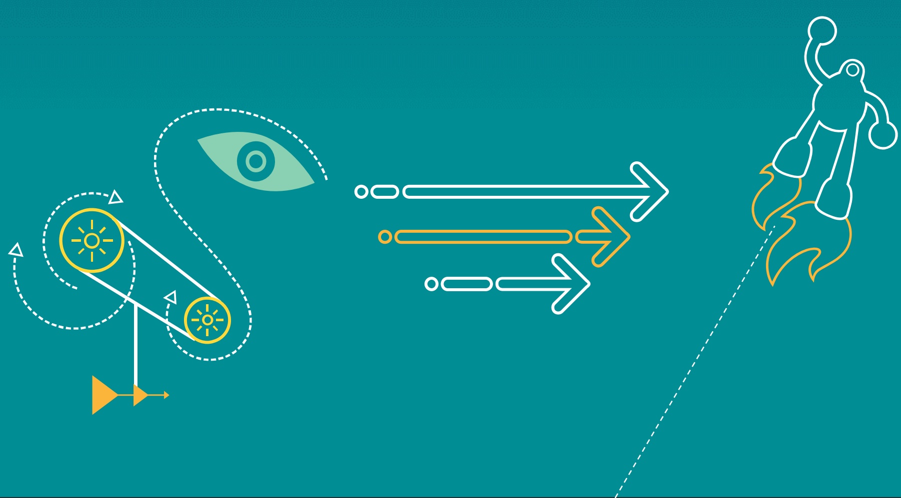
高通多媒体技术期刊 20151014

Qualcomm Technologies, Inc.



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	Oct 2015	Initial release

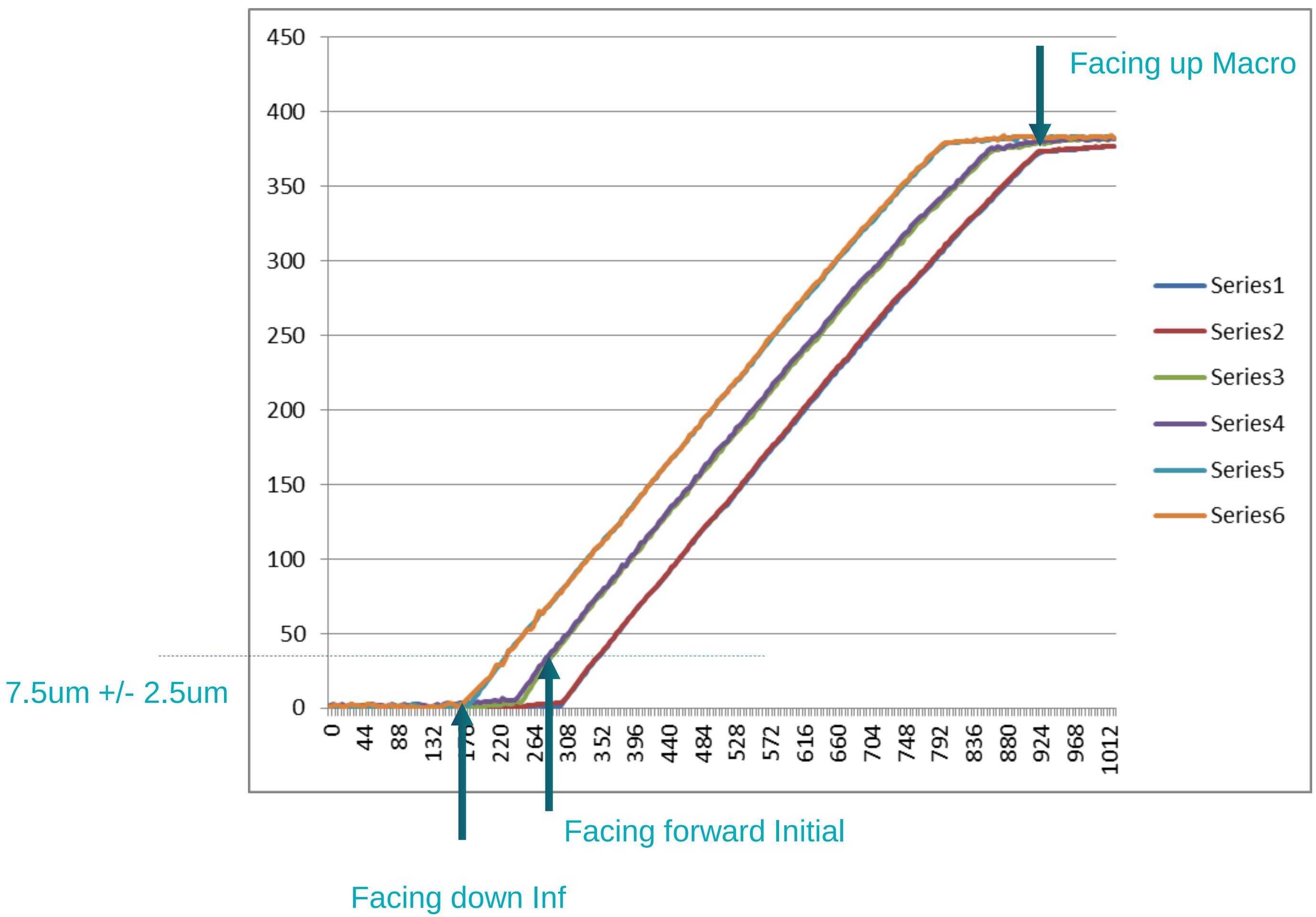
Note: There is no Rev. I, O, Q, S, X, or Z per Mil. standards.

内容

- Camera
  - AF calibration data verification
  - Hysteresis and Ringing test
  - SOP for AF consistency checking
- Display
  - FOSS介绍
  - FOSS如何使能
  - FOSS如何调试
  - 如何设置模拟的Panel

Camera

Typical Open loop DAC/Distance Diagram



Typical AF OTP data

	Direction	Focus distance
Facing down Infinity DAC code	Facing down	Hyperfocal/5m
Facing up Macro DAC code	Facing up	10cm/7cm
Facing forward Infinity DAC code	Facing forward	Hyperfocal/5m
Facing forward Macro DAC code	Facing forward	10cm/7cm
Initial current DAC code	Facing forward	Value to start moving the lens form infinity to 7.5um +/- 2.5um

- The tuning parameter need to be modified according to different type of calibration data.
- Usually module vendor will provide “Facing forward Infinity DAC code” and “Facing forward Macro DAC code” as AF calibration data.
- We will explain how to modify these parameters in following slice.

What we do in AF calibration code

- If the macro_dac and infinity_dac in OTP are all measured facing forward, we need to use INFINITY_MARGIN and MACRO_MARGIN to extend the lens moving range to cover facing_down_inf and facing_up_macro scenarios, that is new_step_bound in above code.
- If the macro_dac is measured facing up, and the infinity_dac is measured facing down, then INFINITY_MARGIN and MACRO_MARGIN could be set to 0, since the otp_setp_bound has already considered the facing_down_inf and facing_up_macro scenarios.
- Since the new_step_bound considered the facing_down_inf and facing_up_macro scenarios, the total_step should also be extended to calculate appropriate code_per_step.

```
  macro_dac = ectrl->eeprom_data.afc.macro_dac;
  infinity_dac = ectrl->eeprom_data.afc.infinity_dac;
  otp_step_bound = macro_dac - infinity_dac;
 /* adjust af_driver_ptr */
  af_driver_tune->initial_code = infinity_dac - otp_step_bound * INFINITY_MARGIN;
  new_step_bound = otp_step_bound * (1 + INFINITY_MARGIN + MACRO_MARGIN);
  af_driver_tune->region_params[0].code_per_step =
   new_step_bound / (float)total_steps * qvalue;
```

- code_per_step is another important value we need to calibrate, it presents the vcm um/DAC sensitivity. For example, if golden module’s code_per_step is 1, then 0.9 means the calibrated module has higher sensitivity than golden, and 1.1 means the calibrated module has lower sensitivity than golden.
- You could see that the new step table is calculated based on the code_per_step

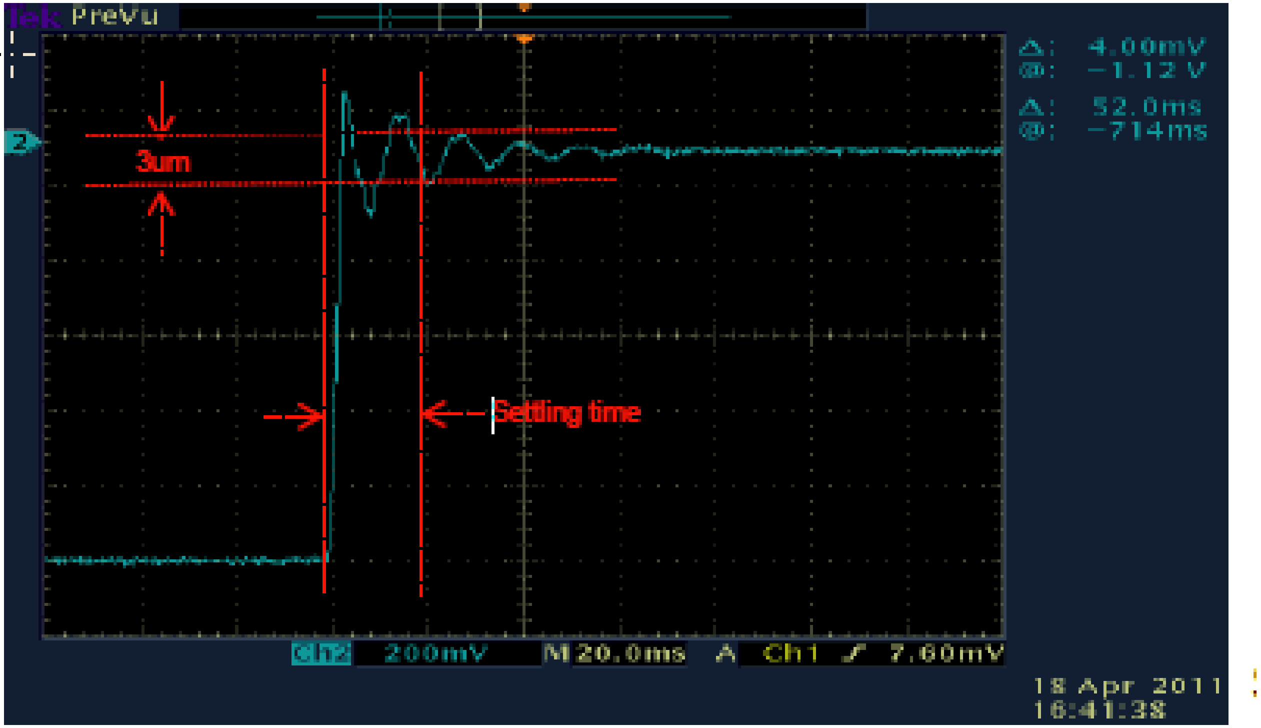
```
    for (; step_index <= step_boundary; step_index++) {
      if ( qvalue > 1 && qvalue <= MAX_QVALUE)
        cur_code = step_index * code_per_step / qvalue;
      else
        cur_code = step_index * code_per_step;
      cur_code += set_info->af_tuning_params.initial_code;
      if (cur_code < max_code_size){
        a_ctrl->step_position_table[step_index] =
          cur_code;
      } else {
        for (; step_index <
          set_info->af_tuning_params.total_steps;
          step_index++)
          a_ctrl->
            step_position_table[
            step_index] =
            max_code_size;
      }
      CDBG("step_position_table [%d] %d\n", step_index,
      a_ctrl->step_position_table[step_index]);
    }
  }
```

- For example, if your OTP calibration data are:
  - Facing forward Infinity, focus distance is 5M
  - Facing forward Macro, focus distance is 7cm
- To cover facing_down_inf and facing_up_macro scenarios
  - set INFINITY_MARGIN to 0.2
  - set MACRO_MARGIN to 0.1
- Then, the total_step should be calculated as following
  - Using golden module
  - Turn off the AF OTP calibration
  - Set the value of step_boundary bigger enough
  - Using AF_FULL_SWEEP mode to get 5M and 7cm DAC value
  - total_step = (5M_DAC – 7cm_DAC) * (1+ 0.2 + 0.1)

Hysteresis and Ringing test

Ringing

- The FV value will not be reliable if the settling time of ringing is long.
- To get rid of ringing influence on FV, we need to fine tune damping or consider adding frame delay.



Method to enable AF_FULL_SWEEP

- Two ways to switch to full sweep algorithm
  - Change AF algorithm header

Variable name	Value	Comment
af_process_type	AF_FULL_SWEEP	
num_steps_between_stat_point	4 (around 2-3 um, so can derive from um/DAC)	In af_tuning_fullsweep_t
frame_delay_norm	2	In af_tuning_fullsweep_t

  - Setprop from adb shell

```
adb root
adb wait-for-device
adb remount
adb wait-for-device
adb shell setprop debug.camera.af_fullsweep 1
```

Note: adb shell setprop debug.camera.af_fullsweep 0 is to disable full sweep

Test scene

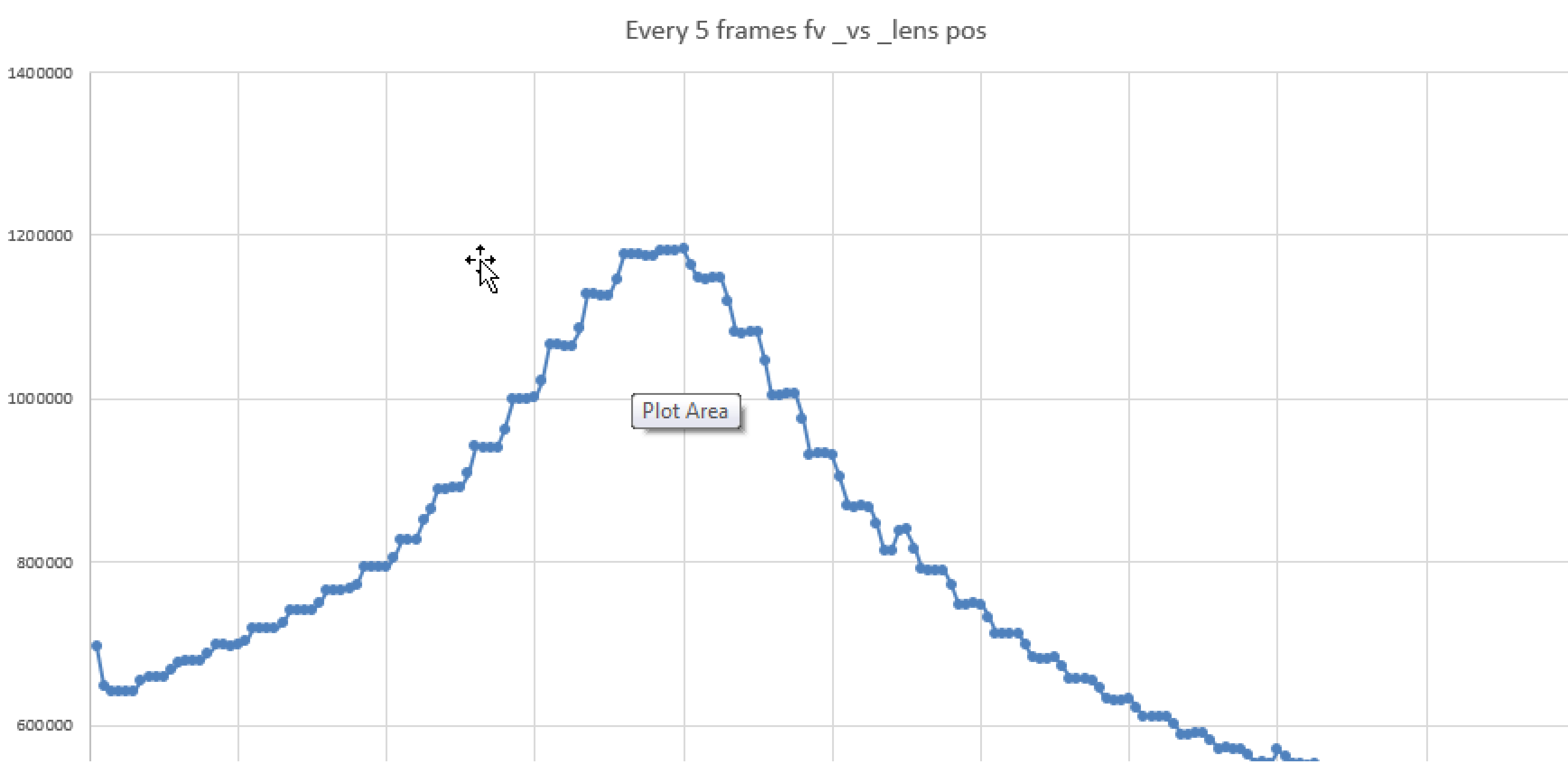
- Get FV and corresponding lens positon from AF log, and plot the data in excel table.

- Log example:
- af_process_parse_stats: fv: 23035088, fv_norm: 359769 lens_pos: 318, nfocus: 1 cur_luma: 43.000000

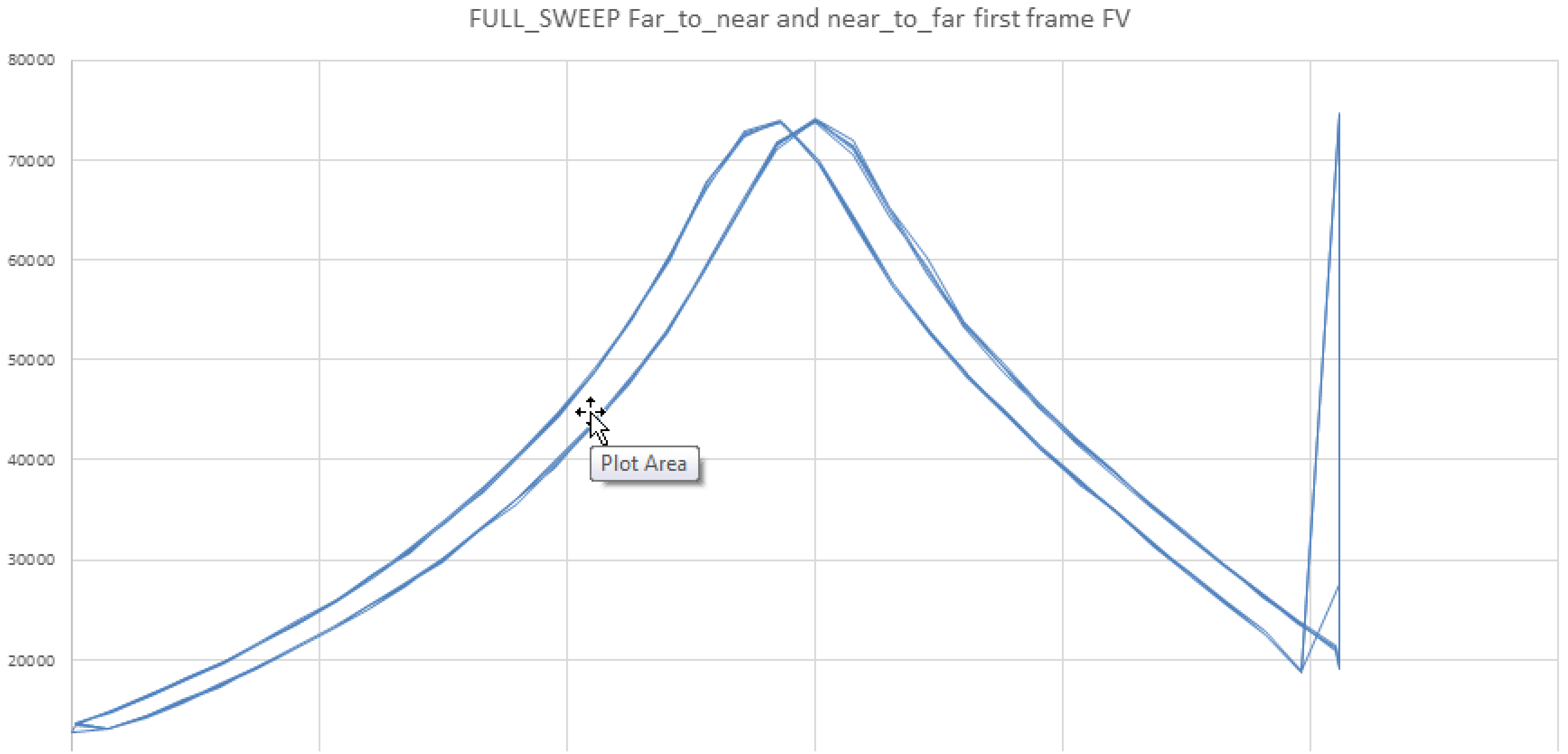
Frame rate	Lux	Exposure time	Frame delay
30fps	Above 1000 lux	1/1000s	2~5
30fps	350 lux	1/30s	2~5
15fps	100 lux	1/15s	2~5

Every 5 frames FV vs Lens position

- From this plot, we can see that the 1st FV is different with other 4 frame's FV, we need to check the ring parameters.

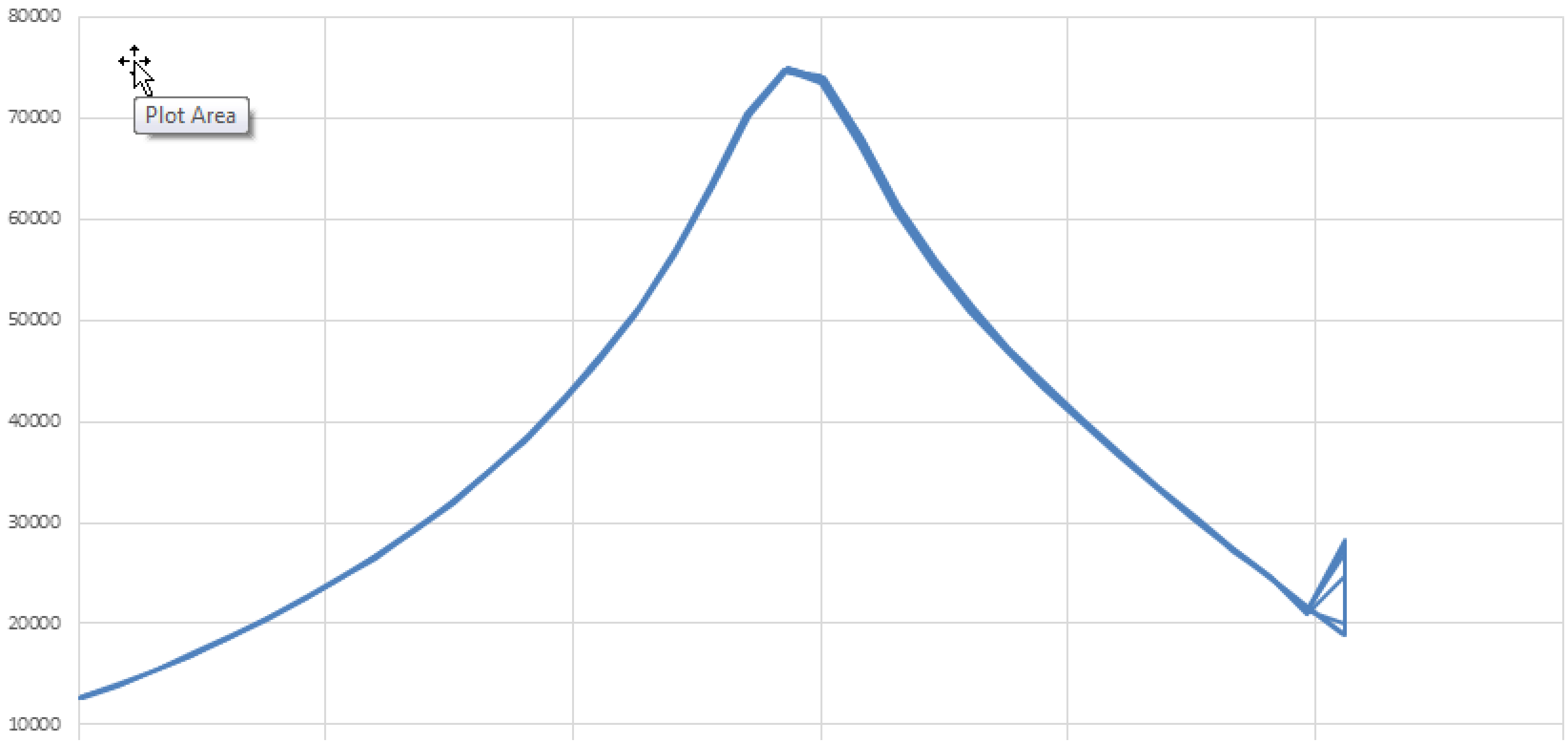


Far_to_near and near_to_far, FV vs Lens position



From this plot, we can see that the far_to_near and near_to_far FV values are mismatch, that means the lens may have Hysteresis problem

Far_to_near and near_to_far, FV vs Lens position



far_to_near and near_to_far FV values are perfect matching.

SOP for AF consistency checking

AF consistency checking

- Using tripod to take image
- Run a set of at least 30 captures of the same scene (retriggering focus every time).
- Checking the image’s sharpness, and whether the final lens position are consistent.

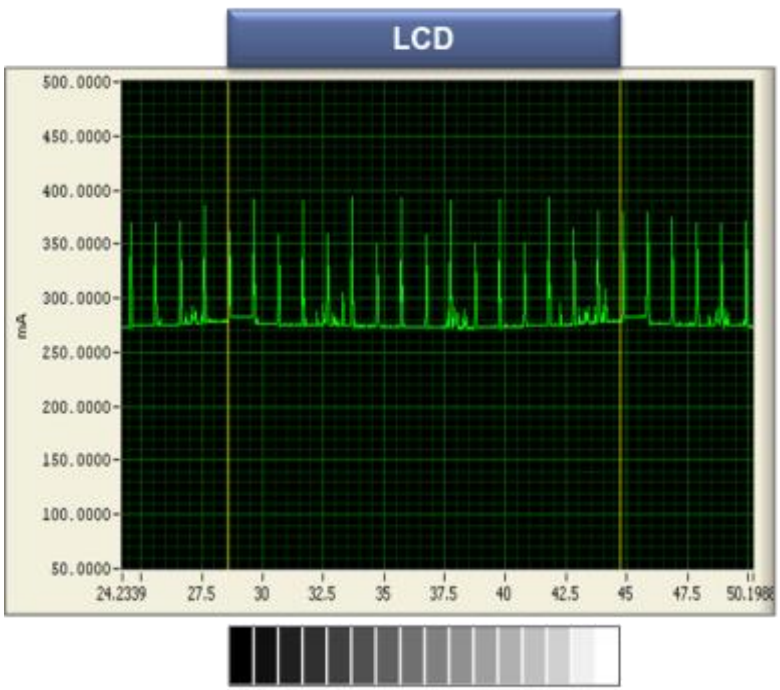
Lux	Distance
1000lux	2M
350lux	2M
20lux	2M
350lux	30cm
350lux	10cm

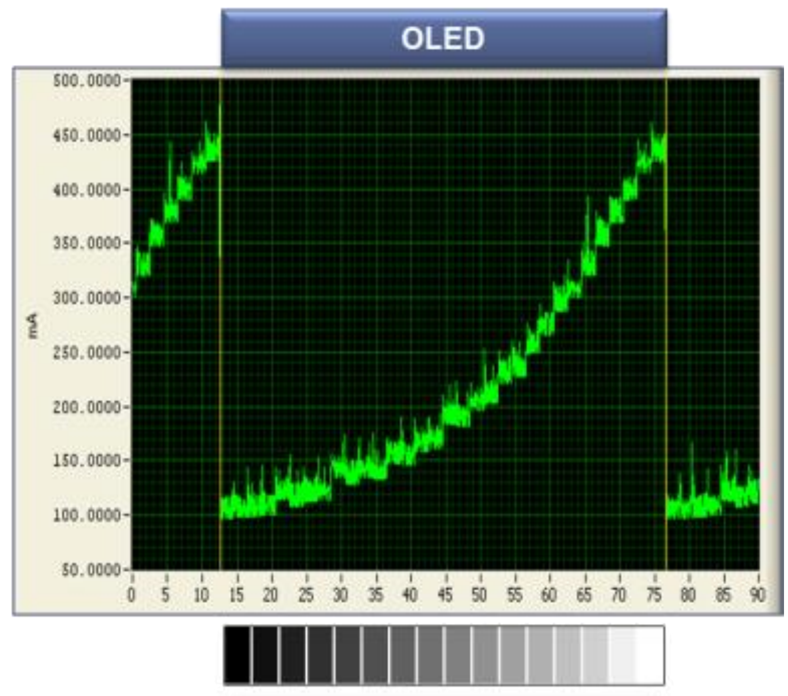
Display

FOSS (Fidelity Optimized Signal Scaling)介绍

- 对于LCD 和 OLED 两种不同的panel，为了优化功耗：
  - CABL 适用于LCD panel
  - FOSS 适用于 OLED panel

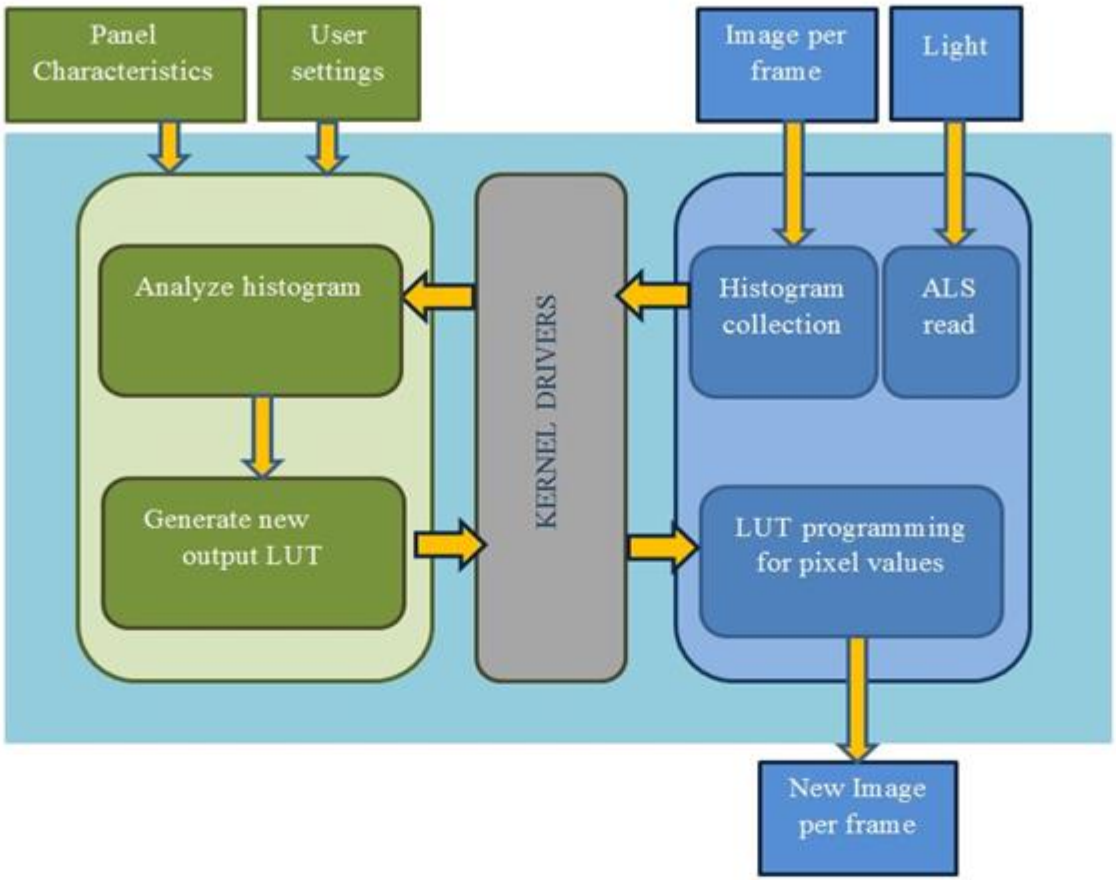
- 如下图所示，不同类型的panel的功耗表现图：





FOSS (Fidelity Optimized Signal Scaling)介绍 – 续一

- 对于OLED panel 来说，由于没有backlight，故根据显示内容的直方图来重新查表达到降低功耗，如下图所示：



- 请参见：80-NV610-94_A_FOSS_Feature_Description.pdf

FOSS (Fidelity Optimized Signal Scaling) 如何使能

- 对于FOSS功能：
  - 不需要license
  - 在8952，8976，8996平台上支持。

- 关于如何使能FOSS，如下：
  - 在build.prop 中，设置ro.qualcomm.cabl=2
  - 在build.prop 中，设置ro.qualcomm.display.paneltype=1 (for OLED)

- 注意：ro.qualcomm.display.paneltype=0 is reserved for LCD)

FOSS (Fidelity Optimized Signal Scaling)如何调试

- FOSS debugging
  - 在build.prop 中，添加debug.cabl.logs=1 去查看 FOSS是否使能
  - 在logs中，当FOSS 使能后，会看到关键字 “FOSS feature selected”
  - Backlight值不会改变
  - 当FOSS 处于active时，logs中会显示 “Active Aba features => FOSS”

- 与CABL类似，如果想得到更多信息，比如Histogram，LUT，请设置：
- debug.cabl.logs=2 in build.prop

如何设置模拟的Panel

- 对于某些场景，需要在不接panel的情况下进行测试，故要保证系统能够正常启动。
- 对于video mode panel，由于没有TE 信号，即使不接panel，系统可以正常启动。
- 但对于command mode panel，只有收到 Panel TE信号后，系统才能正常启动。如果不接panel，系统无法正常启动。
- 对于上述，增加了模拟的Software TE ，即在没有接panel时，系统可以正常启动。
- 关于此功能的具体实现，可以提case到高通。

如何设置模拟的Panel – 续一

- 对于simulate panel，这里有三种模式，如下：
  - sim - for video mode panel - simulator mode – all host reads are disabled.
  - sim-swte - for command mode panel – simulator mode – all host reads are disabled and no requirement for hardware TE (Terminator card)
  - sim-hwte - for command mode panel – simulator mode – all host reads are disabled except that it still expects TE from the hardware
- 可以使用下面的命令来选择不同的模式
  - fastboot -c “” boot boot.img // 通过command 来传递 对应的模式

如何设置模拟的Panel – 续二

- 举例如下:

- 1：首先获取kernel正常启动的log
  - Kernel command line: console=ttyHSL0,115200,n8 androidboot.console=ttyHSL0 androidboot.hardware=qcom user_debug=31 msm_rtb.filter=0x3F ehci-hcd.park=3 androidboot.bootdevice=7824900.sdhci androidboot.emmc=true androidboot.serialno=14138f45 androidboot.authorized_kernel=true androidboot.baseband=msm mdss_mdp.panel=1:dsi:0:qcom,mdss_dsi_jdi_1080p_video

- 2： 然后使用下面命令来配置不同的模式
  - **fastboot -c** "console=ttyHSL0,115200,n8 androidboot.console=ttyHSL0 androidboot.hardware=qcom user_debug=31 msm_rtb.filter=0x3F ehci-hcd.park=3 androidboot.bootdevice=7824900.sdhci androidboot.emmc=true androidboot.serialno=14138f45 androidboot.authorized_kernel=true androidboot.baseband=msm mdss_mdp.panel=<**POSSIBLE VALUES**>" **boot boot.img**

如何设置模拟的Panel – 续三

- 3：通过替换 POSSIBLE VALUES 来选择不同模式， 如下

Simulator single DSI video mode panel

- 1:dsi:0:qcom,mdss_dsi_sim_video:1:none#override:sim

Simulator single DSI cmd mode panel with SW TE

- 1:dsi:0:qcom,mdss_dsi_sim_cmd:1:none#override:sim-swte

Simulator dual DSI video mode panel

- 1:dsi:0:qcom,mdss_dsi_sim_video_0:1:qcom,mdss_dsi_sim_video_1#override:sim

Simulator dual DSI cmd mode panel with SW TE

- 1:dsi:0:qcom,mdss_dsi_sim_cmd_0:1:qcom,mdss_dsi_sim_cmd_1#override:sim-swte

Questions?

https://support.cdmatech.com