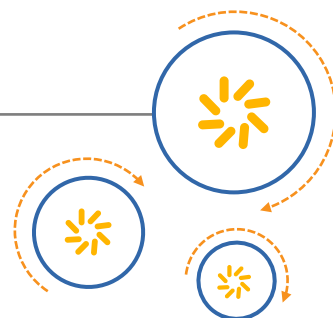




Qualcomm Technologies, Inc.



多媒体驱动程序开发和调通指南 – 音频

80-NU323-1SC版本A

2014年11月25日

机密和专有信息 – Qualcomm Technologies, Inc.

© 2014, 2015 Qualcomm Technologies, Inc.和/或其附属公司。保留所有权利。

禁止公开披露：遇有在公共服务器或网站上发布本文档的情形，请将举报意见发送至：DocCtrlAgent@qualcomm.com。

未经Qualcomm Technologies, Inc.的明确书面许可，不得使用、拷贝、复制或修改其全部或部分内容，或以任何方式向他人泄露其内容。

Qualcomm
Confidential - May Contain Trade Secrets
2020-07-02 04:36:29 PDT
luo.yihua@qnx.com

限制分发：未经Qualcomm配置管理的明确批准，不得分发给Qualcomm Technologies, Inc.及其附属公司员工以外的人员。

Qualcomm是Qualcomm Incorporated在美国及其他国家/地区的注册商标。QUALCOMM Incorporated的所有商标需要得到许可才能使用。其他产品或品牌名称可能是其各自所有者的商标或注册商标。

本技术资料可能受美国和国际出口、转口或转让（统称“出口”）法律的约束。严禁违反美国和国际法律。

Qualcomm Technologies, Inc.
5775 Morehouse Drive
San Diego, CA 92121
U.S.A.

修订记录

版本	日期	描述
A	2014年11月	初始版本

Qualcomm
Confidential - May Contain Trade Secrets
2020-07-02 04:36:29 PDT
luo.yihua@qnx.com

目录

1 简介	6
1.1 文档用途	6
1.2 符号惯例	6
1.3 技术协助	6
2 音频调通	7
2.1 前提条件	7
2.2 音频调通工作流程	8
2.3 确认PIL已成功加载DSP	9
2.4 验证声卡是否已经注册	10
2.5 确认音频编解码器已正确复位	10
2.6 验证硬件设计并完成相应的软件配置	11
2.6.1 耳机/手机MICBIAS配置	11
2.6.2 NC/NO插孔类型设置	11
2.6.3 外部电容设置	12
2.6.4 MICBIAS电压设置	12
2.6.5 主麦克风配置	12
2.6.6 MCLK	12
2.7 只要集成外部音频组件，均需提供软件修改指南	13
2.8 确认音频特定的GPIO已正确配置	13
2.8.1 pdm GPIO的Pin_ctl定义	14
2.8.2 美国(OMTP)和欧洲（非OMTP）GPIO的Pin_ctl定义	14
2.9 验证音频路径	15
2.9.1 播放	15
2.9.2 录音	16
2.9.3 语音呼叫	16
2.9.4 MBHC	19
A 参考资料	21

图

图2-1 音频调通工作流程 8

表

表2-1 MSM8909 GPIO..... 13

Qualcomm
Confidential – May Contain Trade Secrets
2020-07-02 04:36:29 PDT
luo.yihua@qnx.com

1 简介

1.1 文档用途

本文档为驱动程序开发指南，介绍如何在MSM8909 Android平台上调通音频模块。

其他多媒体技术的驱动程序开发指南以及调通步骤，将在以下各个文档中分别进行介绍：

- 《多媒体驱动程序开发和调通指南 – 摄像头》(80-NU323-2)
- 《多媒体驱动程序开发和调通指南 – 显示》(80-NU323-3)
- 《多媒体驱动程序开发和调通指南 – 视频》(80-NU323-5)

1.2 符号惯例

函数声明、函数名称、类型声明以及代码示例以不同字体表示，例如#include。

代码变量括在尖括号内，例如<数字>。

要输入的命令以不同字体表示，例如copy a:*. * b:。

若您通过彩色显示器查看本文档，或用彩色打印机打印本文档，则**红色粗体**表示要**添加**的代码，蓝色带删除线表示要被**替换**或**删除**的代码。

1.3 技术协助

针对本文档中的信息，如需协助或说明，请通过<https://support.cdmatech.com/>向Qualcomm Technologies, Inc. (QTI)提交用例。

如果无法访问CDMATech支持网站，请在注册后进行访问，或发送电子邮件至support.cdmatech@qti.qualcomm.com。

2 音频调通

本文档详细介绍了MSM8909 Linux Android (LA)音频调通的相关步骤。

音频调通流程包括下列步骤：

1. 确认外围图像加载器(PIL)已成功加载DSP – 第2.3节
2. 确认声卡已注册 – 第2.4节
3. 确认编解码器上电顺序并检查是否正确复位 – 第2.5节
4. 按照QTI标准参考原理图验证硬件设计并对软件进行适当更改 – 第2.6节
5. 只要集成外部音频组件，均需提供软件修改指南 – 第2.7节
6. 确认音频特定的GPIO配置 – 第2.7节
7. 确认音频路径 – 第2.9节

2.1 前提条件

1. 由QTI客户工程(CE)硬件团队完成音频硬件原理图审查工作。
 - a. 请按照音频硬件原理图审查清单进行审查
 - b. 在SalesForce中通过“New Case→Record Type of new Record→Design Review”新建用例，从而启动硬件设计审查
2. 确保已针对所有硬件设计审查意见进行了改进

注： ULT音频核心同时支持的接收和发送接口数量有限。应针对具体要求对设计进行审查，并将上述限制考虑在内。

2.2 音频调通工作流程

图2-1所示为整个音频调通工作流程。

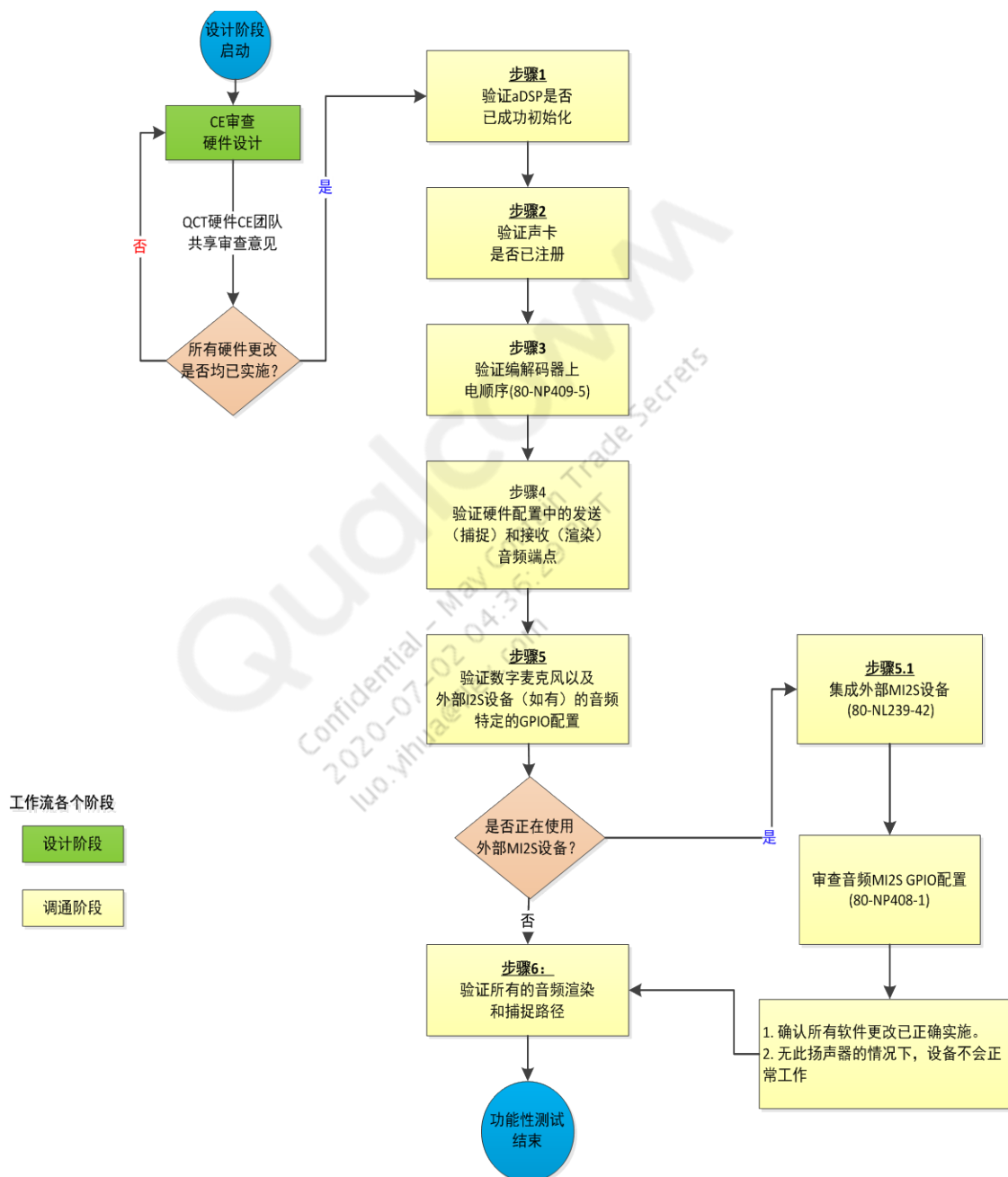


图2-1 音频调通工作流程

2.3 确认PIL已成功加载DSP

以下日志标记可指示PIL已成功加载ADSP图像：

```
adsp-loader qcom,msm-adsp-loader.19: adsp_loader_do: Q6/MDSP image is loaded
```

如果启动日志中不存在此消息，则ADSP图像可能尚未加载或DSP可能已崩溃。请检查obj\KERNEL_OBJ\drivers\soc\qcom\qdsp6v2\文件夹下是否存在module adsp-loader.o。

正常情况下，会以静态方式加载ADSP图像。ADSP图像需要在源代码中正确地进行配置才能成功载入RAM。将以下代码添加至init.target.rc文件：

```
write /sys/kernel/boot_adsp/boot 1
```

/kernel/arch/arm/ configs/msm8909_defconfig中应存在CONFIG_MSM_ADSP_LOADER配置宏。查找以下语句：

```
CONFIG_MSM_ADSP_LOADER=y
```

kernel/drivers/soc/qcom/Kconfig文件中应存在以下代码：

```
config MSM_ADSP_LOADER
tristate "ADSP loader support"
select SND_SOC_MSM_APRV2_INTF
depends on MSM_QDSP6_APRV2 || MSM_QDSP6_APRV3
help
Enable ADSP image loader.
The ADSP loader brings ADSP out of reset
for the platforms that use APRv2.
Say M if you want to enable this module.
```

以下日志标记可指示PIL已成功加载DSP图像，并且SMD驱动程序在ADSP中已发现服务并已通知APR驱动程序：

```
apr_tal:Modem Is Up
```

2.4 验证声卡是否已经注册

映射DAI链接后，会按以下消息所示枚举声卡：

```
msm8x16-asoc-wcd msm-snd-card.0: snd-soc-dummy-dai <-> MultiMedia1 mapping ok
msm8x16-asoc-wcd msm-snd-card.0: snd-soc-dummy-dai <-> MultiMedia2 mapping ok
msm8x16-asoc-wcd msm-snd-card.0: snd-soc-dummy-dai <-> CS-VOICE mapping ok
msm8x16-asoc-wcd msm-snd-card.0: snd-soc-dummy-dai <-> VoIP mapping ok
msm8x16-asoc-wcd msm-snd-card.0: snd-soc-dummy-dai <-> MultiMedia3 mapping ok
msm8x16-asoc-wcd msm-snd-card.0: snd-soc-dummy-dai <-> PRI_MI2S_RX_HOSTLESS
mapping ok
msm8x16-asoc-wcd msm-snd-card.0: snd-soc-dummy-dai <-> INT_FM_HOSTLESS mapping
ok
msm8x16-asoc-wcd msm-snd-card.0: msm-stub-rx <-> msm-dai-q6-dev.241 mapping ok
msm8x16-asoc-wcd msm-snd-card.0: msm-stub-tx <-> msm-dai-q6-dev.240 mapping ok
compress asoc: snd-soc-dummy-dai <-> MultiMedia4 mapping ok
```

通过在adb shell中输入以下命令确认声卡注册是否成功：

```
adb root
adb shell
cat Sys/kernel/debug/asoc/
```

应存在含有声卡名称(msm8909-snd-card)的一个条目。

2.5 确认音频编解码器已正确复位

请参见《MSM8909/MSM8209/MSM8208数字基带设计指南/培训幻灯片》(80-NP409-5)，了解有关编解码器上电顺序的详细信息。

根据所使用的音频硬件接口不同，必须在示波器上检查不同的时钟。

2.6 验证硬件设计并完成相应的软件配置

本部分详细介绍基于硬件更改而所需完成的典型软件配置。

2.6.1 耳机/手机MICBIAS配置

根据耳机和手机的硬件配置，需在kernel/arch/arm/boot/dts/msm8909-xxx.dtsi文件中进行如下更改。

如果耳机具有外部MICBIAS，则采用以下配置：

```
qcom,msm-hs-micbias-type = "external";
"MIC BIAS External", "Headset Mic",
"AMIC2", "MIC BIAS External",
```

如果耳机具有内部MICBIAS，则采用以下配置：

```
qcom,msm-hs-micbias-type = "internal";
"MIC BIAS Internal", "Headset Mic",
"AMIC2", "MIC BIAS Internal",
```

如果手机具有外部MICBIAS，则采用以下配置：

```
"MIC BIAS External", "Handset Mic",
"AMIC1", "MIC BIAS External",
```

如果手机具有内部MICBIAS，则采用以下配置：

```
"MIC BIAS Internal", "Handset Mic",
"AMIC1", "MIC BIAS Internal",
```

2.6.2 NC/NO插孔类型设置

根据插孔设备修改以下dtsi文件参数（0表示NC，1表示NO）：

```
sound {
    ...
    qcom,msm-mbhc-hphl-swh = <0>;
    qcom,msm-mbhc-gnd-swh = <0>;
    ...
}
```

2.6.3 外部电容设置

如果手机具有CAP，则将以下语句添加至dtsti文件：

```
qcom, msm-micbias1-ext-cap
```

如果耳机具有CAP，则将以下语句添加至dtsti文件：

```
qcom,msm-micbias2-ext-cap
```

如果耳机和手机都具有CAP，则添加上述两个语句。

2.6.4 MICBIAS电压设置

如需使用不同于默认设置(1.8v)的MICBIAS电压，请将以下语句添加至dtsti文件：

```
qcom,cdc-micbias-cfilt-mv = <2700>;
```

如未添加该语句，则将使用默认电压。

2.6.5 主麦克风配置

如需将主麦克风连接到不同于默认设置的（音频输入1）音频输入，请对编解码驱动程序文件msm8x16-wcd.c进行以下更改：

```
{"ADC1", NULL, "AMIC1"}
{"ADC2_INP2", NULL, "AMIC1"},
```

2.6.6 MCLK

MSM用于驱动编解码器所需的主时钟。MCLK时钟频率和时钟源在设备树中定义。根据外部编解码器和外部扬声器放大器要求，可能需要修改MCLK频率。

如需修改MCLK频率，请对kernel/arch/arm/boot/dts/msm8909-xxx.dtsi文件进行以下更改：

```
sound {
    compatible = "qcom,msm8x16-audio-codec";
    qcom,model = "msm8909-snd-card";
    qcom,msm-snd-card-id = <0>;
    qcom,msm-codec-type = "internal";
    qcom,msm-ext-pa = "primary";
    qcom,msm-mclk-freq = <9600000>;
```

2.7 只要集成外部音频组件，均需提供软件修改指南

请参见《MSM8916外部MI2S接口概述》(80-NL239-42)，了解该步骤所需进行的软件更改的详细步骤说明。

注：8909不支持独立的外部编解码器。

2.8 确认音频特定的GPIO已正确配置

表2-1所示为针对MSM8909音频子系统中定义的GPIO，用于确保软件已根据QTI建议进行了正确配置。该配置应按第2.1节中所述由QTI CE硬件团队进行审查。

表2-1 MSM8909 GPIO

引脚编号	引脚名称	GPIO编号	备注
音频编解码器接口			
AB28	CDC_PDM0_RX2	GPIO_64	音频编解码器PDM 0接收2
AA28	CDC_PDM0_RX1	GPIO_63	音频编解码器PDM 0接收1
AA29	CDC_PDM0_RX0	GPIO_62	音频编解码器PDM 0接收0
Y29	CDC_PDM0_TX0	GPIO_61	音频编解码器PDM 0发送0
Y29	CDC_PDM0_SYNC	GPIO_60	音频编解码器PDM 0同步
Y28	CDC_PDM0_CLK	GPIO_59	音频编解码器PDM 0时钟
数字麦克风接口			
AE2	DMIC0_CLK	GPIO_4	数字MIC0时钟
AF2	DMIC0_DATA	GPIO_5	数字MIC0数据

请参见《MSM8909/MSM8209/MSM8208器件规范》(80-NP408-1)，了解有关音频GPIO的详细信息。

确保已在dtsi文件(msm8909-pinctrl.dtsi)中定义了以下音频特定的GPIO，并按照MSM GPIO布局对上述GPIO编号进行交叉检查。上述GPIO根据msm8x16.c文件中的用例进行配置。

2.8.1 pdm GPIO的Pin_ctl定义

```
cdc-pdm-lines {
    qcom,pins = <&gp 59>, <&gp 60>, <&gp 61>,
               <&gp 62>, <&gp 63>, <&gp 64>;
    qcom,num-grp-pins = <6>;
    qcom,pin-func = <1>;
    label = "cdc-pdm-lines";
    cdc_pdm_lines_act:pdm_lines_on {
        drive-strength = <8>;
    };
    cdc_pdm_lines_sus:pdm_lines_off {
        drive-strength = <2>;
        bias-disable;
    };
};
```

2.8.2 美国(OMTP)和欧洲（非OMTP）GPIO的Pin_ctl定义

如果硬件不支持美国/欧洲间切换，则无需进行以下配置。

```
cross-conn-det {
    qcom,pins = <&gp 97>;
    qcom,num-grp-pins = <1>;
    qcom,pin-func = <0>;
    label = "cross-conn-det-sw";
    cross_conn_det_act: lines_on {
        drive-strength = <8>;
        output-low;
        bias-pull-down;
    };
    cross_conn_det_sus: lines_off {
        drive-strength = <2>;
        bias-disable;
    };
};
```

2.9 验证音频路径

对于MI2S接口，检查MI2S位和字选择时钟是否以正确频率运行以及数据线路中是否含有数据。同时，还要检查MCLK是否以配置的频率运行。

2.9.1 播放

如需验证音频播放，请在adb shell中执行以下步骤：

1. 启动音频播放
2. 使能接收编解码器路径（扬声器设备）：

```
tinymix 'RX3 MIX1 INP1' 'RX1'  
tinymix 'SPK DAC Switch' 1
```

3. 使能用于通过MI2S接口进行播放的DSP AFE：

```
tinymix 'PRI_MI2S_RX Audio Mixer MultiMedia1' 1
```

4. 播放PCM音频：

```
tinyplay <filename.wav >
```

5. 停止音频播放
6. 禁止接收编解码器路径：

```
tinymix 'RX3 MIX1 INP1' 'ZERO'  
tinymix 'SPK DAC Switch' 0
```

7. 禁止用于通过I2S接口进行音频播放的DSP AFE：

```
tinymix 'PRI_MI2S_RX Audio Mixer MultiMedia1' 0
```

2.9.2 录音

如需确认PCM能够捕捉耳机麦克风，请在adb shell中执行以下步骤：

1. 输入以下命令：

```
#Enable DSP AFE for Audio Recording over I2S
tinymix 'MultiMedia1 Mixer TERT_MI2S_TX' 1
#Enable Codec TX Path
tinymix 'DEC1 MUX' 'ADC2'
tinymix 'ADC2 MUX' 'INP2'
```

2. 启动录音功能：

```
tinycap /data/rec.wav
```

3. 禁止HeadsetX设备(AMIC2):

```
tinymix 'MultiMedia1 Mixer TERT_MI2S_TX' 0
tinymix 'DEC1 MUX' 'ZERO'
tinymix 'ADC2 MUX' 'ZERO'
```

2.9.3 语音呼叫

在语音和VoLTE用例中，仅设备控制路径涉及APPS。所有数据均在语音固件（编码器/解码器）和语音软件之间进行交换。二者均位于调制解调处理器中，且功能上不涉及APPS。

在语音呼叫用例中，采用了多模声码器服务(MVS)和语音服务(VS) API。在VoLTE用例中，采用的是IP多媒体系统(IMS)协议栈。

2.9.3.1 应用路径

在APSS端，将出现以下进程：

1. 呼叫管理模块调用音频管理器中的setMode(MODE_INCALL) API以指示音频层正在建立语音/VoLTE呼叫。
2. 呼叫管理器调用音频管理器中的setMode(MODE_NORMAL)以指示语音/VoLTE层拆除。
3. QCRIL（Qualcomm无线接口层）将通过Android电话的无线接口层(RIL)了解呼叫为语音呼叫还是VoLTE呼叫。随后调用AudioSystem::setParameters()并传递以下信息：
 - 调用能够指示语音/VoLTE会话应转换的目标状态的状态信息
 - 语音/VoLTE呼叫的会话ID

- `update_calls()`函数 – 具有能够启动呼叫(`start_call()`)、停止呼叫(`stop_call()`)、保持呼叫和更新特定呼叫会话（会话ID已传递至该函数）呼叫状态的逻辑。
 - 01-13 14:17:40.638 212 1148 D voice_extn:update_calls: INACTIVE -> ACTIVE vsid:10c01000
- `start_call()`函数 – 用于建立无主机PCM播放并捕捉传递到该函数的用例的会话。
 - 01-13 14:17:40.638 212 1148 D voice:start_call: enter usecase:voice-call
- `stop_call()`函数 – 用于拆除无主机PCM播放并捕捉传递到该函数的用例的会话。
 - 01-13 14:16:04.438 212 1152 D voice: stop_call: exit: status(0)
- 用于在DSP中建立语音/VoLTE呼叫数据路径的混音控件 – 在`device/qcom/<芯片组>/mixer_paths.xml`中列出。
- `enable_audio_route()`函数 – 设置语音/VoLTE用例的混音控件。VoLTE的混音控件可调用内核中的功能，随后内核将向DSP发送命令以建立数据路径。
- 发送至DSP用于管理语音/VoLTE会话的命令 – 由语音驱动程序发送，在内核的`msm-pcm-voice-v2.c`文件中实现。
- “Default VoLTE” 语音或 “default modem voice” 字符串 – 作为从APSS发送到DSP语音模块的命令净荷的一部分传递，用于指示命令是否特定于某一语音/VoLTE会话。

2.9.3.2 调制解调器路径

MVS/VS层可为语音应用客户端提供主要接口。该层是调制解调器协议的网关，用于建立声码器包交换和速率控制。发起或接收新的语音呼叫后，MVS/VS客户端将在通话建立前配置并启用所需声码器路径。QXDM日志中应包含以下信息：

```

Voice Recognition/Low                                12:24:22.830
mvs_module.c 03408 mvs_mod_acquire(): client = 1
Voice Recognition/Low                                12:24:22.830
mvs_module.c 03512 mvs_mod_acquire_proc(): complete - current client 1

Voice Recognition/Low                                12:24:22.830
mvs_module.c 04574 mvs_mod_enable_proc(): client=1 mode=5
Voice Recognition/Low                                12:24:22.858
mvs_module.c 04481 mvs_mod_enable_proc(): complete - current client=1 and
mode=5
  
```

MVS/VS基于单个会话，并且每次仅支持一个处于活动状态的客户端。它是核心语音驱动程序(CVD)的客户端。MVS是1X和GSM/WCDMA子系统类语音客户端的接口，而VS是仅GSM子系统类语音客户端的接口。

如需获取带有所需日志掩码的QXDM日志，请使用QCAT的声码器播放功能分别播放声码器包和PCM。只要发现存在音质问题，请按以下方法操作：

- 接收路径声码器包正常，但PCM存在问题 – 请验证语音接收调试参数
- 发送路径PCM正常，但声码器包存在问题 – 请验证语音发送调试参数。
- 接收路径声码器包存在问题 – 来自协议层的输入可能不正确。检查是否存在CRC错误。如果不存在任何错误，则可能是网络问题。
- 发送路径声码器包正常，但远端仍能听到噪音 – 请检查声码器输入语音样本。如果声码器输入中有噪音，则可能是网络问题。
- 接收或发送路径静音 – 静音可能是设备问题所致。请检查设备编解码器部件连接是否正确以及增益是否静音。同时也可能是音质问题所致，具体视声码器包和PCM播放而定。

2.9.4 MBHC

如需通过adb shell判断编解码驱动程序是否检测到耳机的插入/拔出以及按钮的按下/释放事件，请调用snd_soc_jack_new()来检查在MBHC驱动程序(wcd-mbhc-v2.c)中创建的耳机和按钮插孔。内核日志将显示事件编号（X表示按钮插孔输入事件编号，Y表示耳机插孔输入事件编号）：

```
Input: msm8909-snd-card Button Jack as /devices/platform/soc-  
audio.0/sound/card0/inputX  
Input: msm8909-snd-card Headset Jack as /devices/platform/soc-  
audio.0/sound/card0/inputY
```

2.9.4.1 耳机插入/拔出

如需检测耳机状态为插入还是拔出，请将目标设备通过USB连接至PC，并运行以下adb命令（Y表示耳机插孔输入设备事件编号）：

```
#adb shell getevent /dev/input/eventY
```

如果已插入耳机，则adb shell窗口中将显示以下消息：

```
/dev/input/eventY: 0005 0002 00000001  
/dev/input/eventY: 0005 0004 00000001 (this line is only present if headset  
has a microphone)  
/dev/input/eventY: 0000 0000 00000000
```

如果已拔出耳机，则adb shell窗口中将显示以下消息：

```
/dev/input/eventY: 0005 0002 00000000  
/dev/input/eventY: 0005 0004 00000000 (this line present only if headset  
has a microphone)  
/dev/input/eventY: 0000 0000 00000000
```

耳机的插入或拔出事件还可通过在内核日志中查找“reporting insertion”或“reporting removal”来判断。“Reporting insertion 1”表示耳机不带麦克风。“Reporting insertion 3”表示耳机带麦克风。

2.9.4.2 耳机按钮按下/释放

如需检测耳机按钮状态为按下还是释放，请将目标设备通过USB连接至PC，并运行以下adb命令（X表示耳机按钮插孔输入设备事件编号）：

```
#adb shell getevent /dev/input/eventX
```

如果已按下耳机按钮，则adb shell窗口中将显示以下消息：

```
/dev/input/eventX: 0001 0001 00000001  
/dev/input/eventX: 0000 0000 00000000
```

如果已释放耳机按钮，则adb shell窗口中将显示以下消息：

```
/dev/input/eventX: 0001 0001 00000000  
/dev/input/eventX: 0000 0000 00000000
```

耳机按钮的按下或释放事件还可通过在内核日志中查找“button pressed”或“button released”来判断。如需启用内核MBHC调试日志，请运行以下adb命令：

```
#mount -t debugfs debugfs /sys/kernel/debug  
#echo -n "file msm8x16-wcd.c +p" > /sys/kernel/debug/dynamic_debug/control  
#echo -n "file msm8x16.c +p" > /sys/kernel/debug/dynamic_debug/control  
#echo -n "file wcd-mbhc-v2.c +p" > /sys/kernel/debug/dynamic_debug/control
```

A 参考资料

文档	
Qualcomm Technologies	
应用指南：客户的软件术语	CL93-V3077-1
QRD 硬件组件验证时间表	80-VL976-6
MSM8960 语音/音频拓扑和工具概述	80-N7634-1
应用指南：DMSS 软件的音频 AGC	CL93-V2586-1
自适应输入增益音频调试指南	80-N2736-1
应用指南：增强型回声消除和噪声抑制调试	CL93-V1638-2
应用指南：Fluence 双麦克风噪声抑制和回声消除	CL93-V1638-3
应用指南：多频段音频动态范围控制(MBADRC)	80-VN476-1
MSM8909/MSM8209/MSM8208 器件规范（预告信息）	80-NP408-1
演示文稿：MSM8909/MSM8209/MSM8208 数字基带设计指南/培训幻灯片	80-NP409-5
MSM8916 外部 MI2S 接口概述	80-NL239-42