # supported target check
```

Kbuild Configuration

```
# We can build either as part of a standalone Kernel build or as
# an external module. Determine which mechanism is being used
ifeq ($(MODNAME),)
    KERNEL_BUILD := 1
```

```

else
    KERNEL_BUILD := 0
endif

ifeq ($(KERNEL_BUILD), 1)
    # These are configurable via Kconfig for kernel-based builds
    # Need to explicitly configure for Android-based builds
    AUDIO_BLD_DIR := $(ANDROID_BUILD_TOP)/kernel/msm-4.9
    AUDIO_ROOT := $(AUDIO_BLD_DIR)/techpack/audio
endif

ifeq ($(KERNEL_BUILD), 0)
    ifeq ($(CONFIG_ARCH_SDM845), y)
        include $(AUDIO_ROOT)/config/sdm845auto.conf
        export
        INCS += -include $(AUDIO_ROOT)/config/sdm845autoconf.h
    endif
    ifeq ($(CONFIG_ARCH_SDM670), y)
        include $(AUDIO_ROOT)/config/sdm710auto.conf
        export
        INCS += -include $(AUDIO_ROOT)/config/sdm710autoconf.h
    endif
    ifeq ($(CONFIG_ARCH_SDM450), y)
        include $(AUDIO_ROOT)/config/sdm450auto.conf
        export
        INCS += -include $(AUDIO_ROOT)/config/sdm450autoconf.h
    endif
endif

# As per target team, build is done as follows:
# Defconfig : build with default flags
# Slub      : defconfig + CONFIG_SLUB_DEBUG := y +
#            CONFIG_SLUB_DEBUG_ON := y + CONFIG_PAGE_POISONING := y
# Perf      : Using appropriate msmXXXX-perf_defconfig
#
# Shipment builds (user variants) should not have any debug feature
# enabled. This is identified using 'TARGET_BUILD_VARIANT'. Slub builds
# are identified using the CONFIG_SLUB_DEBUG_ON configuration. Since
# there is no other way to identify defconfig builds, QTI internal
# representation of perf builds (identified using the string 'perf'),
# is used to identify if the build is a slub or defconfig one. This
# way no critical debug feature will be enabled for perf and shipment
# builds. Other OEMs are also protected using the TARGET_BUILD_VARIANT
# config.

##### UAPI #####
UAPI_DIR := uapi
UAPI_INC := -I$(AUDIO_ROOT)/include/$(UAPI_DIR)

##### COMMON #####
COMMON_DIR := include

```

```
COMMON_INC := -I$(AUDIO_ROOT)/$(COMMON_DIR)

##### AW882XX #####

# for AW882XX PA
ifdef CONFIG_SND_SOC_AW882XX
    AW882XX_PA_OBJS += aw882xx.o
endif

LINUX_INC += -include/linux

INCS +=      $(COMMON_INC) \
            $(UAPI_INC)

EXTRA_CFLAGS += $(INCS)

CDEFINES += -DANI_LITTLE_BYTE_ENDIAN \
            -DANI_LITTLE_BIT_ENDIAN \
            -DDOT11F_LITTLE_ENDIAN_HOST \
            -DANI_COMPILER_TYPE_GCC \
            -DANI_OS_TYPE_ANDROID=6 \
            -DPTT_SOCKET_SVC_ENABLE \
-Wall\
-Werror\
-D__linux__

KBUILD_CPPFLAGS += $(CDEFINES)

# Currently, for versions of gcc which support it, the kernel Makefile
# is disabling the maybe-uninitialized warning.  Re-enable it for the
# AUDIO driver.  Note that we must use EXTRA_CFLAGS here so that it
# will override the kernel settings.
ifeq ($(call cc-option-yn, -Wmaybe-uninitialized),y)
EXTRA_CFLAGS += -Wmaybe-uninitialized
endif
#EXTRA_CFLAGS += -Wmissing-prototypes

ifeq ($(call cc-option-yn, -Wheader-guard),y)
EXTRA_CFLAGS += -Wheader-guard
endif

ifeq ($(KERNEL_BUILD), 0)
KBUILD_EXTRA_SYMBOLS +=$(OUT)/obj/vendor/qcom/opensource/audio-kernel/ipc/Module.symvers
KBUILD_EXTRA_SYMBOLS +=$(OUT)/obj/vendor/qcom/opensource/audio-kernel/dsp/Module.symvers
KBUILD_EXTRA_SYMBOLS +=$(OUT)/obj/vendor/qcom/opensource/audio-kernel/asoc/Module.symvers
KBUILD_EXTRA_SYMBOLS +=$(OUT)/obj/vendor/qcom/opensource/audio-kernel/asoc/codecs/Module.symvers
KBUILD_EXTRA_SYMBOLS +=$(OUT)/obj/vendor/qcom/opensource/audio-kernel/soc/Module.symvers
endif

# Module information used by KBuild framework

obj-$(CONFIG_SND_SOC_AW882XX) += aw882xx_dtlm.o
```

```
aw882xx_dtkm-y := $(AW882XX_PA_OBJS)
```

```
# inject some build related information
```

```
DEFINES += -DBUILD_TIMESTAMP="\$(shell date -u +%Y-%m-%dT%H:%M:%SZ)"
```

7.3.3 Define Compilation Options

```
--- a/config/sdm450auto.conf
+++ b/config/sdm450auto.conf
@@ -39,3 +39,4 @@ CONFIG_SND_SOC_ANALOG_CDC=m
CONFIG_SND_SOC_DIGITAL_CDC_LEGACY=m
CONFIG_SND_SOC_MSM_HDMI_CODEC_RX=m
CONFIG_WCD_DSP_GLINK=m
+++CONFIG_SND_SMARTPA_AW882XX=m

diff --git a/config/sdm450autoconf.h b/config/sdm450autoconf.h
index 1aca114..b01004e 100644
--- a/config/sdm450autoconf.h
+++ b/config/sdm450autoconf.h
@@ -55,3 +55,4 @@
#define CONFIG_SND_SOC_MSM_HDMI_CODEC_RX 1
#define CONFIG_COMMON_CLK 1
#define CONFIG_WCD_DSP_GLINK 1
+++#define CONFIG_SND_SMARTPA_AW882XX 1
```

7.3.4 Add Compilation Path

1) Add a new driver directory to the compilation directory

```
--- a/Android.mk
+++ b/Android.mk
@@ -13,6 +13,7 @@ $(shell rm -rf
$(PRODUCT_OUT)/obj/vendor/qcom/opensource/audio-kernel/asoc/codecs/wcd934x/Module.symvers)
+$(shell rm -rf $(PRODUCT_OUT)/obj/vendor/qcom/opensource/audiokernel/asoc/codecs/aw882xx/
Module.symvers)
@@ -22,6 +23,7 @@ include $(MY_LOCAL_PATH)/soc/Android.mk
include $(MY_LOCAL_PATH)/asoc/codecs/wcd934x/Android.mk
+include $(MY_LOCAL_PATH)/asoc/codecs/aw882xx/Android.mk
```

2) Add relevant configurations in the device / qcom / XXXX directory (XXXX represents the project name of ODM / OEM)

BoardConfig.mk

```
BOARD_VENDOR_KERNEL_MODULES := $(KERNEL_MODULES_OUT)/audio_aw882xx.ko
```

XXXX.mk

```
AUDIO_DLKM += audio_aw882xx.ko
```

3) Add driver module audio_aw882xx.ko to init.target.rc.

```
/vendor/bin/modprobe -a -d /vendor/lib/modules audio_q6_pdr audio_q6_notifier
audio_snd_event audio_apr audio_adsp_loader audio_q6 audio_native audio_usf
audio_pinctrl_wcd audio_pinctrl_lpi audio_swr audio_platform audio_hdmi
audio_stub audio_wcd core audio_wsa881x audio_bolero cdc audio_wsa macro
```

```
audio_va_macro audio_rx_macro audio_tx_macro audio_wcd938x audio_wcd938x_slave
audio_machine_kona audio aw882xx
```

7.4 PLATFORM I2C BUS DYNAMIC CHANGES

If the platform will change the i2c bus number, the i2c bus number of dynamic changes will cause the failure of dai_link matching, can solve by modifying the configuration in the device tree and dai_link configuration.

7.4.1 DTS Configuration

Add the rename-flag attribute to the driver node and set the attribute value to 1.

Single PA Configuration

```
diff --git a/arch/arm/boot/dts/qcom/apq8096-dragonboard.dtsi
b/arch/arm/boot/dts/qcom/apq8096-dragonboard.dtsi
index f22db2e..a340a32 100644
--- a/arch/arm/boot/dts/qcom/apq8096-dragonboard.dtsi
+++ b/arch/arm/boot/dts/qcom/apq8096-dragonboard.dtsi
@@ -549,6 +549,8 @@
     i2c_x {                                /*x means the i2c bus number*/
+
+        /* AWINIC AW882XX mono Smart PA */
+        aw882xx_smartpa@34 {
+            compatible = "awinic,aw882xx_smartpa";
+            reg = <0x34>;
+            /* You cannot configure reset-gpio for aw88230, aw88257, aw88261,
aw88082 or aw88265*/
+            reset-gpio = <&tlmm 84 0>;
+            irq-gpio = <&tlmm 136 0x2008>;
+            aw-tx-topo-id = <0x1000ff00>;
+            aw-rx-topo-id = <0x1000ff01>;
+            /*Set this parameter based on the port id of the I2S TX in use*/
+            aw-tx-port-id = <0x1007>;
+            /*Set this parameter based on the port id of the I2S RX in use*/
+            aw-rx-port-id = <0x1006>;
+            aw-re-min = <4000>;                /*Minimum calibration value (mohms)*/
+            aw-re-max= <30000>;                /*Maximum calibration value (mohms)*/
+            /*You cannot configure aw-cali-mode for products with IV*/
+            aw-cali-mode = "none";
+            rename-flag = <1>;
+            status = "okay";
+        };
+    };
+    /* AWINIC AW882XX mono Smart PA End */
+    /*re means resistance of speaker */
```

Multiple PA Configuration

```
diff --git a/arch/arm/boot/dts/qcom/apq8096-dragonboard.dtsi
b/arch/arm/boot/dts/qcom/apq8096-dragonboard.dtsi
index f22db2e..a340a32 100644
--- a/arch/arm/boot/dts/qcom/apq8096-dragonboard.dtsi
+++ b/arch/arm/boot/dts/qcom/apq8096-dragonboard.dtsi
@@ -549,6 +549,8 @@
     i2c_x {                                /*x means the i2c bus number*/
+
+        /* AWINIC AW882XX Smart PA */
+        aw882xx_smartpa@34 {
```

```
+ compatible = "awinic,aw882xx_smartpa";
+ reg = <0x34>;
+ reset-gpio = <&tlmm 84 0>;
+ irq-gpio = <&tlmm 136 0x2008>;
+ sound-channel = <0>; /*0:pri_l 1:pri_r 2:sec_l 3:sec_r*/
+ aw-tx-topo-id = <0x1000ff00>;
+ aw-rx-topo-id = <0x1000ff01>;
+ aw-tx-port-id = <0x1007>;
+ aw-rx-port-id = <0x1006>;
+ aw-re-min = <4000>;
+ aw-re-max= <30000>;
+ /*aw-cali-mode = "none";*/
+ rename-flag = <1>;
+ status = "okay";
+ };
+ aw882xx_smartpa@35 {
+ compatible = "awinic,aw882xx_smartpa";
+ reg = <0x35>;
+ reset-gpio = <&tlmm 82 0>;
+ irq-gpio = <&tlmm 143 0x2008>;
+ sound-channel = <1>; /*0:pri_l 1:pri_r 2:sec_l 3:sec_r*/
+ aw-tx-topo-id = <0x1000ff00>;
+ aw-rx-topo-id = <0x1000ff01>;
+ aw-tx-port-id = <0x1007>;
+ aw-rx-port-id = <0x1006>;
+ aw-re-min = <4000>;
+ aw-re-max= <30000>;
+ /*aw-cali-mode = "none";*/
+ rename-flag = <1>;
+ status = "okay";
+ };
+ /* AWINIC AW882XX Smart PA End */
```

7.4.2 DAI_LINK Configuration

Change the suffix of codec_name and codec_dai_name to sound-channel,different platform dai_link configuration is as follows:

Kernel Versions Prior To 5.4

3) Add awinic_codecs array

Single PA Configuration

```
diff --git a/sound/soc/msm/msm8996.c b/sound/soc/msm/msm8996.c
index ab04888..b5e4bf5 100644
--- a/sound/soc/msm/msm8996.c
+++ b/sound/soc/msm/msm8996.c
+struct snd_soc_dai_link_component awinic_codecs[] = {
+ {
+ .of_node = NULL,
+ .dai_name = "aw882xx-aif-0", /*Change the suffix to sound-channel of
DTS node */
+ .name = "aw882xx_smartpa_0",
+ },
+ };
```

Multiple PA Configuration

```
diff --git a/sound/soc/msm/msm8996.c b/sound/soc/msm/msm8996.c
index ab04888..b5e4bf5 100644
--- a/sound/soc/msm/msm8996.c
+++ b/sound/soc/msm/msm8996.c
+struct snd_soc_dai_link_component awinic_codecs[] = {
+ {
+     .of_node = NULL,
+     .dai_name = "aw882xx-aif-0",    /* Change the suffix to sound-channel of
DTS node */
+     .name = "aw882xx_smartpa_0",
+ },
+ {
+     .of_node = NULL,
+     .dai_name = "aw882xx-aif-1",    /* Change the suffix to sound-channel of
DTS node */
+     .name = "aw882xx_smartpa_1",
+ },
+}
```

4) Modify DAI interface

```
@@ -3775,8 +3777,10 @@ static struct snd_soc_dai_link
msm8996_common_be_dai_links[] = {
{
    .name = LPASS_BE_QUAT_MI2S_RX,
    /* Take the Quaternary MI2S interface for example*/
    .stream_name = "Quaternary MI2S Playback",
    .cpu_dai_name = "msm-dai-q6-mi2s.3",
    .platform_name = "msm-pcm-routing",
+ifdef CONFIG_SND_SMARTPA_AW882XX
+    .num_codecs = ARRAY_SIZE(awinic_codecs),
+    .codecs = awinic_codecs,
+else
    .codec_name = "msm-stub-codec.1",
    .codec_dai_name = "msm-stub-rx",
+endif
    .no_pcm = 1,
    .dpcm_playback = 1,
    .be_id = MSM_BACKEND_DAI_QUATERNARY_MI2S_RX,
    .be_hw_params_fixup = msm_quat_mi2s_rx_be_hw_params_fixup,
    .ops = &msm8996_quat_mi2s_be_ops,
    .ignore_suspend = 1,
},
},
```

Kernel 5.4 And Later

Single PA Configuration

```
/*Take the PRI MI2S RX interface as an example*/
SND_SOC_DAILINK_DEFS(pri_mi2s_rx,
    DAILINK_COMP_ARRAY(COMP_CPU("msm-dai-q6-mi2s.0")),
    /* Change the suffix to sound-channel of DTS node */
    DAILINK_COMP_ARRAY(COMP_CODEC("aw882xx_smartpa_0", "aw882xx-aif-0")),
    DAILINK_COMP_ARRAY(COMP_PLATFORM("msm-pcm-routing")));
```

Multiple PA Configuration

```
SND_SOC_DAILINK_DEFS(pri_mi2s_rx,
    DAILINK_COMP_ARRAY(COMP_CPU("msm-dai-q6-mi2s.0")),
```



```
/* Change the suffix to sound-channel of DTS node */  
DAILINK_COMP_ARRAY (COMP_CODEC("aw882xx_smartpa_0", "aw882xx-aif-0"),  
COMP_CODEC("aw882xx_smartpa_1", "aw882xx-aif-1")),  
DAILINK_COMP_ARRAY (COMP_PLATFORM("msm-pcm-routing")));
```

Awinic Driver