EC2x&AG35-Quecopen

Customizing the Codec Driver

LTE Module Series

Rev. EC2x&AG35-Quecopen_Customizing the Codec Driver _V1.0

Date: 2018-03-01

Status: Preliminary

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.
7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China
Tel: +86 21 5108 6236
Email: info@quectel.com

Or our local office. For more information, please visit:
http://www.quectel.com/support/sales.htm

For technical support, or to report documentation errors, please visit:
http://www.quectel.com/support/technical.htm
Or email to: support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

About the Document

History

Revision	Date	Author	Description
1.0	2018-03-01	Yang Yang	Initial

Contents

1 Codec Customization Requirements

Quectel supports several Codec (ALC5616, NAU8814, NAU8810, and TLV320AIC3104) by default. The Codec driver supports both dynamic and static loading. Users can modify Codec loading method according to personal needs. If users do not need to use the Codec that Quectel recommends, please refer to this document to modify the code in order to integrate the corresponding Codec driver.

1.1. Hardware Connection Diagram and GPIO Configuration

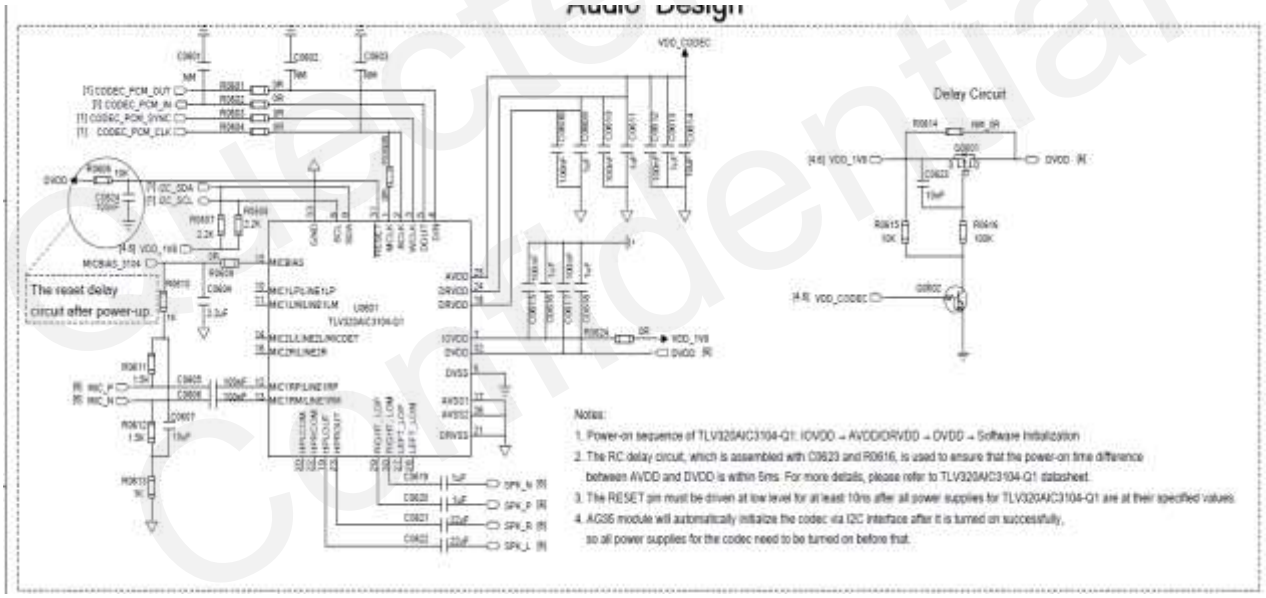


Figure 1：TLV320AIC3104 Hardware Connection Diagram

Table 1：CODEC PIN

CODEC PIN	MDM9x28 PIN
CODEC_PCM_OUT	GPIO77/GPIO22
CODEC_PCM_IN	GPIO76/GPIO21
CODEC_PCM_SYNC	GPIO79/GPIO20
CODEC_PCM_CLK	GPIO78/GPIO23
I2C_SDA	GPIO18(AGxx)，GPIO6(ECxx)，
I2C_SCL	GPIO19(AGxx)，GPIO7(ECxx)，

In Table 1, CODEC_PCM_OUT, CODEC_PCM_IN, CODEC_PCM_SYNC and CODEC_PCM_CLK are the pins to control audio data and clock signals. I2C_SDA(GPIO18) and I2C_SCL(GPIO19) are the pins of I2C data and I2C clock respectively, which control the operations of reading and writing the Codec register.

1.2. The Method to Add the Codec Driver

If users need to add another new codec driver, go through the following four steps.

A. Confirm the parameters on PCM such as mode, fsync, clock, format etc. according to PCM interface requirements on codec datasheet. Then configure mdm9607.dtsi according to these parameters so that to configure the multiplexing of module PCM pins and PCM format of the module end. For more detailed configurations, please refer to Chapter 1.2.1.
B. Obtain the Codec driver code and add it to kernel. For detailed methods, please refer to Chapter 1.2.2.
C. Configure the Codec name at boot. For detailed methods, please refer to Chapter 1.2.3.
D. Recompile and download to confirm that the newly added Codec can be used normally.

Take ALC5616 as an example to introduce the specific method of adding the Codec driver.

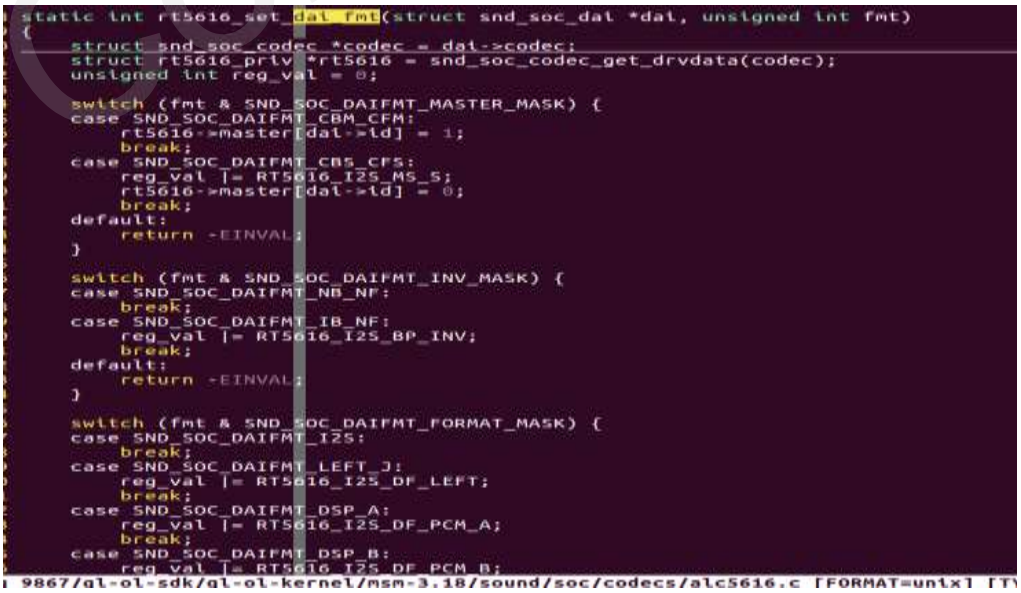
1.2.1 Configure the Module PCM Interface Parameters Based on Codec PCM Parameters

Modify msm-3.18/arch/arm/boot/dts/qcom/mdm9607.dtsi

```
    sound-9306 {
            compatible = "qcom,mdm9607-audio-tapan";
```

```
qcom,model = "mdm9607-tapan-i2s-snd-card";
status = "disabled";
……
+++dai_sec_auxpcm: qcom,msm-sec-auxpcm {
    +++compatible = "qcom,msm-auxpcm-dev";
    +++qcom,msm-cpudai-auxpcm-mode = <0>, <0>;
    +++qcom,msm-cpudai-auxpcm-sync = <1>, <1>;
    +++qcom,msm-cpudai-auxpcm-frame = <5>, <5>;
    +++qcom,msm-cpudai-auxpcm-quant = <2>, <2>;
    +++qcom,msm-cpudai-auxpcm-num-slots = <1>, <1>;
    +++qcom,msm-cpudai-auxpcm-slot-mapping = <1>, <1>;
    +++qcom,msm-cpudai-auxpcm-data = <0>, <0>;
    +++qcom,msm-cpudai-auxpcm-pcm-clk-rate = <2048000>, <4096000>;
    +++qcom,msm-cpudai-afe-clk-ver = <2>;
    +++qcom,msm-auxpcm-interface = "secondary";
    +++pinctrl-names = "default", "idle";
    +++pinctrl-0 = <&sec_auxpcm_ws_active
        &sec_auxpcm_sck_active
        &sec_auxpcm_dout_active
        &sec_auxpcm_din_active>;
    +++pinctrl-1 = <&sec_auxpcm_ws_sleep
        &sec_auxpcm_sck_sleep
        &sec_auxpcm_dout_sleep
        &sec_auxpcm_din_sleep>;
};
```

Configure Codec driver parameters by using Codec driver interface rt5616_set_dai_fmt of alc5616.



1.2.2 Add the Codec Driver to Kernel

A. Write the Codec driver source code based on alc5616 codec datasheet.
B. Copy the Codec driver alc5616.c, alc5616.h into the directory msm-3.18/sound/soc/codecs.
C. Modify msm-3.18/arch/arm/config/mdm9607_degconfig(debug) mdm9607-perf_defconfig(release)
+++CONFIG_SND_SOC_ALC5616=y or +++CONFIG_SND_SOC_ALC5616=y
Modify msm-3.18/sound/soc/codec/Makefile ,

```
snd-soc-tas2552-objs := tas2552.o
+++snd-soc-alc5616-objs := alc5616.o,
+++obj-$(CONFIG_SND_SOC_ALC5616) += snd-soc-alc5616.o
```

Modify msm-3.18/sound/soc/codecs/Kconfig

```
Config SND_SOC_TLV320AIC3XX
    Tristate”Texas Instruments TLV320AIC31xx”
    Depends on I2C
    Select REGMAP_I2c
    +++config SND_SOC_ALC5616
    +++tristate “alc5616 codecs”
```

1.2.3 Select the Codec Loaded at Boot

When codec_name is registering Codec driver, the code obtains the name of i2c_driver, i2c bus and slave addr, and also splices the three strings together. Here codec_name = “i2c->driver->name.i2c_bus-i2c->addr” when loading the new Codec driver. For rx_dai_name and tx_dai_name, keep them the same name as that of Codec driver code snd_soc_dai_driver.

```
struct snd_soc_dai_driver rt5616_dai[] = {
    {
        .name = "rt5616-aif1",
        .id = RT5616_AIF1,
        .playback = {
            .stream_name = "AIF1 Playback",
            .channels_min = 1,
            .channels_max = 2,
```

Modify msm-3.18/sound/soc/msm/mdm9607.c

```
---static char quec_codec_name[32] = {‘a’};
---static char quec_rx_dai_name[32]= {‘a’};
---static char quec_tx_dai_name[32] = {‘a’};
+++static char quec_codec_name[32] = {“alc5616-codec.4-001b”};
+++static char quec_rx_dai_name[32]= {“rt5616-aif1”};
+++static char quec_tx_dai_name[32] = {“rt5616-aif1”}
```

1.2.4 Compile, Download and Debug to Confirm If the New Codec Driver Can be Loaded

Compile SDK Development Environment.

A. make kernel_menuconfig (Must-do: Modify xxx_defconfi,)
B. make kernel
C. Check if the driver code has compiled and generated alc5616.o

D. Check If PCM device of sound card has registered in ls /dev/snd

1.3. Codec Driver Tests

A. The primary PCM Test

```
playback
    amix 'AUX_PCM_RX Audio Mixer MultiMedia1' 1
    aplay /data/ringtone1.wav
recording
    amix 'MultiMedia1 Mixer AUX_PCM_UL_TX'
    arec -C 1 -R 8000 data/rec.wav
voice call
    amix 'AUX_PCM_RX_Voice Mixer CSVoice' 1
    amix 'Voice_Tx Mixer AUX_PCM_TX_Voice' 1
    aplay -D hw:0,2 -P&arec -D hw:0,2 -P -R 8000 -C 1
```

B. The second PCM Test

```
playback
    amix 'SEC_AUX_PCM_RX Audio Mixer MultiMedia1' 1
    aplay /data/ringtone1.wav
recording
```

```
            amix 'MultiMedia1 Mixer SEC_AUX_PCM_UL_TX' 1
            arec -C 1 -R 8000 data/rec.wav
        voice call
            amix 'SEC_AUX_PCM_RX_Voice Mixer CSVoice' 1
            amix 'Voice_Tx Mixer SEC_AUX_PCM_TX_Voice' 1
aplay -D hw:0,2 -P&arec -D hw:0,2 -P -R 8000 -C 1
```