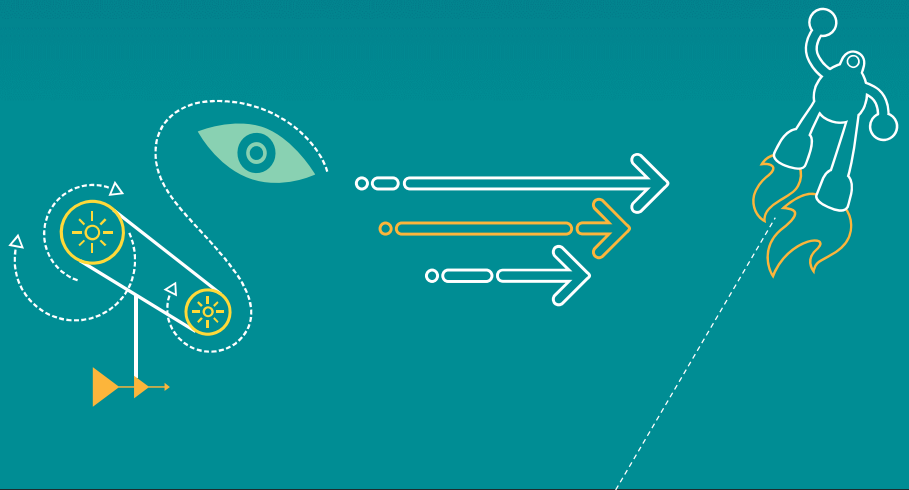
高通多媒体技术期刊　20161116

Qualcomm Technologies, Inc.



Confidential and Proprietary – Qualcomm Technologies, Inc.

Revision History

Revision	Date	Description
A	Oct. 2016	Initial release

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

内容

- Display
  - 关于 mm-pp-dpps crash 问题
  - 两类Panel bring up 问题的处理
  - 如何用Panel GC GT WPC 做白点校正（调整色温）

Display

关于 mm-pp-dpps crash 问题

如果碰到mm-pp-dpps发生crash，并有如下类似log出现，通常是由于用户写的client端代码，发完socket命令，但是没有等待系统状态返回而直接将socket关闭造成的。

```
Dpps    : AddNewDppsClient():838 New DPPS client accepted, client fd = 26, poll idx 2
Dpps    : RunPollIteration():774 Client 26 has pending read events, revents 1
Dpps    : HandleDppsClientEvent():1259 Incoming message on client fd 26: 'ad:off;0'
Dpps    : Post():49 payload cmd kAdOffCmd
Dpps    : ProcessAmbientLightThread():112 Poll unblocked ret 1
Dpps    : Read():138 client cmd kAdOffCmd, clear read pipe
Dpps    : ProcessAlMsg():395 incoming msg kAdOffCmd from queue for processing
Dpps    : RunPollIteration():774 Client 26 has pending read events, revents 17
Dpps    : ProcessAlCommands():130 command recieved kAdOffCmd
Dpps    : AdOff():1145 AD in off/Disabling state 0
Dpps    : HandleDppsClientEvent():1250 Client connection 26 has closed
Dpps    : RemoveServerClient():705 closed fd 26 poll_idx 2
--------- beginning of crash
libc    : Fatal signal 13 (SIGPIPE), code 0 in tid 726 (AL_THREAD)
```

关于 mm-pp-dpps crash 问题

```
DEBUG   : *** *** *** *** *** *** *** *** *** *** *** *** *** *** *** ***
DEBUG   : AM write failed: Broken pipe
DEBUG   : Revision: '0'
DEBUG   : ABI: 'arm64'
DEBUG   : pid: 777, tid: 726, name: AL_THREAD  >>> /system/vendor/bin/mm-pp-dpps <<<
DEBUG   : signal 13 (SIGPIPE), code 0 (SI_USER), fault addr --------
DEBUG   :     x0   fffffffffffffe0 x1   0000007fa5fff178 x2   0000000000000007 x3   0000007fa5fff098
DEBUG   :     x4   0000007fa5fff09c x5   00000000ffffffff x6   0000000000000004 x7   0000000000000000
DEBUG   :     x8   0000000000000040 x9   0000007fa822e012 x10  0000000000000000 x11  0000007fa822d000
DEBUG   :     x12  0000007fa8218090 x13  0000000000000001 x14  000000007fffffff x15  0000000000000000
DEBUG   :     x16  0000007fa8217ea0 x17  0000000000000000 x18  0000007fa822d000 x19  0000007fa822d000
DEBUG   :     x20  0000007fab56a5fc x21  0000000000000007 x22  0000007fa5fff178 x23  0d3fd53a4508e36f
DEBUG   :     x24  000000555d344ce5 x25  000000555d344d2d x26  0000000000000000 x27  0000000000000001
DEBUG   :     x28  000000555d347710 x29  0000007fa5fff0a0 x30  0000007fa821c2b4
DEBUG   :     sp   0000007fa5fff0a0 pc   0000007fab56a604  pstate 0000000080000000
DEBUG   : backtrace:
DEBUG   :     #00 pc 0000000000069604  /system/lib64/libc.so (write+8)
DEBUG   :     #01 pc 00000000000032b0  /system/vendor/lib64/libcneconn.so (write+168)
DEBUG   :     #02 pc 000000000000119c  /system/vendor/lib64/libNimsWrap.so (write+44)
DEBUG   :     #03 pc 0000000000012120  /system/vendor/bin/mm-pp-dpps
DEBUG   :     #04 pc 000000000000d830  /system/vendor/bin/mm-pp-dpps
DEBUG   :     #05 pc 000000000000f264  /system/vendor/bin/mm-pp-dpps
DEBUG   :     #06 pc 000000000001179c  /system/vendor/bin/mm-pp-dpps
DEBUG   :     #07 pc 0000000000012ce0  /system/vendor/bin/mm-pp-dpps
DEBUG   :     #09 pc 0000000000066094  /system/lib64/libc.so (_ZL15__pthread_startPv+52)
DEBUG   :     #10 pc 000000000001ef84  /system/lib64/libc.so (__start_thread+16)
```

两类Panel bring up 问题的处理

1. 如果遇到LK正常显示，启动到kernel后不显示，休眠唤醒后又正常显示。这类问题我们通常怀疑是由于kernel启动过程中的一些配置(GPIO, POWER)对LCD产生了影响造成的， 比如 LCD 需要的GPIO由于某种原因被别的模块占用控制等等。检测的最好方式是直接用示波器测量观察LCD 的POWER，reset 等硬件资源，确保其在启动过程中没有异常。

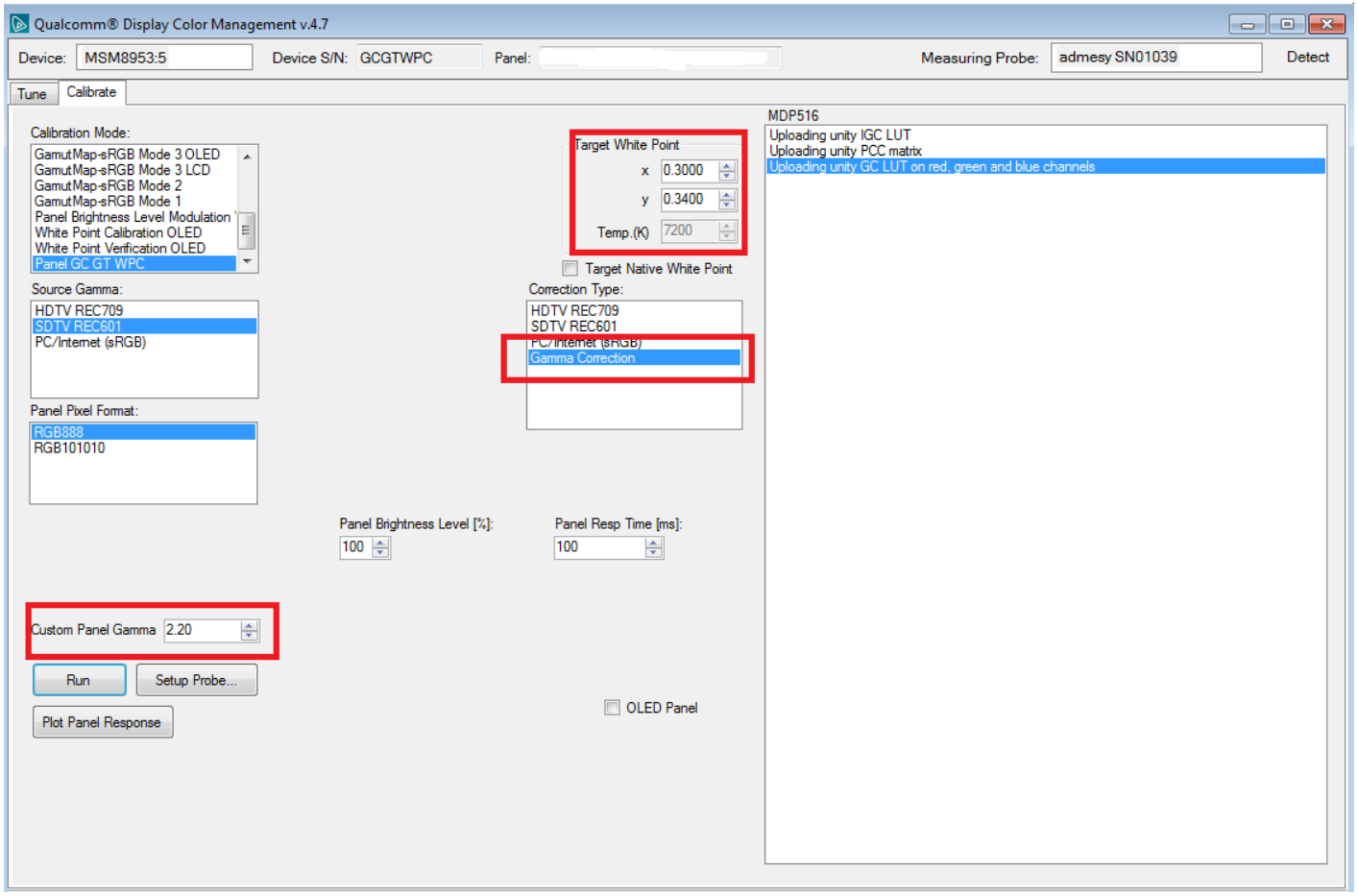
2. 如果遇到LK显示正常，Kernel第一次显示正常，休眠唤醒后不显示，这种问题通常是由于kernel的LCD配置与LK不匹配造成的， 需要我们仔细检查Kernel的DTS与LK的配置是否完全一致。

如何用Panel GC GT WPC做白点校正（调整色温）

80-Nxxxx-1 A

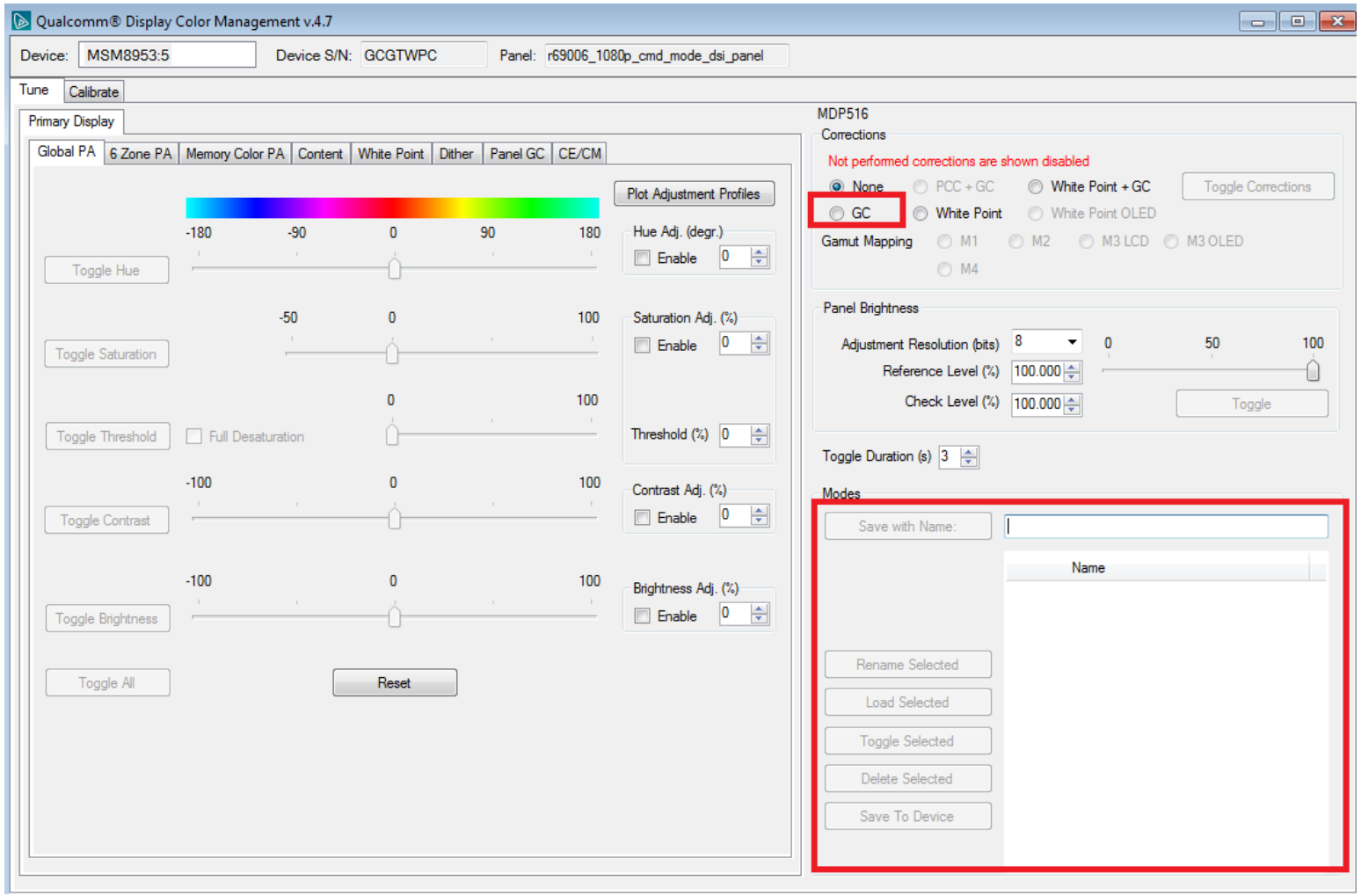
首先进行Panel GC GT WPC calibration

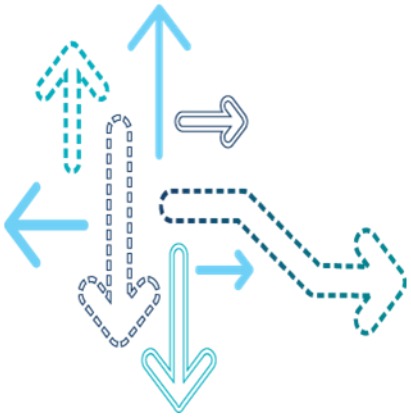
- 打开最新版本的QDCM (v4.7)，切到calibrate -> Panel GC GT WPC
- 设定“Target White Point” 为目标值
- 设定“Correction Type” 为“Gamma Correction”



载入校准值，并保存模式

- 切换到 Tune, 选中“GC” 这个correction
- 保存模式





Questions?

https://support.cdmatech.com