高通 CAMIF 和 OV sensor 调试总结

手机事业部 CDMA 二部 时慧钦

2008 年 2 月

【摘要】

要借用某高通平台的 camera 接口，联合 OV(OmniVision)公司的 sensor，实现手机摄像头的拍照及录像功能，需要处理两芯片、显示屏和需求配合的问题，在这个过程中遇到并解决了许多问题。

【关键词】

拍照　预览　CAMIF

一、问题的提出

新手上路，第一次见到 ov sensor，第一次认识 Qualcomm 的 CAMIF，没有任何经验，调试中遇到诸多劫难，如没有预览不到任何象素点、图像色彩不对、拍照无效区域、dispsize 设置不合适预览全屏问题、黑白模式上层设不成、预览和拍照范围不一致的问题、软件转 90 度压扁问题等等。

二、解决思路

先做基础理论的储备。

VGA ： 640x480；

QVGA ： 320x240；

YUV 格式：4：2：2

曝光控制/伽玛增益/白平衡等都是效果方面的调整。

对于象素数较大的 sensor，如 1280x1024，由于数据量较大，通常预览分辨率 640x512 拍照分辨率是 1280x1024，且拍照时的 PCLK 是预览时的 2 倍，这样可以对 VFE（video front end）来说是同样的帧速率。

Ov7670 的寄存器 0x15 的 bit6 可以切换 sensor 输出 HREF 或 HSYNC，我们用 HREF。

Camera_process_config_vfe 初始化 VFE 寄存器；

Qcamraw_set_header 设置 sensor 帧头；

代码分层：

层	Drivers	services	Oem 层	App 层
代码位置	camsensor	camera	Oemcamera.c	Qcamera.c/Qcamcorder.c

Trace32命令:data.image addr. 640. 480. /GS8，可方便的看某buffer地址中的图片，判断取到的预览图片内容和最终显示的屏上的差异。

camera_qdsp_cb 是收到帧等事件的回调，根据预览和拍照的不同需要，QDSP 会送来不同的格式，本例中拍照格式是 YUV422PS，预览格式是 RGB565LE。

要得到需要的帧率，需要给 sensor 寄存器设置时加入空白象素和空白行。对于 ov7670，0x92/0x93 加入空白行个数。

帧率的计算按照以下公式：

fps*(640+144)*(510+x)*2 = 12M(Pclk)

其中 x 是空行数，{0x92, 0x00},{0x93, 0x00}时，x 为 0，fps 为 15；{0x92, 0x19},{0x93, 0x01}时，x 为 281（0x119），fps 为 9.67。

如下图 1 是 x 为 0 时的时序图：

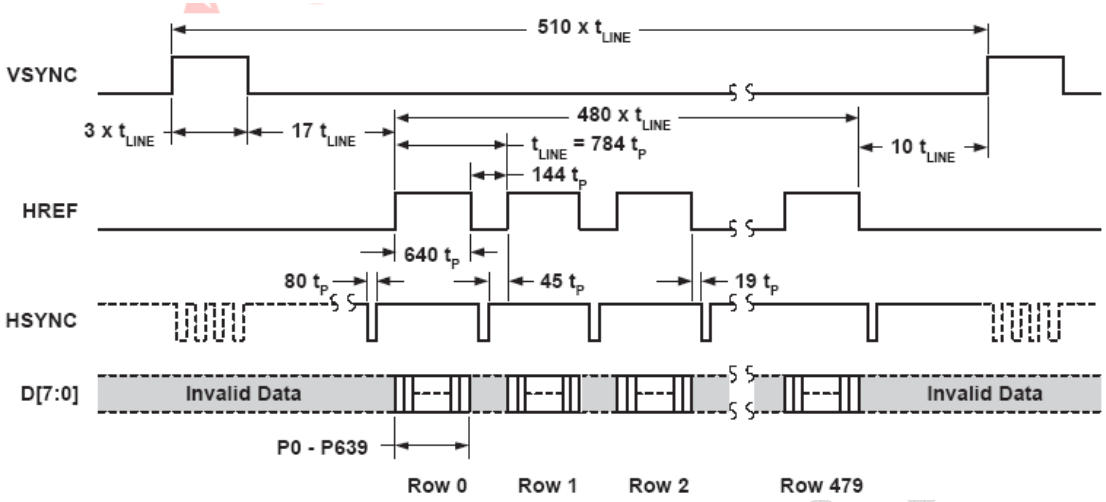


图 1　VGA Frame Timing

三、实践情况

3.1 预览

首先关注寄存器设置是否成功，测试发现写完寄存器，再读出的值和写的部分不同，因为某些寄存器是在自动刷新的。

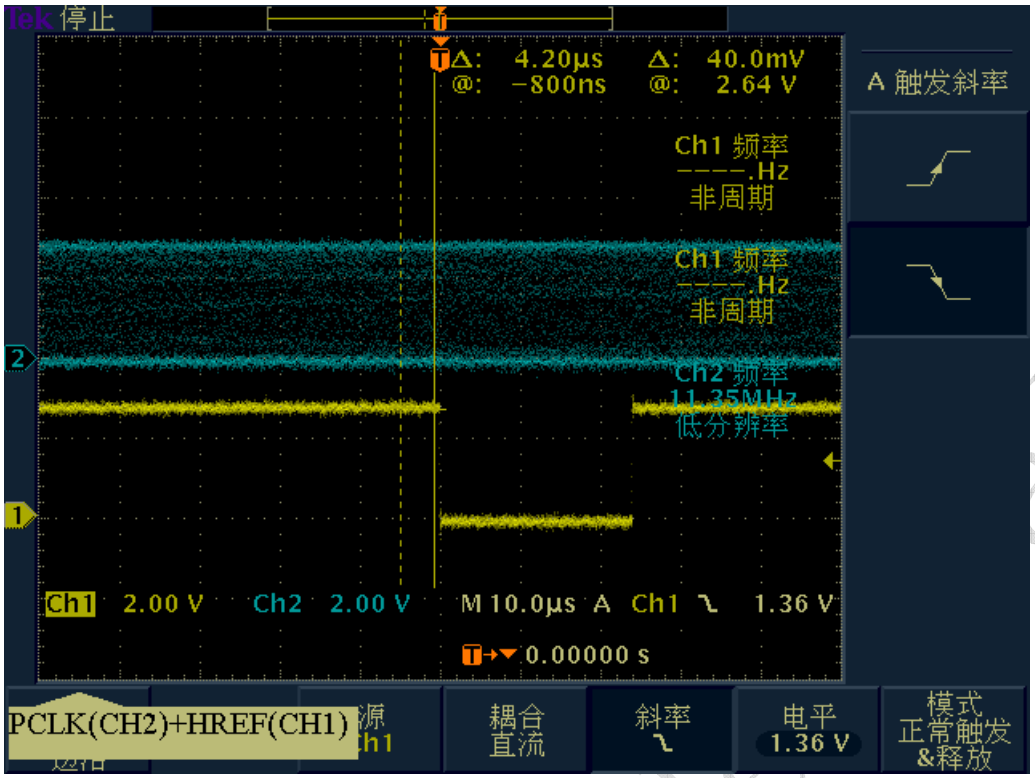
对于 sensor，只要供电正常，且有 MCLK，就应该有行场同步及 PCLK 信号。开始没有测到信号，后查出来同步信号在传输过程中由于某管脚对地短路，衰减了。

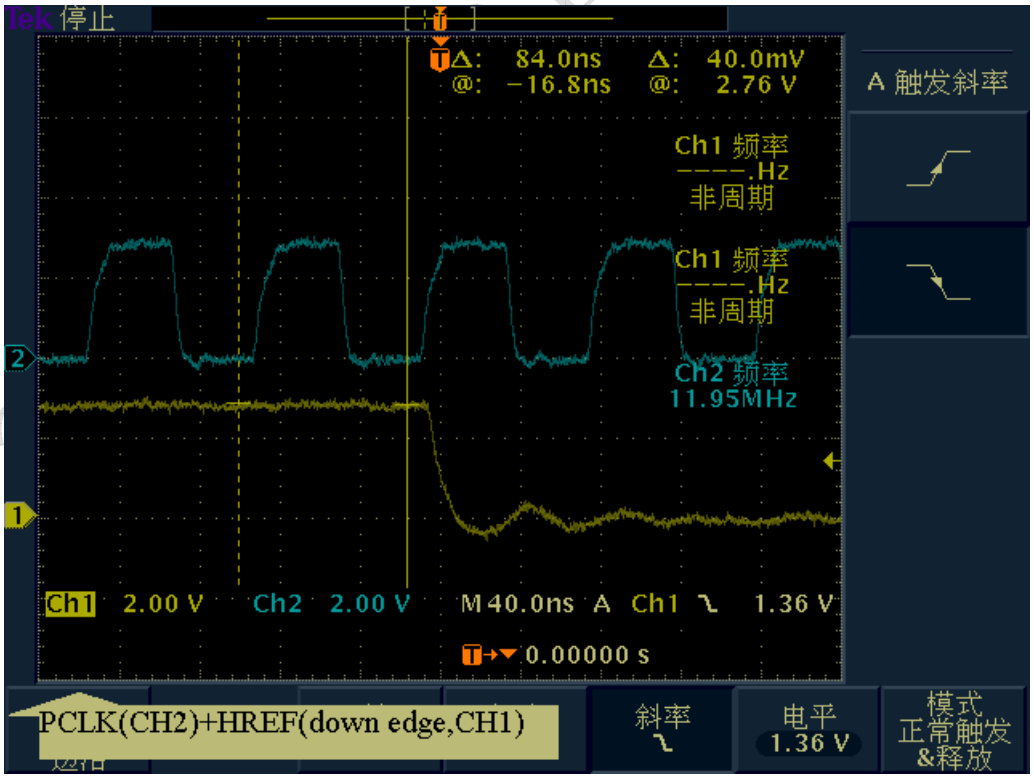
要保证代码中主芯片和 sensor 侧象素数（宽、高）、同步信号极性（高低电平）和采样频率（PCLK）设置一致，才能预览。主芯片通过 CAMIF 接口的寄存器设置。在 camera_process_config_vfe 中写入。

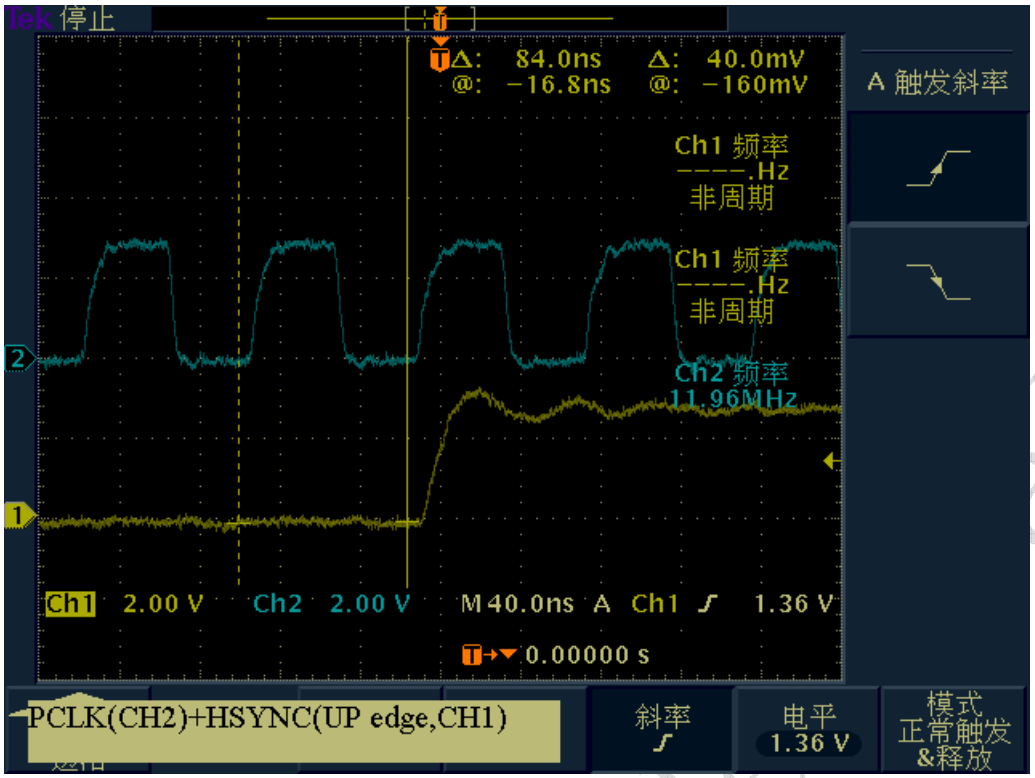
软件设置如下：

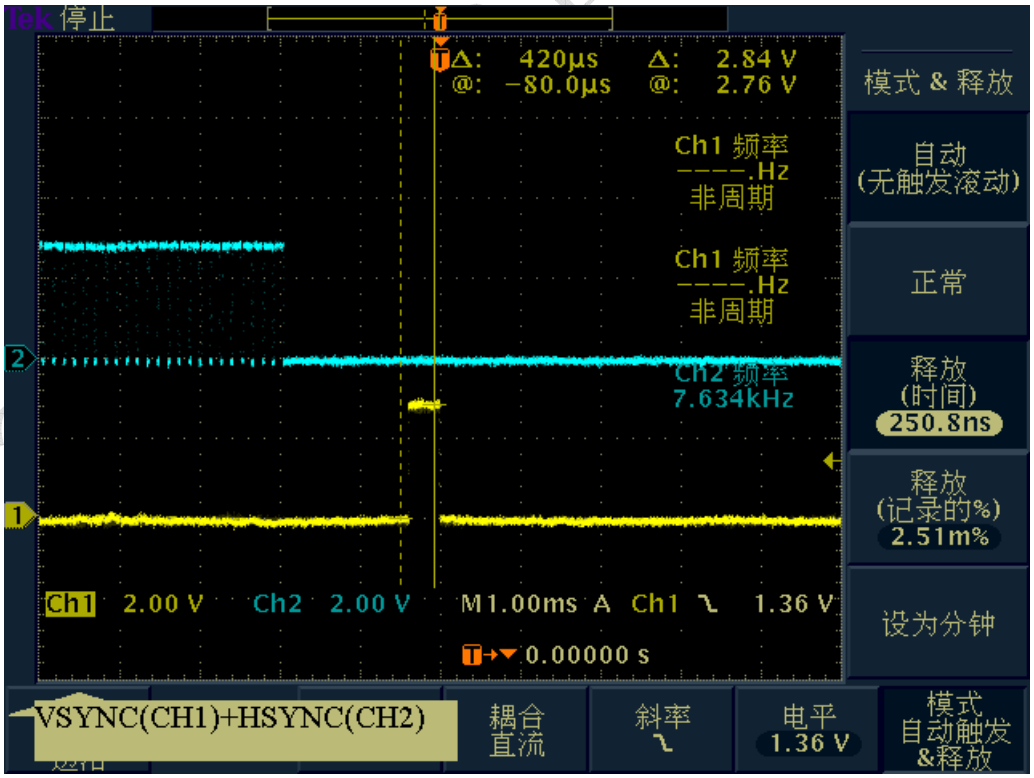
```
/* CAMIF sync config */
camsensor_info->camif_sync.sync_mode = 0; /* APS sync mode */
camsensor_info->camif_sync.vsync_polarity = 0; /* VSYNC active high */
camsensor_info->camif_sync.hsync_polarity = 0; /* HSYNC active high */
```

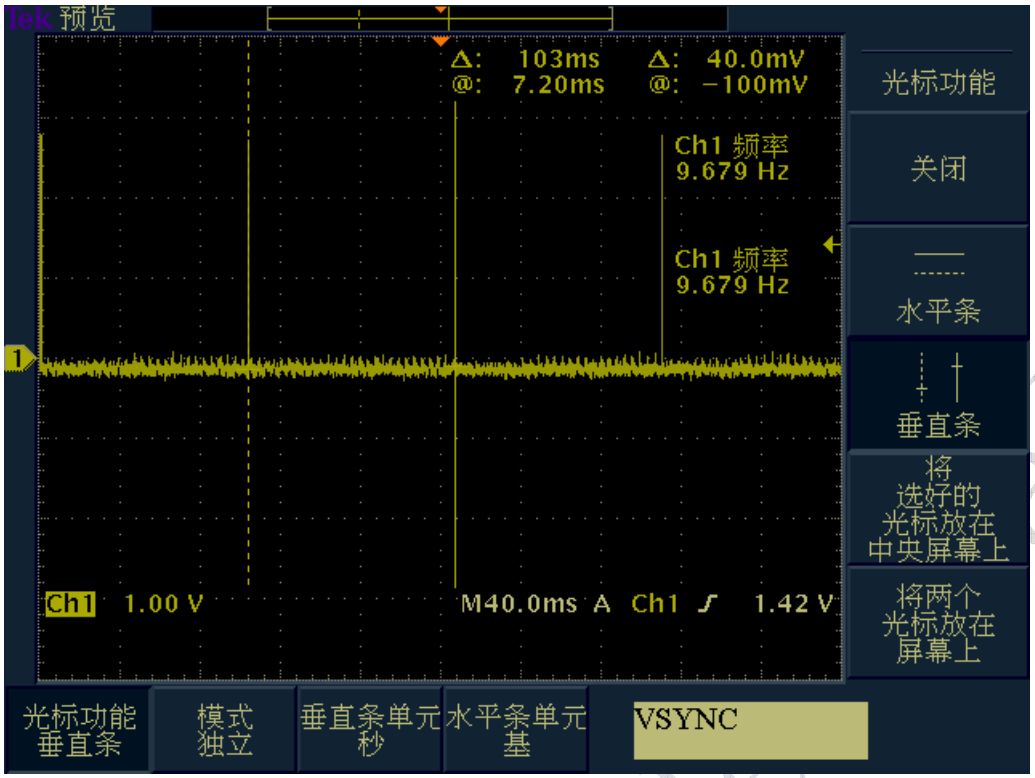
测出 9.679fps 时的波形如下图：


Tek 停止
Δ: 4.20µs Δ: 40.0mV
@: −800ns @: 2.64 V
A 触发斜率
Ch1 频率
----.Hz
非周期
Ch1 频率
----.Hz
非周期
Ch2 频率
11.35MHz
低分辨率
Ch1 2.00 V Ch2 2.00 V M 10.0µs A Ch1 1.36 V
0.00000 s
PCLK(CH2)+HREF(CH1)
耦合
直流
斜率
电平
1.36 V
模式
正常触发
&释放


Tek 停止
Δ: 84.0ns Δ: 40.0mV
@: −16.8ns @: 2.76 V
A 触发斜率
Ch1 频率
----.Hz
非周期
Ch1 频率
----.Hz
非周期
Ch2 频率
11.95MHz
Ch1 2.00 V Ch2 2.00 V M 40.0ns A Ch1 1.36 V
0.00000 s
PCLK(CH2)+HREF(down edge,CH1)
斜率
电平
1.36 V
模式
正常触发
&释放







15fps 的时序如下：

VSYNC High:66ms（约 510=17+480+10 lines）Low 394us　周期约 66ms 即 15fps;

HSYNC High:106us (约 1280 clks) low 24.4us 周期约 130us;

VSYNC 上升沿到 HSYNC 上升沿的 time :1.98ms，约 17 lines;

HSYNC 下降沿到 VSYNC 下降沿的 time :1.57ms，约 10 lines;

PCLK 是 12MHz.

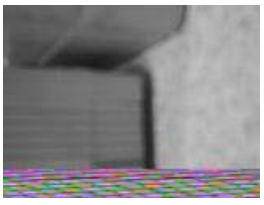
HREF、VSYNC 都用做同步信号，高通新近平台上通过采样同步信号得到数据，有效行前边的无效行不需精确设置，只要同步信号电平给定即可。检测计数到够一帧才会产生中断数据。

查完一切状态正常，但从 log 文件可以看看主芯片还是是未接到 sensor 的帧；

后再测信号，发现 HREF 信号实际板子上没传到 CPU 子板上。

总结经验：硬件调试时注意，信号测试要逐级测试，从输出一直到输入点上，考虑硬件通路可能存在的传输问题。硬件电路容易有对地短路的点，如遇到信号传输异常的。可以断电侧对地电阻，确认是否对地短路。

3.2 拍照无效区域的问题



Dispsize 设为 216x160 可以在 128*160 的屏上全屏预览，但拍出的照片有一条杂色区域，如图所示，这个问题在默认 Dispsize 设置 128x96 时也有。

30 万像素，数据量比较小，预览和拍照可以用同一分辨率，所以 snapshot 不用重设寄存器，驱动中重设寄存器引起了拍照无效区域的问题，去掉解决。

3.4 预览和拍照范围不一致的问题

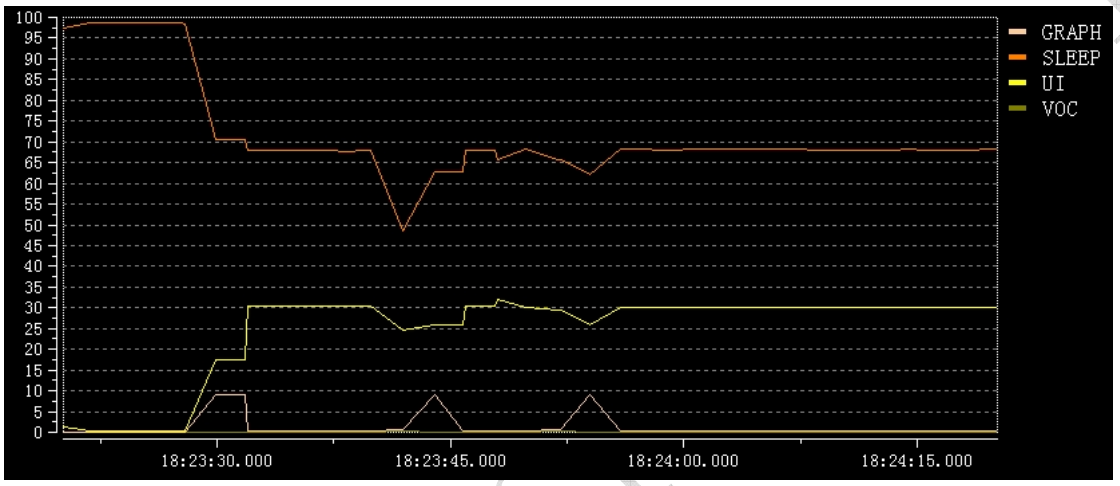
CAMIF 输出的是一个横长的图像，在竖长的屏上预览需要裁减，因此造成了用户看到的和拍到的明显不一致，为解决这个问题，需要把 sensor 从横着放置改为竖直放置，但是所用高通平台没有一个 MDP，即图像旋转的硬件加速器，软件上的旋转可能会增加负担，对此，做了必要的验证工作。

从原理上看，不旋转和旋转 90 度的处理算法没有明显差异。

现象上看，软件令 camera_default_rotation =90，camera 预览和拍照及 camcorder 预览及录下的视频都顺时针转了 90 度。

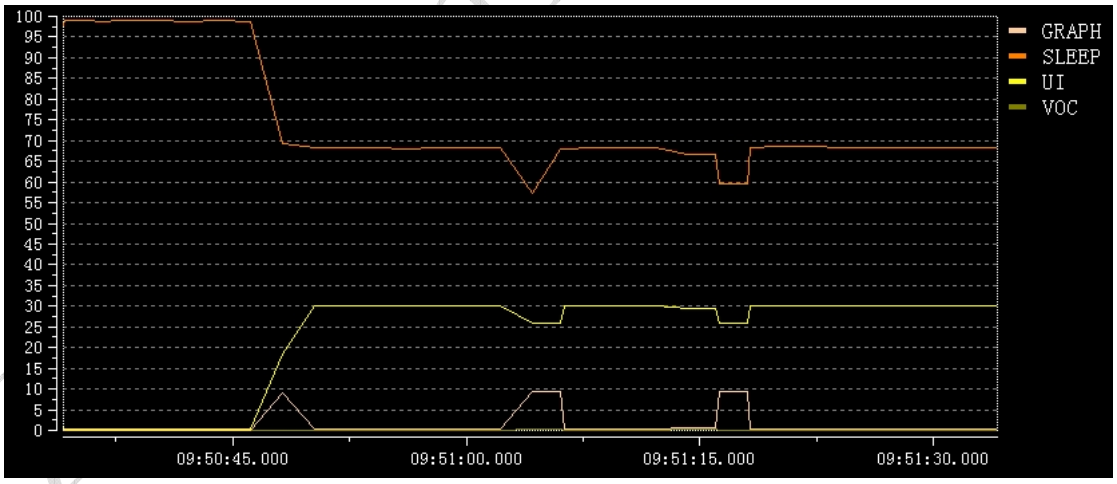
用 task profiling 跟踪，不旋转和旋转 90 度占用 CPU 时间一样。详如下图：

旋转：0 度； display size: 216x160； Image size :160*120



九宫格 QCAMERA 预览 拍照 拍照 QCAMERA 预览

旋转：90 度； display size: 216x160； Image size :160*120



九宫格 QCAMERA 预览 拍照 拍照 QCAMERA 预览

3.5 拍小尺寸照片压扁问题

软件和硬件都转过 90 度之后，拍出来的大尺寸照片和预览只有了微小差异，但小照片却被压扁了，从理论上分析，宽高比等和屏同样很协调，怎么也想不出道理。

最终用仿真器仔细跟踪拍照过程中各参数的设置，找到了原因：预览的 CAMIF window 根据

display_aspect_ratio 不同分别基于宽或高来计算，拍照的 CAMIF window 也需要这么处理，否则看到的和拍到的会不同。

因此在代码上加了类似的处理。

四、效果评价

如此这般，处理完一系列小的绊脚石之后，手机 camera 的功能比较理想的满足了需求，圆满达到设计目的。

五、推广建议

此实践对类似高通 CAMIF 上外加 ov sensor 开发 camera 的功能的同事都有参考意义。

参考资料

OV7670_DS 20(1.3)-2
80-V6212-1_F_CAMIF_Qcamera

—— 完 ——