
高通多媒体技术期刊 20150401



Qualcomm Technologies, Inc.

Confidential and Proprietary – Qualcomm Technologies, Inc.

机密和专有信息——高通技术股份有限公司



Confidential and Proprietary – Qualcomm Technologies, Inc.

Confidential and Proprietary – Qualcomm Technologies, Inc.

NO PUBLIC DISCLOSURE PERMITTED: Please report postings of this document on public servers or web sites to: DocCtrlAgent@qualcomm.com. **禁止公开：**如在公共服务器或网站上发现本文档，请报告至：DocCtrlAgent@qualcomm.com.

Restricted Distribution: Not to be distributed to anyone who is not an employee of either Qualcomm or its affiliated without the express approval of Qualcomm's Configuration Management. **限制分发：**未经高通配置管理部门的明示批准，不得发布给任何非高通或高通附属及关联公司员工的人。 Not to be used, copied, reproduced, or modified in whole or in part, nor its contents revealed in any manner to others without the express written permission of Qualcomm Technologies, Inc. 未经高通技术股份有限公司明示的书面允许，不得使用、复印、复制、或修改全部或部分文档，不得以任何形式向他人透露其内容。

The user of this documentation acknowledges and agrees that any Chinese text and/or translation herein shall be for reference purposes only and that in the event of any conflict between the English text and/or version and the Chinese text and/or version, the English text and/or version shall be controlling. 本文档的用户知悉并同意中文文本和/或翻译仅供参考之目的，如英文文本和/或版本和中文文本和/或版本之间存在冲突，以英文文本和/或版本为准。

This document contains confidential and proprietary information and must be shredded when discarded. 未经高通明示的书面允许，不得使用、复印、复制全部或部分文档，不得以任何形式向他人透露其内容。本文档含有高通机密和专有信息，丢弃时必须粉碎销毁。

Qualcomm reserves the right to make changes to the product(s) or information contained herein without notice. No liability is assumed for any damages arising directly or indirectly by their use or application. The information provided in this document is provided on an "as is" basis. 高通保留未经通知即修改本文档中提及的产品或信息的权利。本公司对使用或应用本文档所产生的直接或间接损失概不负责。本文档中的信息为基于现状所提供，使用风险由用户自行承担。

Qualcomm is a trademark of QUALCOMM Incorporated, registered in the United States and other countries. All QUALCOMM Incorporated trademarks are used with permission. Other product and brand names may be trademarks or registered trademarks of their respective owners. Qualcomm是高通公司在美国及其它国家注册的商标。所有高通公司的商标皆获得使用许可。其它产品和品牌名称可能为其各自所有者的商标或注册商标。

This technical data may be subject to U.S. and international export, re-export, or transfer ("export") laws. Diversion contrary to U.S. and international law is strictly prohibited. 本文档及所含技术资料可能受美国和国际出口、再出口或转移出口法律的 限制。严禁违反或偏离美国和国际的相关法律。

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

高通技术股份有限公司，美国加利福尼亚州圣地亚哥市莫豪斯路 5775 号，邮编 92121

Revision History

Revision	Date	Description
A	Apr 2015	Initial release

Note: There is no Rev. I, O, Q, S, X, or Z per Mil. standards.

内容

- Display
 - 屏幕唤醒速度（延时）分析-----display子系统部分
- Camera
 - Dual LED Tuning



Display

屏幕唤醒的关键函数(基于android L release)-----display子系统部分

▪ Framework部分

base\services\core\java\com\android\server\display PowerManagerService.

base\services\core\java\com\android\server\power DisplayManagerService

- 屏幕唤醒由framework 层的PowerManagerService and DisplayManagerService 负责，相应的 call stack 和关键函数如下：

android_view_SurfaceControl.cpp

```
static void nativeSetDisplayPowerMode(JNIEnv* env, jclass clazz, jobject tokenObj, jint mode) {
```

```
    ALOGD_IF_SLOW(100, "Excessive delay in setPowerMode()");
```

```
    SurfaceComposerClient::setDisplayPowerMode(token, mode);
```

```
}
```

SurfaceFlinger.cpp

```
void SurfaceFlinger::setPowerModeInternal(const sp<DisplayDevice>& hw,int mode)
```

▪ Display HAL部分

Hwc.cpp

```
static int hwc_setPowerMode(struct hwc_composer_device_1* dev, int dpy,
```

```
int mode){
```

```
    (ioctl(ctx->dpyAttr[dpy].fd, FBIOLANK, value) < 0 )
```

```
}
```

屏幕唤醒的关键函数(基于android L release)-----display子系统部分

▪ Kernel 部分

▪ mdss_fb.c

- static int mdss_fb_blank_sub(int blank_mode, struct fb_info *info, switch (blank_mode) {
- case FB_BLANK_UNBLANK:
- + start_time = ktime_get();
- pr_debug("unblank called. cur pwr state=%d\n", cur_power_state);
- ret =mdss_fb_unblank_sub(mfd); //kernel FB/display wake-up routines
- + end_time = ktime_get();
- +pr_err ("unblank called. Delay: %llu\n", (ktime_to_ms(end_time) - ktime_to_ms(start_time)));

▪ mdss_dsi_panel.c

- static int mdss_dsi_panel_on(struct mdss_panel_data *pdata)----- panel specific wake up

Dsi-panel-jdi-nt35595-1080p-video-ram.dtsi //panel config dtsi file

- qcom,mdss-dsi-on-command = [
▪ 15 01 00 00 01 00 02 34 00
▪ 05 01 00 00 78 00 02 11 00] // init cmds
- qcom,mdss-dsi-reset-sequence = <1 5>, <0 5>, <1 120>; //Reset GPIO Config

■ 相关的DTSI配置文件

- **msm8939-mdss.dtsi**
- qcom,ctrl-supply-entries {
 - #address-cells = <1>;
 - #size-cells = <0>;
 - qcom,ctrl-supply-entry@0 {
 - reg = <0>;
 - qcom,supply-name = "vdda";
 - qcom,supply-min-voltage = <1200000>;
 - qcom,supply-max-voltage = <1200000>;
 - qcom,supply-enable-load = <100000>;
 - qcom,supply-disable-load = <100>;
 - qcom,supply-post-on-sleep = <20>;
 - };
- };

屏幕唤醒的关键函数(基于android L release)-----display子系统部分

```
▪ qcom,panel-supply-entries {
▪     #address-cells = <1>;
▪     #size-cells = <0>;
▪     qcom,panel-supply-entry@0 {
▪         reg = <0>;
▪         qcom,supply-name = "vddio";
▪         qcom,supply-min-voltage = <1800000>;
▪         qcom,supply-max-voltage = <1800000>;
▪         qcom,supply-enable-load = <100000>;
▪         qcom,supply-disable-load = <100>;
▪     };
▪     qcom,panel-supply-entry@1 {
▪         reg = <1>;
▪         qcom,supply-name = "vsp";
▪         qcom,supply-min-voltage = <2850000>;
▪         qcom,supply-max-voltage = <2850000>;
▪         qcom,supply-enable-load = <100000>;
▪         qcom,supply-disable-load = <100>;
▪         qcom,supply-pre-on-sleep = <0>;
▪         qcom,supply-post-on-sleep = <1>;
▪     };
▪     qcom,panel-supply-entry@2 {
▪         reg = <2>;
▪         qcom,supply-name = "vsn";
▪         qcom,supply-min-voltage = <2850000>;
▪         qcom,supply-max-voltage = <2850000>;
▪         qcom,supply-enable-load = <100000>;
▪         qcom,supply-disable-load = <100>;
▪         qcom,supply-pre-on-sleep = <0>;
▪         qcom,supply-post-on-sleep = <40>;
▪     };
▪ }
```

屏幕唤醒的log分析(基于android L release)-----display子系统部分

Logcat

- 01-01 01:12:24.195 277 599 I qdhwcomposer: handle_blank_event: dpy:0 panel power state: 1
- 01-01 01:12:24.348 277 277 D qdhwcomposer: hwc_setPowerMode: Done setting mode 2 on display 0
- 01-01 01:12:24.349 861 1078 D SurfaceControl: Excessive delay in setPowerMode(): 493ms
493ms就是屏幕唤醒的时长。

Kmsg ,需要打开更多的log。

```
adb shell "echo -n 'file mdss_fb.c +tp'> /d/dynamic_debug/control"
adb shell "echo -n 'file mdss_dsi.c +tp'> /d/dynamic_debug/control"
adb shell "echo -n 'file mdss_mdp_intf_cmd.c +tp'> /d/dynamic_debug/control"
adb shell "echo -n 'file mdss_fb.c +tp'> /d/dynamic_debug/control"
adb shell "echo -n 'file mdss_dsi_host.c +tp'> /d/dynamic_debug/control"
adb shell "echo -n 'file mdss_mdp_intf_video.c +tp'> /d/dynamic_debug/control"
```

- [11761.712226] [281] mdss_fb_blank_sub: mdss_fb_blank+0xf0/0x104 mode:0
- --- mdss_fb_blank_sub is the Qualcomm kernel driver entry of LCD panel wake_up
- [11761.712232] [281] mdss_fb_blank_sub: unblank called. cur pwr state=0

屏幕唤醒的delay分析-----display子系统部分

屏幕唤醒 (FB unblank) 所需时间分析

a) `mdss_fb_unblank_sub()`是内核display子系统部分的入口函数

-----一般来说，Display 相应的初始化函数，已经进行了反复的优化。进一步优化的空间不大，函数本身一般不会引入额外的延时。

b) LCD panel本身需要的时延

- 1. GPIO reset 需要的时延
- 2. 初始化命令序列需要的时延
- 3. 屏幕上电序列需要的时延

1, 2, 3 一般由LCD panel vendor提供，以及帮助优化。

c) Dtsi 文件中对应的配置项。

1.

```
qcom,mdss-dsi-reset-sequence = <1 5>, <0 5>, <1 120>;
```

2.

```
qcom,mdss-dsi-on-command = [  
    15 01 00 00 01 00 02 34 00  
    05 01 00 00 78 00 02 11 00]
```

3.

```
qcom,supply-pre-on-sleep = <0>;  
qcom,supply-post-on-sleep = <40>;
```

屏幕唤醒的delay分析-----display子系统部分

- 1和2一般在LCD panel 的DTSI 文件中配置

1.

```
qcom,mdss-dsi-reset-sequence = <1 5>, <0 5>, <1 120>;
```

这个配置用来设定LCD panel reset GPIO的时序波形。5, 5, 120加在一起，就是Reset 所用时间。

2.

```
qcom,mdss-dsi-on-command = [  
    15 01 00 00 01 00 02 34 00  
    05 01 00 00 78 00 02 11 00]
```

上面的红色数字0x01,0x78就是单条初始化命令所指定的延时。把所有的初始化命令的延时，加在一起，就是初始化命令序列需要的时延。LCD panel初始化序列，一般由LCD panel vendor提供。

OEM需要和LCD panel vendor一起进行优化。

-----减少初始化命令个数。一般来说，建议尽量减少初始化命令个数。

-----减少每条命令本身的时延。

-----合并初始化命令。比如把若干条初始化命令，合并为一个gen_write命令。

3. 上电时延一般在msmxxxx-mdss.dtsi(比如msm8939-mdss.dtsi)中配置。每一路供电，都可能在上电之前和之后添加时延。

```
qcom,supply-pre-on-sleep = <0>;
```

```
qcom,supply-post-on-sleep = <40>;
```

-----减少单独供电所需的时延。

建议优化方法(基于android L release)-----display子系统部分

■ 建议的优化 (debug) 方法 :

1. 在下列函数中添加log , 确定所消耗的时间。

```
mdss_fb_unblank_sub(mfd); //kernel FB/display wake-up whole delay
mdss_dsi_panel_on(struct mdss_panel_data *pdata) //panel specific delay
```

2. 和LCD panel vendor 一起优化初始化命令所引起的延时。

-----减少初始化命令个数。一般来说, 建议尽量减少初始化命令个数。

-----减少每条命令本身的时延。

-----合并初始化命令。比如把若干条初始化命令, 合并为一个gen_write命令。

3. 和LCD panel vendor , 一起优化reset GPIO和上电序列的延时。

4. 尝试Boost CPU (阻止 CPU进入power collapse状态或其它低功耗模式)

```
adb root
```

```
adb shell "echo Y >/sys/module/lpm_levels/parameters/sleep_disabled"
```

如果以上命令可以显著降低屏幕唤醒时间, 建议与performance team一起进一步优化CPU 工作状态。

也可以在脚本中, 或者framework 层, 在屏幕唤醒时, 短时设置CPU状态。

-----由于屏幕的初始化命令是单条传输。每条命令发送完毕, 都可能引入系统线程切换。

Boost CPU会加快关联函数的运行。

-----除kernel display部分的LCD panel相关的函数运行之外, 实际上屏幕唤醒还包括framework层的相关service运行

以及相关的各种content rendering , Boost CPU 对整个系统的运行都有加速作用。

PowerManagerService.和DisplayManagerService控制整个屏幕唤醒的流程。



Camera

概述

- Qualcomm 从8939平台开始支持Dual LED Flash功能
 - 文档部分请参阅 80-NU333-1_A_Dual_LED_Flash_Tuning_Overview.pdf
 - 本期刊将在上述文档基础上讲述 Dual LED Tuning 需要注意的事项

- Dual LED Flash 的引入
 - 单闪光灯方案通常会选择一颗高色温灯
 - 在低色温的环境中，被拍摄的主体和背景环境出现混色光的情况
 - 单闪光灯工作时，由于混合光的原因，拍摄图片有偏色现象
 - 示例图片请参考 80-NU333-1
 - 双闪光灯可以很好解决单闪光灯的偏色问题
 - 双闪光灯方案中，会选用一颗高色温灯和一颗低色温等进行组合
 - 不同于单闪光的固定色温的输出，双闪灯可以通过两个灯不同电流值的组合来线性输出一定范围色温内的光
 - 双闪光灯在工作时，会输出和环境光一样色温的光，来补强环境光，从而解决单闪光和背景光色温不一致引入的偏色问题

LED 灯的挑选

- 双闪光方案中最重要的部分就是两个灯的挑选
- 为了达到双闪光的最佳效果，请一定注重闪光灯的挑选
- 如果灯没有选好，那么最终效果也是无法通过Tuning来弥补的
- 在项目camera模组确定好后，就要开始灯的挑选，因为LED vendor可能要调整灯内荧光粉的比例来适配模组达到最优的效果，而这需要一定的时间周期
- 选灯的方法在文档80-NU333-1中有详细说明

LED 灯的选择方法

- 在选灯之前，要确保LED driver可以正常工作，tuning 工程师可以通过命令行自由控制两个灯的电流值大小
- 量测过程中，需要使用两台手机，其中一台手机打闪，另一台手机进入camera应用通过log读取AWB统计值
- 为了量测的准确性，打闪的手机尽量使用主闪的电流值打闪
 - 为了防止闪光灯过热烧毁，可以打闪300ms, 关闭30ms，不断重复的方法来打闪
 - 也可以用主闪电流值的80%来打闪
- 另外一台手机，进入camera应用，在全黑环境下，100%FoV对着灰卡，通过读取下述log来获取AWB的统计值
 - simple-grey-world (rg,bg,cnt)=(%f,%f,%d)
- 量测多个电流值组合下的 AWB的统计值，从而得到这组LED灯对应的camera模组的 rg, bg响应
- 画出LED 的rg, bg 连线，最优的选择是这条连线与AWB 参考点的曲线在低色温区尽量重合，请参考80-NU333-1 第17页

Dual LED 参数的说明 (8939)

- 80-NU333-1 文档中有针对8994 平台Dual LED参数的说明，由于8939平台与8994平台 chromatix header 的版本不一样，所以两个平台 Dual LED 参数的位置略有不同，本期刊将对8939 的参数进行补充说明
- 8939中Dual LED的一部分参数放置在awb.h中

```
/* Dual Led Params */
#define DUAL_LED_HIGH_LOW_LED_RATIO (7.5f)
#define DUAL_LED_INTERSECT_SLOPE (2.0f)
#define DUAL_LED_RED_RG_ADJ (1.0f)
#define DUAL_LED_RED_BG_ADJ (1.0f)
#define DUAL_LED_BLUE_RG_ADJ (1.0f)
#define DUAL_LED_BLUE_BG_ADJ (1.0f)
#define USE_ENHANCED_K_INTERPOLATION (0)
```

```
/* Dual LED tuning data, rg/bg ratios
 * Filled with golden led module data
 * Need to match the number of entries in Chromatix */
static float dual_led_ratios[] = {
    1.0998f, 0.2901f, // rg_ratio, bg_ratio (Entry 1)
    0.9953f, 0.3404f, // rg_ratio, bg_ratio (Entry 2)
    0.9058f, 0.3859f, // rg_ratio, bg_ratio (Entry 3)
    0.8659f, 0.4075f, // rg_ratio, bg_ratio (Entry 4)
    0.7927f, 0.4468f, // rg_ratio, bg_ratio (Entry 5)
    0.7416f, 0.4751f, // rg_ratio, bg_ratio (Entry 6)
    0.6929f, 0.5022f, // rg_ratio, bg_ratio (Entry 7)
    0.6327f, 0.5371f, // rg_ratio, bg_ratio (Entry 8)
    0.6035f, 0.5542f, // rg_ratio, bg_ratio (Entry 9)
    0.5466f, 0.5893f, // rg_ratio, bg_ratio (Entry 10)
    0.4884f, 0.6219f, // rg_ratio, bg_ratio (Entry 11)
};
```

Dual LED 参数的说明 (8939)

- DUAL_LED_HIGH_LOW_LED_RATIO
 - 这个参数定义了主闪电流与预闪电流的比例，在Chromatix header中我们需要填入主闪电流的组合表，而预闪电流的大小由主闪电流除以这个比例得到
- DUAL_LED_INTERSECT_SLOPE
 - 打闪前Preview AWB的rg, bg 向 Dual LED rg, bg连线做投影，这个参数定义了投影线的斜率
- DUAL_LED_RED_RG_ADJ
- DUAL_LED_RED_BG_ADJ
- DUAL_LED_BLUE_RG_ADJ
- DUAL_LED_BLUE_BG_ADJ
 - 这四个参数定义了双闪光灯的gain adjust, 两个灯的gain adjust可以分别调节，实际使用的gain adjust会根据两个灯在打闪时的电流比例进行差值
- USE_ENHANCED_K_INTERPOLATION
 - 是否使用优化的K差值算法，对于Dual LED 方案，请配置此参数为“1”

Dual LED 参数的说明 (8939)

- 在chromatix header中，有Mix LED table 参数，这组参数定义了最多16组的双灯电流值组合。Table的大小可以调节，默认是11组。
- AWB.h中 dual_led_ratio数组为不同电流值组合下量测到的 rg bg值

```
/* Mix LED Table */
{
    0, /* Table Size */
    {
        /* Entry 0 */
        {
            0, /* CCT */
            0, /* LED1 Config */
            0, /* LED2 Config */
        },
        /* Entry 1 */
        {
            0, /* CCT */
            0, /* LED1 Config */
            0, /* LED2 Config */
        },
        /* Entry 2 */
        {
            0, /* CCT */
            0, /* LED1 Config */
            0, /* LED2 Config */
        },
    },
}
```

```
/* Dual LED tuning data, rg/bg ratios
 * Filled with golden led module data
 * Need to match the number of entries in Chromatix */
static float dual_led_ratios[] = {
    1.0998f, 0.2901f, // rg_ratio, bg_ratio (Entry 1)
    0.9953f, 0.3404f, // rg_ratio, bg_ratio (Entry 2)
    0.9058f, 0.3859f, // rg_ratio, bg_ratio (Entry 3)
    0.8659f, 0.4075f, // rg_ratio, bg_ratio (Entry 4)
    0.7927f, 0.4468f, // rg_ratio, bg_ratio (Entry 5)
    0.7416f, 0.4751f, // rg_ratio, bg_ratio (Entry 6)
    0.6929f, 0.5022f, // rg_ratio, bg_ratio (Entry 7)
    0.6327f, 0.5371f, // rg_ratio, bg_ratio (Entry 8)
    0.6035f, 0.5542f, // rg_ratio, bg_ratio (Entry 9)
    0.5466f, 0.5893f, // rg_ratio, bg_ratio (Entry 10)
    0.4884f, 0.6219f, // rg_ratio, bg_ratio (Entry 11)
};
```

Dual LED log (8939)

- awb_estimate_led_settings: led_off rg (%f), bg (%f), estimated on rg (%f), bg (%f)
 - led_off rg bg: camera preview的rg bg值
 - estimated on rg bg: led_off rg bg 对Dual LED rg bg 连线做投影后，得到的rg bg 值
- prepare for snapshot k %f led high %f led off %f
 - k: 闪光灯与环境光的比例，k 值越大，闪光灯所占比例越大
- prepare for snapshot gain adjust: led_adjust (%f, %f), ambient adjust (%f, %f)
 - led_adjust: 根据两个闪光灯电流比例算出的闪光灯的gain adjust
 - ambient adjust: 环境光的gain adjust
 - 最终的gain adjust由 led_adjust 与ambient adjust 根据k 进行差值得到

Questions?

<https://support.cdmatech.com>

