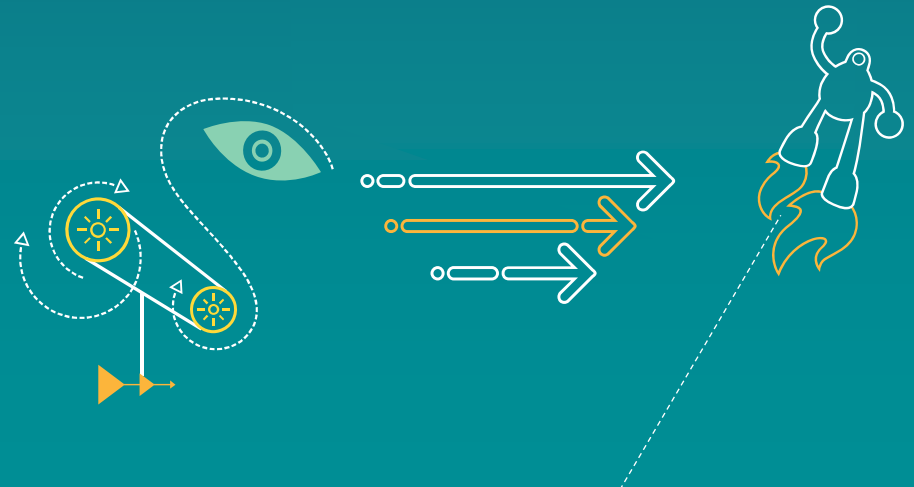

高通多媒体技术期刊 20140820



Qualcomm Technologies, Inc.



Revision History

Revision	Date	Description
A	Aug 2014	Initial release

Contents

- Audio
- Camera



Audio

Boost使能后PMIC8916烧坏问题

- 8916 PMIC芯片烧坏问题，请检查在确保基线代码有以下补丁

Issue	CR	External patch
SW change 1 : Change to Android SW driver Boost PA enable/disable sequence ensuring clock is present before the boost is enabled or disabled	673800 (Ear path boost in bypass mode register sequence needs to be updated)	https://www.codeaurora.org/cgit/quic/la/platform/vendor/qcom/msm8916_32/patch/?id=0d6cfc9b2e1c7a7e36157acef57217165696a8c0
SW change 2 : Fix the Boost PA current limit control bit that was incorrectly documented and therefore not enabled on r1.1 nor r2.0	675258 (8916: codec: PMIC heating issue requires default setting of ANALOG_CURRENT_LIMIT register)	https://www.codeaurora.org/cgit/quic/la/kernel/msm-3.10/patch/?id=0a7482d797e08177b9b560c118b71e2919d70ae4
SW change 3 : msm8x16-wcd: make default spkr_ocp hold enable	688472 (PM8916 SPK BOOST/PA broken issue)	https://www.codeaurora.org/cgit/quic/la/kernel/msm-3.10/patch/?id=5ffdd68ef6830da32a1f8fab8ecbb4f46bd95678
SW change 4 : PMIC 2.0 will be damaged at the scenario below: 1 – Audio playback with headset connected 2 – Remove the headset, the playback will pause 3 – Resume the playback, w/o the headset, the PMIC blows up.	684711 (MSM 8916 LA 1.1 : No voice on SPKR Phone when I remove the wired US headset device from phone during device switch from HPH to SPKR phone)	https://www.codeaurora.org/cgit/quic/la/kernel/msm-3.10/patch/?id=ac549a1aa1cb46382e18729a3e24142ac29bf390
SW change 5 : Regression of CR 684711 on pre-CS	691605 (regression of CR 684711 on pre-CS)	https://www.codeaurora.org/cgit/quic/la/kernel/msm-3.10/patch/?id=f13ae605c001ef362da49b9f04929241ac89b21c

- 请尽量使用PMIC2.0
- 如果项目不使用BOOST可以通过修改mixer_paths.xml来关闭
 - 删除或注释掉 <ctl name="Speaker Boost" value="ENABLE" />

8916/8939如何使能数字MIC

- 在msm8916/8939-mtp/cdp.dtsi 和msm8916/8939-pinctrl.dtsi配置数字MIC的GPIO
- 在machine driver文件msm8x16.c中添加widget的定义和函数实现，主要是用于控制GPIO
 - SND_SOC_DAPM_MIC("Digital Mic1", msm8x16_dmic_event),
 - SND_SOC_DAPM_MIC("Digital Mic2", msm8x16_dmic_event),
- 在dtsi文件如：msm8916/8939-mtp.dtsi把DMIC添加到MIC BIAS的router中配置中
 - "DMIC1", "MIC BIAS Internal1",
 - "MIC BIAS Internal1", "Digital Mic1",
 - "DMIC2", "MIC BIAS Internal1",
 - "MIC BIAS Internal1", "Digital Mic2";
- 如果是两个双数字MIC一起使用，还要注意修改msm8x16-wcd.c中的函数msm8x16_wcd_codec_enable_dmic(..){

```
switch (event) {
case SND_SOC_DAPM_PRE_PMU:
    (*dmic_clk_cnt)++;
    if (*dmic_clk_cnt == 1) {
        snd_soc_update_bits(codec, dmic_clk_reg, 0x0E, 0x02);
        --snd_soc_update_bits(codec, MSM8X16_WCD_A_CDC_TX1_DMIC_CTL, 0x07, 0x01);
        snd_soc_update_bits(codec, dmic_clk_reg, dmic_clk_en, dmic_clk_en);
    }
    break;
    ++if (dmic == 1)
    ++  snd_soc_update_bits(codec, MSM8X16_WCD_A_CDC_TX1_DMIC_CTL, 0x07, 0x01);
    ++if (dmic == 2)
    ++  snd_soc_update_bits(codec, MSM8X16_WCD_A_CDC_TX2_DMIC_CTL, 0x07, 0x01);
```
- 更多的代码细节可以提case要求patch
- 验证方法
 - tinymix commands to be used
 - tinymix 'MultiMedia1 Mixer TERT_MI2S_TX' 1
 - tinymix 'DEC1 MUX' 'DMIC1'
 - tinycap /sdcard/pcm.wav -C 1 -R 48000

8916/8939耳机按键阈值配置

- 不同的耳机其耳机按键对应的电阻不一样，对其进行配置是一个必不可少的定制化工作
 - 1，测量MIC和地之间的电阻值，按下不同的按键，对应的值不一样
 - 2，把耳机插入测量IN2P的电压，按下不同的按键，对应的电压也不一样；另外记录什么都不按下MIC工作时候的电压
 - 3，得到以上的数值后，提case，我们会针对这些值提供寄存器和参数配置
 - 4，8916和以前的平台不一样，不是直接配置IN2P的DCE电压值，而是写寄存器
 - MSM8X16_WCD_A_ANALOG_MBHC_BTN0_ZDETL_CTL (0x153)
 - MSM8X16_WCD_A_ANALOG_MBHC_BTN1_ZDETM_CTL (0x154)
 - MSM8X16_WCD_A_ANALOG_MBHC_BTN2_ZDETH_CTL (0x155)
 - MSM8X16_WCD_A_ANALOG_MBHC_BTN3_CTL (0x156)
 - MSM8X16_WCD_A_ANALOG_MBHC_BTN4_CTL (0x157)
- 寄存器的定义请参考文档[80-NK808-2X PM8916 SOFTWARE INTERFACE FOR OEMS](#)
- 函数实现
 - 写寄存器的函数实现在wcd_program_btn_threshold中
 - def_tapan_mbhc_cal里面是配值，low用来定义为current source检测模式的值，high用来定义mic_bias检测模式的值
 - btn_low[0] = 25;
 - btn_high[0] = 25;
 - btn_low[1] = 50;
 - btn_high[1] = 50;
 - btn_low[2] = 75;
 - btn_high[2] = 75;
 - btn_low[3] = 112;
 - btn_high[3] = 112;
 - btn_low[4] = 137;
 - btn_high[4] = 137;



Camera

Dump Camera输出帧

- 为什么需要dump camera 输出帧？
 - 有时候camera显示的预览帧不正常，或者拍出来的jpeg文件不正常，这时候就需要dump camera输出帧进行检查
- 如何dump camera输出帧：
 - Dump HAL层输出
 - \$ adb root
 - \$ adb shell chmod 777 /data
 - \$ adb shell setprop persist.camera.dumpimg <value>
 - Tip:
 - <value>决定了dump多少帧，怎么跳帧还有dump什么类型的帧
 - 31:16 - 决定dump多少帧。31:16可以设，如果设了就是设置dump多少帧，如果不设 (缺省0)，那么就默认dump 10 帧
 - 15:8 - 每次跳多少帧dump一帧。
 - 7: 0 - dump什么类型的帧，可以组合设置. 比如想同时dump preview帧和snapshot帧。7:0设置成 $1 + (1 \ll 2) = 5$
 - #define QCAMERA_DUMP_FRM_PREVIEW 1
 - #define QCAMERA_DUMP_FRM_VIDEO (1<<1)
 - #define QCAMERA_DUMP_FRM_SNAPSHOT (1<<2)
 - #define QCAMERA_DUMP_FRM_THUMBNAIL (1<<3)
 - #define QCAMERA_DUMP_FRM_RAW (1<<4)
 - #define QCAMERA_DUMP_FRM_JPEG (1<<5)
 - <value>必须是十进制数字,如655365. $655365 = 0xA0005$, 表示dump 10个preview帧和10个snapshot帧
 - 更多细节可以直接参看QCamera2HardwareInterface::dumpFrameToFile()函数实现

Dump Camera输出帧 - 续1

- Dump ISP输出帧
 - \$ adb root
 - \$ adb shell chmod 777 /data
 - \$ adb shell setprop persist.camera.isp.dump <value>
 - Tip
 - Dump preview 帧: <value>设置成 2
 - Dump snapshot 帧: <value>设置成 8
 - Dump Video 帧: <value>设置成 16
 - Dump类型可以组合使用, 如想同时dump preview和video frame, <value>设置成 $2+16 = 18$
 - 每次设置后, dump 10帧, 如果想再dump 10帧, 可以先设置<value>成0, 然后重新设置<value>成想设置的值
 - 更多细节可以直接参看函数isp_ch_util_dump_frame()
- Dump的帧可以在/data目录下找到
- 如何使用dump出的帧定位问题
 - Dump出的YUV数据可以用YUV查看工具打开。选YUV420格式
 - 如果Dump ISP输出帧已经有问题, 很可能sensor输出有问题, 也有可能问题来至ISP模块
 - 如果ISP dump没问题, 但是HAL层dump有问题。说明后处理模块如CPP/C2D有问题。
 - 如果ISP/HAL dump都没问题, 那问题来至camera应用或者Graphic/Display, 需要UI Framework team和Graphic/Display team进一步分析

动态打开HAL/mm-camera-interface/mm-jpeg-interface log

- MSM8916可以设置persist.camera.hal.debug.mask属性来动态打开HAL/mm-camera-interface/mm-jpeg-interface的log
- 如何设置?
 - `$ adb root`
 - `$ adb shell setprop persist.camera.hal.debug.mask <value>`
 - Tip:
 - 31:28: debug log level
 - 低27位: 模块选择 (只用了3bit)
 - `#define HAL_DEBUG_MASK_HAL` (1<<0)
 - `#define HAL_DEBUG_MASK_MM_CAMERA_INTERFACE` (1<<1)
 - `#define HAL_DEBUG_MASK_MM_JPEG_INTERFACE` (1<<2)
 - <value>默认值是268435463。268435463 = 0x10000007。即默认打开HAL/mm-camera-interface/mm-jpeg-interface HIGH level log

打开sensor module log

- 何时需要打开？
 - Sensor驱动问题
 - Resolution选择异常
 - 闪光灯开闪异常(同时要开AE Clog和内核flash驱动log)
 - 摄像头打开时间 KPI break down。可以得到power up, write init reg, setting resource等时间
- 如何打开？
 - 在文件vendor/qcom/proprietary/mm-camera/mm-camera2/media-controller/modules/sensors/module/sensor_dbg.h中进入 “#define SLOG_LOW”
 - 重新编译libmmcamera2_sensor_modules.so等模块
 - Push到设备，重启即可
 - \$ adb root
 - \$ adb remount
 - \$ adb push path/to/libmmcamera2_sensor_modules.so /system/vendor/lib/
 - \$ adb reboot
- 如何打开更多log，请参考[80-NL239-33](#) Linux_Camera_Debugging_Guide

如何得到preview 的fps

- 方法一
 - 设置“persist.debug.sf.showfps”为1
 - \$ adb root
 - \$ adb shell setprop persist.debug.sf.showfps 1
 - 抓adb log，在log中搜索关键字 PROFILE_PREVIEW_FRAMES_PER_SECOND
- 方法二
 - 不用设置“persist.debug.sf.showfps”，直接在带时间戳的adb log中过滤出同时包含mm_stream_read_msm_frame和”stream type 1”的行
 - 数出每一秒同时包含mm_stream_read_msm_frame和”stream type 1”的行数即为preview的fps

Camera SW常见问题Q/A

- 如何对拍照插值，比如0.3MP插值到1.3MP？
 - 请参考 [80-NK847-1 App_Notes_MSM_8x26_and_8x10_12_Camera_Snapshot_Upscale_Porting](#)，该文档对MSM8916同样适用. 如果希望加入的插值分辨率被所有的camera app识别到，请回退下面修改：
 - <https://www.codeaurora.org/cgit/quic/la/platform/hardware/qcom/camera/commit/?id=89c30a73902f8fc3d1ebec98eebc8bef222f9fcb>
- 想对拍照的帧通过第三方算法进行处理，但是HAL层得到的YUV数据对齐如何处理？
 - `stride = PAD_TO_SIZE(width, 64)`
 - `scanline = PAD_TO_SIZE(height, 64)`
 - `len = PAD_TO_SIZE(stride*scanline, 0)`
 - 可参考QCamera2HardwareInterface::dumpFrameToFile()处理帧数据
- 引入第三方算法做图像后处理拍出来的照片为何有灰色条纹？
 - 需在第三方算法处理后，JPEG压缩前调用cleanInvalidateCache()清一下缓存
 - `QCameraMemory *memObj = (QCameraMemory *)main_frame->mem_info;`
 - `memObj->cleanInvalidateCache(main_frame->buf_idx);`

Camera SW常见问题Q/A - 续1

- HDR 拍出来的照片没效果，但是同一个驱动在另外一个基线上是正常的，这是为什么？
 - 需要在<sensorname>_lib.c中正确设置sensor_num_HDR_frame_skip。
 - 如果sensor_set_aec_init_settings()如下，把sensor_num_HDR_frame_skip设置成和sensor_num_frame_skip一样的值：

```
sensor_set_aec_init_settings()  
{  
    ...  
    ctrl->s_data->ae_bracket_info.apply_index = 0;  
    ctrl->s_data->ae_bracket_info.sof_counter = aec_data->valid_entries;  
    ...  
}
```

- 如果sensor_set_aec_init_settings()如下，把sensor_num_HDR_frame_skip设置成sensor_num_frame_skip-1

```
sensor_set_aec_init_settings()  
{  
    ..  
    ctrl->s_data->ae_bracket_info.apply_index = 1;  
    ctrl->s_data->ae_bracket_info.sof_counter = aec_data->valid_entries - 1;  
    ...  
}
```

Questions?

You may also submit questions to:

<https://support.cdmatech.com>

