

密级状态：绝密() 秘密() 内部资料() 公开(√)

Camera for RockChip

文件状态： <input type="checkbox"/> 草稿 <input type="checkbox"/> 正式发布 <input checked="" type="checkbox"/> 正在修改	文件标识：	Company-Project-RD-UR
	当前版本：	3.0.2
	作 者：	邓达龙、钟以崇
	完成日期：	2012-11-07

RockChip Camera 联系人：

钟以崇
邓达龙

zyc@rock-chips.com
ddl@rock-chips.com

历史版本

版本	日期	描述	作者	审核
V1.0	2011-3-25	建立文档, 主要介绍 32kernel sensor 驱动移植的注意事项	邓达龙	
V1.1	2011-5-17	Camera 驱动根据 sensor 分辨率进行内存优化说明	邓达龙	
V1.2	2011-6-20	Camera 支持 USB 摄像头	邓达龙	
V1.3	2011-9-28	Camera CTS 部分说明	邓达龙	
V1.4	2011-10-12	增加摄像头方向说明	邓达龙	
V1.5	2011-10-26	Camera 数码变焦支持	邓达龙	
V2.0	2011-12-20	Android4.0 以及 kernel 3.0 说明	邓达龙 洪慧斌	
V2.1	2012-02-01	测试帧率说明	邓达龙	
V2.2	2012-02-07	4.0.3 ICS cts 测试项补充说明 (CameraGLTest、 SystemFeaturesTest 、 CamcorderProfileTest)	邓达龙	
V2.3	2012-02-21	针对开发商培训增加部分硬件调试内容	邓达龙	
V2.4	2012-3-22	1 增加软件支持最多 6 个 sensor 的配置说明; 2 CameraHal v0.2.3/v0.2.4, Camera driver v0.1.7/0.1.8	邓达龙	
V2.5	2012-3-29	1. CameraHal v0.2.7, Camera driver v0.1.a	邓达龙	
V2.6	2012-4-5	1. CameraHal v0.2.8 关于 ICS cts 以下 2 项 : testPreviewFpsRange 、 testCameraToSurfaceTextureMetadata 测试在某些机器概率性测试不过的说明 2. android 4.0 media_profile.xml 针对 Camera 驱动 0.1.7 及其以上版本的说明	邓达龙	

v2.7 v2.7.1	2012-4-20	增加对 rk30 kernel board 文件的配置说明	钟以崇	
v2.7.2	2012-4-20	1. CameraHal v0.2.a 关于 ICS cts 以下 2 项 : testPreviewFpsRange 、 testCameraToSurfaceTextureMetadata 测试在某些机器概率性测试不过的说明	邓达龙	
v2.7.3	2012-5-18	1、RK30 cif 电源控制说明; 2、rk30 camera driver v0.2.b 用户自定义 sensor 序列说明; 3、版本约定说明; 4、模组与驱动 flip 及 mirror 匹配说明; 5、Camera_test v1.1 版本	钟以崇	
V2.7.4	2012-6-18	1、ICS r3 cts 测试, 单个前置摄像头测试说明 ; 2、sensor 支持列表更新;	邓达龙	
V3.0.0	2012-8-10	1、 CameraHal v0.3.5 版本对概率性 testSetPreviewTexturePreviewCallback 测试失败, 导致后续 camera 测试全部失败; 2、 全文梳理;	邓达龙	
V3.0.1	2012-11-3	1、 CameraHal v0.3.17 版本针对视频通话远端图像镜像的说明; 2、 增加说明各个 sensor i2c 地址; 3、 增加插值说明	邓达龙 钟以崇	

目 录

目 录.....	4
1 目录说明.....	6
2 camera 驱动结构框图.....	7
3 Camera Host (VIP Controller) 驱动简介.....	7
4 sensor 驱动简介.....	9
4.4.2.2 RK30 Kernel Board 文件配置以及 sensor 驱动移植说明.....	15
4.5 各公司 sensor 不同点.....	17
6 当前 RK29SDK/RK30SDK 支持的 SOC_CAMERA 型号.....	20
7. USB 摄像头支持说明.....	21
8 android camera 模块配置注意点.....	27
8.1 DV 分辨率设置.....	27
8.1.1 android 2.3 media_profile.xml.....	27
8.1.2 android 4.0 media_profile.xml.....	29
8.2 4.0. Panorama(全景拍照) and FaceLock (人脸解锁)	31
9 android camera 模块各项目 CTS 测试注意事项.....	31
testPreviewFpsRange 测试.....	31
9.1.1 android 2.3 版本 testPreviewFpsRange.....	31
9.1.2 android 4.0.3 版本 testPreviewFpsRange.....	32
android.hardware.cts.CameraGLTest 测试.....	33
android.hardware.cts. SystemFeaturesTest 测试.....	35
android.hardware.cts. CamcorderProfileTest 测试.....	35
单个前置摄像头 CTS 测试注意事项.....	35
10 android camera 摄像头模组方向说明.....	36
GC0308(i2c addr: 0x42):.....	38
Gc0309(i2c addr: 0x42):	38
Gc0329(i2c addr: 0x62):	38
Gc2015(i2c addr: 0x60):	39
Gc2035(i2c addr: 0x78):	39
Gt2005(i2c addr: 0x78):	39
Hi253(i2c addr: 0x40):	39
Hi704(i2c addr: 0x60):	40
Mt9d112(i2c addr: 0x7a/0x78):	40
Nt99250(i2c addr: 0x6c):	41
Ov2640(i2c addr: 0x60):	41
Ov2655(i2c addr: 0x60):	41
Ov2659(i2c addr: 0x60):	42
Ov3640(i2c addr: 0x78):	43
Ov5642(i2c addr: 0x78):	43
Ov7670:	44
Ov7675(i2c addr: 0x78):	45
Sid103b(i2c addr: 0x37):	45

11 Camera Digital Zoom.....	46
12 Camera Memory.....	47
13 Camera 模块各源码版本规则说明.....	48
14 android4.0 预览垂直及水平镜像问题说明.....	48
15 Camera_test 测试程序使用说明.....	49
16 针对视频通话远端图像镜像问题说明.....	49
17 Camera 插值说 明.....	50

1 目录说明

RK CAMERA 相关驱动文件目录，下面简单对文件结构简单说明如下：

drivers/media/video:

- |__ rk29_camera_oneframe.c VIP/CIF Driver, vip/cif 控制器单帧模式
- |__ rk30_camera_oneframe.c
- |__ rk30_camera_pingpong.c
- |__ ov2655.c ov5642.c ov2659.c ov5640.c OV 公司 sensor 驱动
- |__ mt9p111.c mt9d112.c mt9m112.c Micron(Aptina)公司 sensor 驱动
- |__ s5k6aa.c s5k5ca.c Samsung 公司 sensor 驱动
- |__ soc_camera.c soc_camera.h soc_camera 设备驱动
- |__ v4l2-xxxxx.c v4l2 设备驱动
- |__ rk29_camera.c RK29/RK30 camera IO 及设备注册相关代码
- |__ rk30_camera.c

arch/arm/plat-rk/plat:

- |__ rk_camera.c IO 操作代码

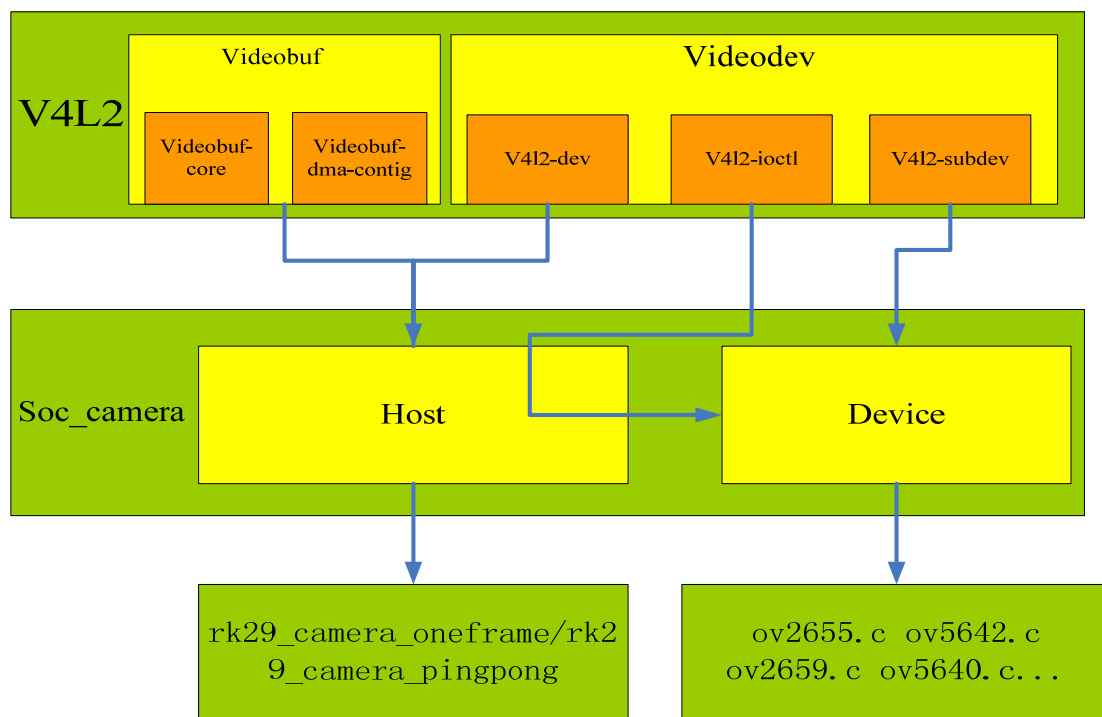
arch/arm/plat-rk/plat/include:

- |__ rk_camera.h RK camera 共用定义头文件

arch/arm/mach-rkxx:

- |__ board_rk29sdk.c 板级配置文件
- |__ include/mach/include/rk29_camera.h 各芯片平台 camera 模块头文件
- |__ include/mach/include/rk30_camera.h

2 camera 驱动结构框图



3 Camera Host (VIP Controller) 驱动简介

Camera Host 驱动主要实现如下:

1、videobuf 回调以及控制; 参考文档

2、VIP Controller 设置;

3、Camera 休眠唤醒;

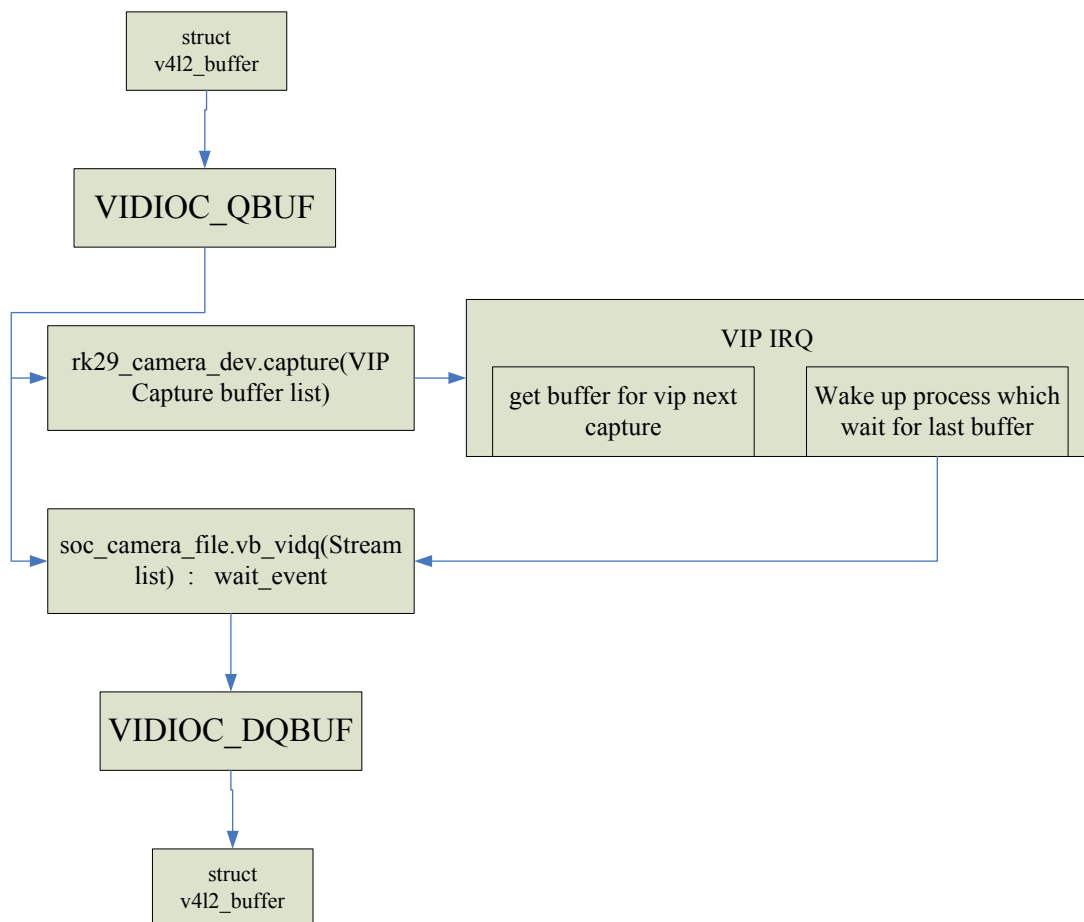
4、IPP scale 控制;

3.1 videobuf 回调以及控制

由参考文档得知, 使用 videobuf 机制必须实现 videobuf_queue_ops 结构体中的 4 个回调函数。驱动对 videobuf 的控制流程如下:



G:\参考资料\
Andriod & Linux\R



3.2 VIP Controller 设置

VIP 控制器的设置主要涉及：VIP 各工作时钟控制、VIP 输出时钟（Sensor 工作时钟）输出、VIP 采集时序极性控制；

以上控制主要集中在以下函数中：

rk29_camera_set_bus_param;

rk29_camera_setup_format;

rk29_camera_set_fmt;

3.3 Camera 休眠唤醒；

VIP 控制器在休眠唤醒中需要寄存器进行备份设置，在唤醒时将备份的寄存器值重新恢复到寄存器中。

3.4 IPP Scale 控制（RK2918/RK30XX）

Sensor 能够输出的分辨率不一定能够完全满足用户的需求，这个时候就必须对 sensor 的输出图像进行 scale。

在使用到 IPP scale 时，vip 采集的 buf 就不能是用户提供的 videobuf，vip 驱动必须获取一段 buf 作为采集用，采集结束后将该段 buf 中的数据利用 IPP 进行 scale 处理，同时将输出到用户指定的 videobuf 中，最后唤醒因获取该 videobuf 而睡眠的进程。IPP 处理必须在内核线程中进行处理。

IPP 操作还相关到数码变焦的实现，参见“Camera Digital Zoom”一节。

4 sensor 驱动简介

Sensor 驱动功能基本类似，所以在驱动中具备多个配置宏，下面针对这些配置宏说明如下

4.1 在驱动文件开头，为了针对不同 sensor 的区别，列举了以下配置宏：

```
#define SENSOR_NAME RK29_CAM_SENSOR_MT9P111      sensor 名字
#define SENSOR_V4L2_IDENT V4L2_IDENT_MT9P111     sensor 在 v4l2 中的编号，参
                                                    考 v4l2-chip-ident.h

#define SENSOR_ID 0x96                            sensor 识别号
#define SENSOR_ID_REG SEQUENCE_END               sensor id 寄存器地址
#define SENSOR_RESET_REG 0x0010                  sensor reset 寄存器地址
#define SENSOR_RESET_VAL 0x0115                  sensor reset 值
#define SENSOR_MIN_WIDTH 176                     sensor 驱动支持的最小分辨率
#define SENSOR_MIN_HEIGHT 144
#define SENSOR_MAX_WIDTH_REAL 2592               sensor 驱动实际支持最大分辨率
#define SENSOR_MAX_HEIGHT_REAL 1944
#if defined(CONFIG_SOC_CAMERA_OV5642_INTERPOLATION_8M)
#define SENSOR_MAX_WIDTH 3264                    sensor 驱动插值后达到的最大分辨率
#define SENSOR_MAX_HEIGHT 2448
#else
#define SENSOR_MAX_WIDTH SENSOR_MAX_WIDTH_REAL
#define SENSOR_MAX_HEIGHT SENSOR_MAX_HEIGHT_REAL
#endif
#define SENSOR_INIT_WIDTH 320                     sensor 驱动初始化后的默认分辨率
#define SENSOR_INIT_HEIGHT 240
#define SENSOR_INIT_WINSEQADR sensor_qvga        初始化默认分辨率对应的分辨率
                                                    序列
#define SENSOR_INIT_PIXFMT V4L2_PIX_FMT_UYVY    初始化默认输出的数据格式

#define CONFIG_SENSOR_WhiteBalance 1             是否支持白平衡，支持需要提供
相应白平衡序列，填充至 sensor_WhiteB_XXX[]相关数组中；

#define CONFIG_SENSOR_Brightness 0               是否支持亮度调节，支持需要提供相应
亮度序列，填充至 sensor_BrightnessXXX[]相关数组中；

#define CONFIG_SENSOR_Contrast 0                 是否支持对比度调节，支持需要提供
相应的序列，填充至 sensor_ContrastXXX[]相关数组中；

#define CONFIG_SENSOR_Saturation 0              是否支持色彩饱和度调节，支持需要
提供相应的序列，填充至 sensor_SaturationXXX[]相关数组中；
```

`#define CONFIG_SENSOR_Effect` 1 是否支持效果调节，支持需要提供相应序列，填充至 `sensor_Effect_XXXX[]` 相关数组中；

`#define CONFIG_SENSOR_Scene` 0 是否支持场景调节，支持需要提供相应序列，填充至 `sensor_SceneXXX[]` 相关数组中；

`#define CONFIG_SENSOR_DigitalZoom` 0 是否支持数码变焦，支持需要提供相应序列，填充至 `sensor_ZoomXXX[]` 相关数组中；

`#define CONFIG_SENSOR_Focus` 0 是否支持光学调焦，支持需要编写 `sensor_set_focus()` 函数的相关代码；注意 AF 的实现针对不同 sensor 有相当的差别，具体实现可以参考 `ov5642.c mt9p111.c ov5640.c`

`#define CONFIG_SENSOR_Exposure` 0 是否支持曝光调节，支持需要提供相应序列，填充至 `sensor_ExposureXXX[]` 相关数组中；

`#define CONFIG_SENSOR_Flash` 0 是否支持闪光灯，支持需要编写 `sensor_set_flash ()` 函数的相关代码；

`#define CONFIG_SENSOR_Mirror` 0 是否支持水平镜像调节，支持需要提供相关序列，填充至 `sensor_MirrorXXX[]` 相关数组中；

`#define CONFIG_SENSOR_Flip` 0 是否支持垂直镜像调节，支持需要提供相关序列，填充至 `sensor_FlipXXX[]` 相关数组中；

以上配置宏，都必须根据具体的 sensor 进行配置；

在编写新 sensor 驱动还需要注意一点，将 `rk_camera.h` 中添加该 sensor 必要的信息宏定义：

```
#define RK29_CAM_SENSOR_XXXXX XXXXX 定义该 sensor 识别字符
#define RK29_CAM_SENSOR_NAME_xxx "xxx" 定义该 sensor 识别字符的字符串
#define xxxxx_FULL_RESOLUTION 定义该 sensor 全分辨率尺寸，该定义将影响到
board - xxx.c 对于 camera pmem 的内存大小规划，具体代码详见 rk_camera.c 以及
board-rkxxx.c 文件
```

```

00003: #ifndef PMEM_CAM_SIZE
00004: #ifndef CONFIG_VIDEO_RK29
00005: /*----- Camera Sensor Fixed Macro Begin -----*/
00006: // Below Macro is fixed, programmer don't change it!!!!
00007: #define CONS(a,b) a##b
00008: #define CONS(a,b) _CONS(a,b)
00009:
00010: #define _STR(x) #x
00011: #define STR(x) _STR(x)
00012: #define STR(x) _STR(x)
00013:
00014: #if (CONFIG_SENSOR_IIC_ADDR_0 != 0x00)
00015: #define PMEM_SENSOR_FULL_RESOLUTION_0 CONS(CONFIG_SENSOR_0, _FULL_RESOLUTION)
00016:
00017: #ifdef CONFIG_SENSOR_CIF_INDEX_0
00018: #define SENSOR_CIF_BUSID_0 CONS(RK_CAM_PLATFORM_DEV_ID_, CONFIG_SENSOR_CIF_INDEX_0)
00019: #else
00020: #define SENSOR_CIF_BUSID_0 RK29_CAM_PLATFORM_DEV_ID
00021: #endif
00022:
00023: #if !(PMEM_SENSOR_FULL_RESOLUTION_0)
00024: #undef PMEM_SENSOR_FULL_RESOLUTION_0
00025: #define PMEM_SENSOR_FULL_RESOLUTION_0 RK_CAM_SUPPORT_RESOLUTION
00026: #endif
00027:
00028: #if (SENSOR_CIF_BUSID_0 == RK_CAM_PLATFORM_DEV_ID_0)
00029: #define PMEM_SENSOR_FULL_RESOLUTION_CIF_0 PMEM_SENSOR_FULL_RESOLUTION_0
00030: #define PMEM_SENSOR_FULL_RESOLUTION_CIF_1 0
00031: #else
00032: #define PMEM_SENSOR_FULL_RESOLUTION_CIF_1 PMEM_SENSOR_FULL_RESOLUTION_0
00033: #define PMEM_SENSOR_FULL_RESOLUTION_CIF_0 0
00034: #endif
00035: #else
00036: #define PMEM_SENSOR_FULL_RESOLUTION_CIF_0 0x00
00037: #define PMEM_SENSOR_FULL_RESOLUTION_CIF_1 0x00

```

4.2 sensor 驱动序列注意点：

sensor 的输出数据格式在初始化序列中设置相关控制寄存器实现。所以针对输出数据格式必须满足 SENSOR_BUS_PARAM 宏的相关定义；

该宏定义确定 sensor 数据的数据采集边沿、行场同步信号的采集边沿 这些定义与 sensor 实际输出不一致的话，导致 VIP 控制器直接采集不到数据

4.3 针对不同的 sensor 型号，sensor 驱动文件需要修改的函数如下：

1)、控制寄存器读写函数：

```

static int sensor_write(struct i2c_client *client, u8 reg, u8 val);
static int sensor_read(struct i2c_client *client, u8 reg, u8 *val);

```

2)、sensor 的复位操作：

sensor_init、sensor_video_probe 函数中有相关复位操作，需要针对不同 sensor 进行编写；

3)、sensor 识别号确认：

sensor_video_probe 函数中有读取 sensor 识别号来确认驱动是否与硬件 sensor 相符，相关代码需要针对不同 sensor 进行编写；

4)、分辨率设置函数：

sensor_s_fmt 函数必须根据不同 sensor 支持的分辨率设置进行编写；

4.4 Board-XXX.c 文件中关于 sensor 的一些板级的配置修改：

4.4.1 2.6.32 kernel board 文件配置

```

/*----- Camera Sensor Configuration Begin -----*/

```

```
#define CONFIG_SENSOR_0 RK29_CAM_SENSOR_OV5642    /* back camera sensor */
后置 Sensor 型号配置，型号定义缺少需要在 rk29_camera.h 中增加定义，相应的 sensor 驱动
文件 SENSOR_NAME 宏定义必须一致
#define CONFIG_SENSOR_IIC_ADDR_0                0x78
后置 Sensor i2c 器件地址配置（如果硬件没有连接，请将此地址定位 0x00）
#define CONFIG_SENSOR_IIC_ADAPTER_ID_0         1
后置 Sensor 连接到主控的 i2c 通道配置
#define CONFIG_SENSOR_POWER_PIN_0              INVALID_GPIO
后置 Sensor 电源控制引脚配置
#define CONFIG_SENSOR_RESET_PIN_0              INVALID_GPIO
后置 Sensor 复位引脚配置
#define CONFIG_SENSOR_POWERDN_PIN_0            RK29_PIN6_PB7
后置 Sensor Standby 引脚配置
#define CONFIG_SENSOR_FLASH_PIN_0              INVALID_GPIO
后置 Sensor 闪光灯控制引脚配置
#define CONFIG_SENSOR_POWERACTIVE_LEVEL_0 RK29_CAM_POWERACTIVE_L
后置 Sensor 电源开启的有效电平配置
#define CONFIG_SENSOR_RESEACTIVE_LEVEL_0 RK29_CAM_RESEACTIVE_L
后置 Sensor 复位有效电平配置
#define CONFIG_SENSOR_POWERDNACTIVE_LEVEL_0 RK29_CAM_POWERDNACTIVE_H
后置 Sensor Standby 有效电平配置
#define CONFIG_SENSOR_FLASHACTIVE_LEVEL_0 RK29_CAM_FLASHACTIVE_L
后置 Sensor 闪光灯开启有效电平配置

#define CONFIG_SENSOR_1 RK29_CAM_SENSOR_OV2659    /* front camera sensor */
前置 Sensor 型号配置，型号定义缺少需要在 rk29_camera.h 中增加定义，相应的 sensor 驱动
文件 SENSOR_NAME 宏定义必须一致
#define CONFIG_SENSOR_IIC_ADDR_1                0x60
前置 Sensor i2c 器件地址配置
#define CONFIG_SENSOR_IIC_ADAPTER_ID_1         1
前置 Sensor 连接到主控的 i2c 通道配置
#define CONFIG_SENSOR_POWER_PIN_1              INVALID_GPIO
前置 Sensor 电源控制引脚配置
#define CONFIG_SENSOR_RESET_PIN_1              INVALID_GPIO
前置 Sensor 复位引脚配置
#define CONFIG_SENSOR_POWERDN_PIN_1            RK29_PIN5_PD7
前置 Sensor Standby 引脚配置
#define CONFIG_SENSOR_FLASH_PIN_1              INVALID_GPIO
前置 Sensor 闪光灯控制引脚配置
#define CONFIG_SENSOR_POWERACTIVE_LEVEL_1 RK29_CAM_POWERACTIVE_L
前置 Sensor 电源开启的有效电平配置
#define CONFIG_SENSOR_RESEACTIVE_LEVEL_1 RK29_CAM_RESEACTIVE_L
前置 Sensor 复位有效电平配置
#define CONFIG_SENSOR_POWERDNACTIVE_LEVEL_1 RK29_CAM_POWERDNACTIVE_H
```

前置 Sensor Standby 有效电平配置

```
#define CONFIG_SENSOR_FLASHACTIVE_LEVEL_1 RK29_CAM_FLASHACTIVE_L
```

前置 Sensor 闪光灯开启有效电平配置

```
/*----- Camera Sensor Configuration End-----*/
```

以上宏定义分别定义了 power、reset、powerdown、flash 的 io 脚配置，这些配置信息在 rk29_camera.c 文件中各个控制函数（sensor_power_default_cb..）中都被用到，rk29_camera.c 文件定义各个默认控制函数都是基于 IO 的简单控制实现的，但是在某些项目中，可能这些函数并不能满足其控制要求，例如 power 函数，只是简单用 IO 来控制一个 LDO 的使能位，在 PMU 控制时，可能就涉及到其他控制了。所以在 board-rk29-xxx.c 定义了各个控制函数的用户自定义函数，只要将相关的配置宏打开，然后显示其控制函数即可，例如：

```
#define CONFIG_SENSOR_POWER_IOCTL_USR 0 /* 是否使用用户自定义控制函数 */
```

```
#if CONFIG_SENSOR_POWER_IOCTL_USR /* 该宏打开后，麻烦实现以下函数 */
```

```
static int sensor_power_usr_cb (struct rk29camera_gpio_res *res,int on)
```

```
{
```

```
    #error "CONFIG_SENSOR_POWER_IOCTL_USR is 1, sensor_power_usr_cb function must be written!!";
```

```
}
```

```
#endif
```

4.4.2 3.0 kernel board 文件配置

3.0kernel board 文件定义除了 32kernel 的定义外，增加以下定义：

```
#define CONFIG_SENSOR_O RK29_CAM_SENSOR_OV5642 /
#define CONFIG_SENSOR_IIC_ADDR_O 0x78
#define CONFIG_SENSOR_IIC_ADAPTER_ID_O 1
#define CONFIG_SENSOR_ORIENTATION_O 0
#define CONFIG_SENSOR_POWER_PIN_O INVALID_GPIO
#define CONFIG_SENSOR_RESET_PIN_O INVALID_GPIO
#define CONFIG_SENSOR_POWERDN_PIN_O RK29_PIN6_PB7
#define CONFIG_SENSOR_FLASH_PIN_O INVALID_GPIO
#define CONFIG_SENSOR_POWERACTIVE_LEVEL_O RK29_CAM_POWERACTIVE_L
#define CONFIG_SENSOR_RESETACTIVE_LEVEL_O RK29_CAM_RESETACTIVE_L
#define CONFIG_SENSOR_POWERDNACTIVE_LEVEL_O RK29_CAM_POWERDNACTIVE_H
#define CONFIG_SENSOR_FLASHACTIVE_LEVEL_O RK29_CAM_FLASHACTIVE_L
```

```
#define CONFIG_SENSOR_QCIF_FPS_FIXED_O 15
#define CONFIG_SENSOR_QVGA_FPS_FIXED_O 15
#define CONFIG_SENSOR_CIF_FPS_FIXED_O 15
#define CONFIG_SENSOR_VGA_FPS_FIXED_O 15
#define CONFIG_SENSOR_480P_FPS_FIXED_O 15
#define CONFIG_SENSOR_SVGA_FPS_FIXED_O 15
#define CONFIG_SENSOR_720P_FPS_FIXED_O 30
```

```
#define CONFIG_SENSOR_1 RK29_CAM_SENSOR_OV2659 /* front camera sensor */
#define CONFIG_SENSOR_IIC_ADDR_1 0x60
#define CONFIG_SENSOR_IIC_ADAPTER_ID_1 1
#define CONFIG_SENSOR_ORIENTATION_1 0
#define CONFIG_SENSOR_POWER_PIN_1 INVALID_GPIO
#define CONFIG_SENSOR_RESET_PIN_1 INVALID_GPIO
#define CONFIG_SENSOR_POWERDN_PIN_1 RK29_PIN5_PD7
#define CONFIG_SENSOR_FLASH_PIN_1 INVALID_GPIO
#define CONFIG_SENSOR_POWERACTIVE_LEVEL_1 RK29_CAM_POWERACTIVE_L
#define CONFIG_SENSOR_RESEACTIVE_LEVEL_1 RK29_CAM_RESEACTIVE_L
#define CONFIG_SENSOR_POWERD娜ACTIVE_LEVEL_1 RK29_CAM_POWERD娜ACTIVE_H
#define CONFIG_SENSOR_FLASHACTIVE_LEVEL_1 RK29_CAM_FLASHACTIVE_L
```

```
#define CONFIG_SENSOR_QCIF_FPS_FIXED_1 15
#define CONFIG_SENSOR_QVGA_FPS_FIXED_1 15
#define CONFIG_SENSOR_CIF_FPS_FIXED_1 15
#define CONFIG_SENSOR_VGA_FPS_FIXED_1 15
#define CONFIG_SENSOR_480P_FPS_FIXED_1 15
#define CONFIG_SENSOR_SVGA_FPS_FIXED_1 15
#define CONFIG_SENSOR_720P_FPS_FIXED_1 30
```

Android 4.0 + 3.0 kernel 中 CONFIG_SENSOR_0 定义的为后置 sensor, CONFIG_SENSOR_1 定义的为前置 sensor。在 32kernel+Android2.3 中, 在只有定义一个 sensor 的情况下, 默认为后置, 不管其定义在 CONFIG_SENSOR_0 还是 CONFIG_SENSOR_1 上。

Android 4.0 + 3.0 kernel 中 CONFIG_SENSOR_ORIENTATION_X 定义的角度即为 android2.3+32kernel 中 android 中定义的角度, 角度定义规则与原来一致, 详见“android camera 摄像头模组方向说明”章节和“android4.0 预览垂直及水平镜像问题说明”章节

Android 4.0 + 3.0 kernel 中 CONFIG_SENSOR_XXX_FPS_FIXED_X 定义的是该 sensor 在指定分辨率下的帧率。Android 在 camera 启动会查询该类信息。Dv 录制时会将其与 media_profile.xml 中定义的同样分辨率下的帧率进行校验, 不相符的话不进行 dv 录制动作。所以该项帧率定义与 media_profile.Xml 中一定一致, 帧率的测试以及 media_profile.xml 的填写规则详见“DV 分辨率设置”章节说明;

4.4.2.1 3.0 kernel board 的文件配置 (针对多达 6 个 sensor)

在 Camera Driver v0.1.7 版本及其以上版本支持多达 6 个 sensor 的配置, 同一个硬件上一般不会有 2 个的 sensor, 但是如此配置可以达到一个目的: 不同硬件 (sensor 型号等不同) 之间共用同一个软件固件。配置宏如下:

```
#define CONFIG_SENSOR_01 RK29_CAM_SENSOR_OV5642 /* back camera sensor 1 */
#define CONFIG_SENSOR_IIC_ADDR_01 0x78
#define CONFIG_SENSOR_IIC_ADAPTER_ID_01 1
#define CONFIG_SENSOR_ORIENTATION_01 90
#define CONFIG_SENSOR_POWER_PIN_01 INVALID_GPIO
#define CONFIG_SENSOR_RESET_PIN_01 INVALID_GPIO
#define CONFIG_SENSOR_POWERDN_PIN_01 RK29_PIN6_PB7
#define CONFIG_SENSOR_FLASH_PIN_01 INVALID_GPIO
#define CONFIG_SENSOR_POWERACTIVE_LEVEL_01 RK29_CAM_POWERACTIVE_L
#define CONFIG_SENSOR_RESEACTIVE_LEVEL_01 RK29_CAM_RESEACTIVE_L
#define CONFIG_SENSOR_POWERD娜ACTIVE_LEVEL_01 RK29_CAM_POWERD娜ACTIVE_H
#define CONFIG_SENSOR_FLASHACTIVE_LEVEL_01 RK29_CAM_FLASHACTIVE_L

#define CONFIG_SENSOR_QCIF_FPS_FIXED_01 15000
#define CONFIG_SENSOR_240X160_FPS_FIXED_01 15000
#define CONFIG_SENSOR_QVGA_FPS_FIXED_01 15000
#define CONFIG_SENSOR_CIF_FPS_FIXED_01 15000
#define CONFIG_SENSOR_VGA_FPS_FIXED_01 15000
#define CONFIG_SENSOR_480P_FPS_FIXED_01 15000
#define CONFIG_SENSOR_SVGA_FPS_FIXED_01 15000
#define CONFIG_SENSOR_720P_FPS_FIXED_01 30000

#define CONFIG_SENSOR_02 RK29_CAM_SENSOR_OV5640 /* back camera sensor 2 */
#define CONFIG_SENSOR_IIC_ADDR_02 0x78
#define CONFIG_SENSOR_IIC_ADAPTER_ID_02 1
#define CONFIG_SENSOR_ORIENTATION_02 90
#define CONFIG_SENSOR_POWER_PIN_02 INVALID_GPIO
```

```

50: #define CONFIG_SENSOR_11 RK29_CAM_SENSOR_OV2659 /* front camera sensor 1 */
51: #define CONFIG_SENSOR_IIC_ADDR_11 0x60
52: #define CONFIG_SENSOR_IIC_ADAPTER_ID_11 1
53: #define CONFIG_SENSOR_ORIENTATION_11 270
54: #define CONFIG_SENSOR_POWER_PIN_11 INVALID_GPIO
55: #define CONFIG_SENSOR_RESET_PIN_11 INVALID_GPIO
56: #define CONFIG_SENSOR_POWERDN_PIN_11 RK29_PIN5_PD7
57: #define CONFIG_SENSOR_FLASH_PIN_11 INVALID_GPIO
58: #define CONFIG_SENSOR_POWERACTIVE_LEVEL_11 RK29_CAM_POWERACTIVE_L
59: #define CONFIG_SENSOR_RESEACTIVE_LEVEL_11 RK29_CAM_RESEACTIVE_L
60: #define CONFIG_SENSOR_POWERDRACTIVE_LEVEL_11 RK29_CAM_POWERDRACTIVE_H
61: #define CONFIG_SENSOR_FLASHACTIVE_LEVEL_11 RK29_CAM_FLASHACTIVE_L
62:
63: #define CONFIG_SENSOR_QCIF_FPS_FIXED_11 15000
64: #define CONFIG_SENSOR_240X160_FPS_FIXED_11 15000
65: #define CONFIG_SENSOR_QVGA_FPS_FIXED_11 15000
66: #define CONFIG_SENSOR_CIF_FPS_FIXED_11 15000
67: #define CONFIG_SENSOR_VGA_FPS_FIXED_11 15000
68: #define CONFIG_SENSOR_480P_FPS_FIXED_11 15000
69: #define CONFIG_SENSOR_SVGA_FPS_FIXED_11 15000
70: #define CONFIG_SENSOR_720P_FPS_FIXED_11 30000
71:
72:
73: #define CONFIG_SENSOR_12 RK29_CAM_SENSOR_OV2655 /* front camera sensor 2 */
74: #define CONFIG_SENSOR_IIC_ADDR_12 0x60
75: #define CONFIG_SENSOR_IIC_ADAPTER_ID_12 1
76: #define CONFIG_SENSOR_ORIENTATION_12 270
77: #define CONFIG_SENSOR_POWER_PIN_12 INVALID_GPIO
78: #define CONFIG_SENSOR_RESET_PIN_12 INVALID_GPIO

```

即后置 sensor 配置在 CONFIG_SENSOR_0X 的宏定义上 (X 代表数字 1、2)，前置 sensor 配置在 CONFIG_SENSOR_1X 的宏定义上。

如此配置完之后，kernel 已经可以支持多达 6 个 sensor 硬件的注册。针对 6 个不同 sensor，Android 部分的 media_profiles.xml 文件的配置规则如下：

- 1) 如果硬件的 sensor 配置在 CONFIG_SENSOR_0X/1Y 上，那么这 2 个 sensor 的 DV 配置信息必须配置在 media_profilesXY.xml 上；
- 2) 如果对应的 media_profilesXY.xml 文件不存在，dv 配置信息默认采用 media_profiles.xml 文件的配置信息；

项目工程配置成多余 2 个 sensor 之后，sensor 驱动有某些地方不兼容，必需进行修改，修改地方如下：

```

02821: SENSOR_DG("\n%s\n",SENSOR_NAME_STRING(),__FUNCTION__,__cmd);
02822: switch (cmd)
02823: {
02824:     case RK29_CAM_SUBDEV_DEACTIVATE:
02825:     {
02826:         sensor_deactivate(client);
02827:         break;
02828:     }
02829:
02830:     case RK29_CAM_SUBDEV_IOREQUEST:
02831:     {
02832:         sensor->sensor_io_request = (struct rk29camera_platform_data*)arg;
02833:         if (sensor->sensor_io_request != NULL) {
02834:             sensor->sensor_gpio_res = NULL;
02835:             for (i=0; i<RK29_CAM_SUPPORT_NUMS;i++) {
02836:                 if (sensor->sensor_io_request->gpio_res[i].dev_name &&
02837:                     (strcmp(sensor->sensor_io_request->gpio_res[i].dev_name, dev_name(icd->pdev)) == 0)) {
02838:                     sensor->sensor_gpio_res = (struct rk29camera_gpio_res*)&sensor->sensor_io_request->gpio_res[i];
02839:                 }
02840:             }
02841:             if (sensor->sensor_gpio_res == NULL) {
02842:                 SENSOR_TR("\n%s\n", "obtain gpio resource failed when RK29_CAM SUBDEV IOREQUEST \n",SENSOR_NAME_STRING(),__FUNCTION__);
02843:                 ret = -EINVAL;
02844:                 goto sensor_ioctl_end;
02845:             }
02846:         } else {
02847:             SENSOR_TR("\n%s\n", "RK29_CAM SUBDEV IOREQUEST fail\n",SENSOR_NAME_STRING(),__FUNCTION__);
02848:             ret = -EINVAL;
02849:             goto sensor_ioctl_end;
02850:         }
02851:     }
02852:     /* ddl@rock-chips.com : if gpio_flash havn't been set in board-xxx.c, sensor driver must notify is not support for this project */
02853:     #if CONFIG_SENSOR_FLASH

```

参考 ov2659.c 驱动的修改。

4.4.2.2 RK30 Kernel Board 文件配置以及 sensor 驱动移植说明

30 中有两个 CIF 控制器，sensor 可以挂接在任意一个控制器上。board 文件中增加 `CONFIG_SENSOR_CIF_INDEX_X` (X 为 sensor 编号) 宏，该值表示对应 sensor 挂接在对应值的 CIF 控制器上，值可以为 0 或者 1，具体定义需要根据具体硬件情况作修改。

```
00052:
00053: #ifdef CONFIG_VIDEO_RK29
00054: /*----- Camera Sensor Macro Define Begin -----*/
00055: /*----- Camera Sensor Configuration Macro Begin -----*/
00056: #define CONFIG_SENSOR_0_RK29_CAM_SENSOR_OV5642 /* back camera sensor */
00057: #define CONFIG_SENSOR_IIC_ADDR_0 0x78
00058: #define CONFIG_SENSOR_IIC_ADAPTER_ID_0 4
00059: #define CONFIG_SENSOR_CIF_INDEX_0 1
00060: #define CONFIG_SENSOR_ORIENTATION_0 90
00061: #define CONFIG_SENSOR_POWER_PIN_0 INVALID_GPIO
00062: #define CONFIG_SENSOR_RESET_PIN_0 INVALID_GPIO
00063: #define CONFIG_SENSOR_POWERDN_PIN_0 RK30_PIN1_PD6
00064: #define CONFIG_SENSOR_FLASH_PIN_0 INVALID_GPIO
00065: #define CONFIG_SENSOR_POWERACTIVE_LEVEL_0 RK29_CAM_POWERACTIVE_L
00066: #define CONFIG_SENSOR_RESEACTIVE_LEVEL_0 RK29_CAM_RESEACTIVE_L
00067: #define CONFIG_SENSOR_POWERDNACTIVE_LEVEL_0 RK29_CAM_POWERDNACTIVE_H
00068: #define CONFIG_SENSOR_FLASHACTIVE_LEVEL_0 RK29_CAM_FLASHACTIVE_L
00069:
```

menuconfig 中增加了“RKXX CIF work simultaneity (Two cif controller cann't work simultaneity) --->”的配置，默认为 off，如果需要两个 sensor 同时（注意是同时，不是都）工作，则需要开启该配置。

项目 sensor 驱动如果在 rk29 平台上可以正常使用，那么在 rk30 平台上，移植注意事项如下：

- 1)、`#include <mach/rk29_camera.h> ---> #include <plat/rk_camera.h>`
- 2)、rk30 平台 camera 驱动框架已经是支持 6 个 sensor 设备注册的，如果项目 sensor 驱动尚未打上 rk29 相关支持 6 个 sensor 设备注册的补丁 Camera_Patch_v1.0 以及 Camera_Patch_v1.1，参考 4.2.2.1 中关于 sensor 驱动头的修改（参考 ov2659）

注释：服务器中该提交之后，sensor 驱动都可以在 rk29、rk30 平台编译通过：

```
commit f4c030d7646445f6c9e72bf7526a9fab885212ed
Author: ddl <ddl@rock-chips.com>
Date: Thu Apr 12 11:23:26 2012 +0800
```

camera: all sensor driver(ov3640/ov9650 exception) can be compiled success in rk30 and rk29

- drivers/media/video/gc0307.c
- drivers/media/video/gc0308.c
- drivers/media/video/gc0309.c
- drivers/media/video/gc0309_for_td8801.c
- drivers/media/video/gc2015.c
- drivers/media/video/gt2005.c
- drivers/media/video/hi253.c
- drivers/media/video/hi704.c

drivers/media/video/mt9d112.c
 drivers/media/video/mt9d113.c
 drivers/media/video/mt9m112.c
 drivers/media/video/mt9p111.c
 drivers/media/video/mt9t111.c
 drivers/media/video/nt99250.c
 drivers/media/video/ov2640_rk.c
 drivers/media/video/ov5640.c
 drivers/media/video/ov5640_for_td8801.c
 drivers/media/video/ov7675.c
 drivers/media/video/ov7690.c
 drivers/media/video/s5k6aa.c
 drivers/media/video/sid130B.c
 drivers/media/video/siv120b.c

3)、 board 文件 Sensor 电源控制说明

2.6.32 内核 board 文件已经增加项目独立的 power/powerdown/flash/reset 的控制函数，目前 rk30sdk 实现了 power 函数，power 函数实现后，camera 在退出时候会将 sensor 模组的电源关闭，这就必须确认该 sensor 模组的 I2C 由独占（即该 i2c 通道硬件上未连接其他 i2c 设备），board-rk30-sdk.c 相关代码如下：

```

0211: #define CONFIG_SENSOR_POWER_IOCTL_USR 1 //define this refer to your board layout
0212: #define CONFIG_SENSOR_RESET_IOCTL_USR 0
0213: #define CONFIG_SENSOR_POWERDOWN_IOCTL_USR 0
0214: #define CONFIG_SENSOR_FLASH_IOCTL_USR 0
0215:
0216: static void rk_cif_power(int on)
0217: {
0218:     struct regulator *ldo_18,*ldo_28;
0219:     ldo_28 = regulator_get(NULL, "ldo7"); // vcc28_cif
0220:     ldo_18 = regulator_get(NULL, "ldo1"); // vcc18_cif
0221:
0222:     if(on == 0){
0223:         regulator_disable(ldo_28);
0224:         regulator_put(ldo_28);
0225:         regulator_disable(ldo_18);
0226:         regulator_put(ldo_18);
0227:         mdelay(500);
0228:     }
0229:     else{
0230:         regulator_set_voltage(ldo_28, 2800000, 2800000);
0231:         regulator_enable(ldo_28);
0232:         // printk("%s set ldo7 vcc28_cif=%dmV end\n", __func__, regulator_get_voltage(ldo_28));
0233:         regulator_put(ldo_28);
0234:
0235:         regulator_set_voltage(ldo_18, 1800000, 1800000);
0236:         // regulator_set_suspend_voltage(ldo, 1800000);
0237:         regulator_enable(ldo_18);
0238:         // printk("%s set ldo1 vcc18_cif=%dmV end\n", __func__, regulator_get_voltage(ldo));
0239:         regulator_put(ldo_18);
0240:     }
0241: }
0242:
0243: #if CONFIG_SENSOR_POWER_IOCTL_USR
0244: static int sensor_power_usr_cb (struct rk29camera_gpio_res *res,int on)
0245: {
0246:     // #error "CONFIG_SENSOR_POWER_IOCTL_USR is 1, sensor_power_usr_cb function must be writed!";
0247:     rk_cif_power(on);
0248: }
0249: #endif
    
```

4)、自定义 sensor 序列

RK30 Camera driver v0.2.b 及其以后版本，

为了保证 sdk 的 sensor 驱动的统一，sensor 驱动序列一般不做修改。但是，有时如 flip 和 mirror 等可能因为项目不同而需要相应做出更改。因此为了适应项目需求，将 sensor 序列做成可自定义方式。

需要自定义 sensor 序列时，请按照以下步骤操作：

(1)、在 `drivers/media/video/` 目录下增加一个文件，命名为 `xx_user_series.c` (`xx` 为对应的 sensor 驱动文件名，如 sensor 驱动文件名为 `ov2659.c`，则 `xx` 为 `ov2659`)。

(2)、将相应 sensor 驱动中

`#ifdef CONFIG_XX_USER_DEFINED_SERIES` 宏定义的 **#else 开始到#end** 的代码(即 sensor 相关序列数组，如 `sensor_init_data[]` 等) 复制到 `xx_user_series.c` 文件中，再在该文件对相应的序列做出修改；

(3)、编译时需要配置如下 `menuconfig`，以 `ov2659` 为例：

```
Kernel : menuconfig:
Device Drivers --->
Multimedia support --->
Video capture adapters --->
ov2659 camera support for rockchip
[*]OV2659 user defined init series
```

4.5 各公司 sensor 不同点

OV 公司的 sensor 在进行 `preview`、`capture` 时都必须修改 sensor 的相关分辨率，推荐 `capture` 的流程可以参考 `ov5642`，简单实现可以参考 `ov2659` 等；

Aptina (原 Micron) 公司 sensor 在初始化时就将 `preview`、`capture` 分辨率设置在不同的寄存器(通过页地址切换)中，在进行 `capture` 与 `preview` 之间切换时只要进行相关地址的切换，参考代码：`mt9p111.c`、`mt9m112.c` 等

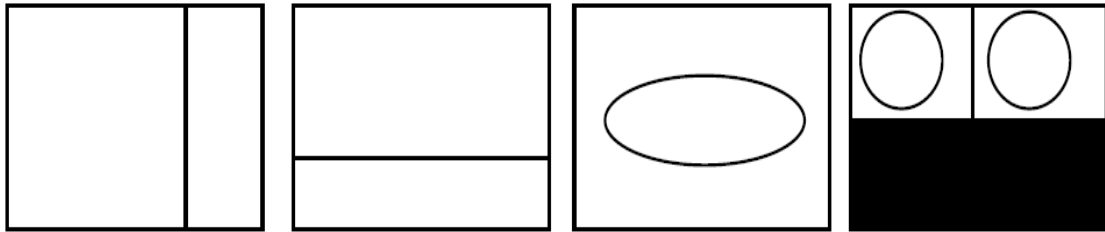
Samsang 公司的 sensor 在设置上与 OV 公司的较为类似，但是在休眠唤醒中需要进行特殊处理。参考代码：`s5k6aa.c`

5 sensor 调试注意点：

- 1、 确认 sensor 的各电源引脚是否供电正常？ **其中包括 sensor 的 IO 电源与 VIP 控制器的 IO 电源是否相符？**
- 2、 确认 sensor 复位电平是否正常？
- 3、 确认 sensor power down 控制 IO 是否控制正常，有些板子 sensor Ido 的控制脚也单独引出，该脚也需要确认是否控制正常？
- 4、 确认 VIP 输出的 `mclk` 是否满足 sensor 要求？
- 5、 最后确认 I2C 是否正常？如果在 I2C 输出波形正常的情况下，sensor 在 i2c 的第 9 位未产生 ACK 信号，那么在确认以上 4 点后，需要考虑其他器件的影响，在 25 内核版本的 `tca6424` 驱动中，为了满足其清中断要求，会在 i2c 操作之后紧跟着操作 i2c 的 `sck` 和 `sda` 线。这样操作会影响到某些 I2C 器件的读写!!!
- 6、 图像异常时需要判断如下情况：
 - 1)、HREF/VSYNC 的有效电平是否与 VIP 设置一致；
 - 2)、PCLK 输出有效数据的边沿是否与 VIP 设置一致；
 - 3)、PCLK 最高只能 96MHz；

- 4)、数据输出格式是否满足要求: YUV422, 输出顺序: UYVY、YUYV
- 5)、sensor IO 的电平是否与 VIP IO 的电平不一致 (1.8v/2.8v);
- 7、Dual-Sensor 共用 VIP 控制器的 DATA/VSYNC/HREF/PCLK 信号线时, 确认各个 Sensor 的 Stanby 动作是否符合要求, 同一时刻只能有一个 sensor 处于工作状态。同时由于共用 IO 的负载增大, 所以必须关注各个 sensor 的 IO 驱动能力是否需要调整;

• 图像窗口位置

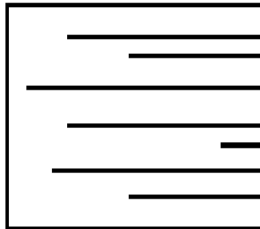


行同步问题

场同步问题

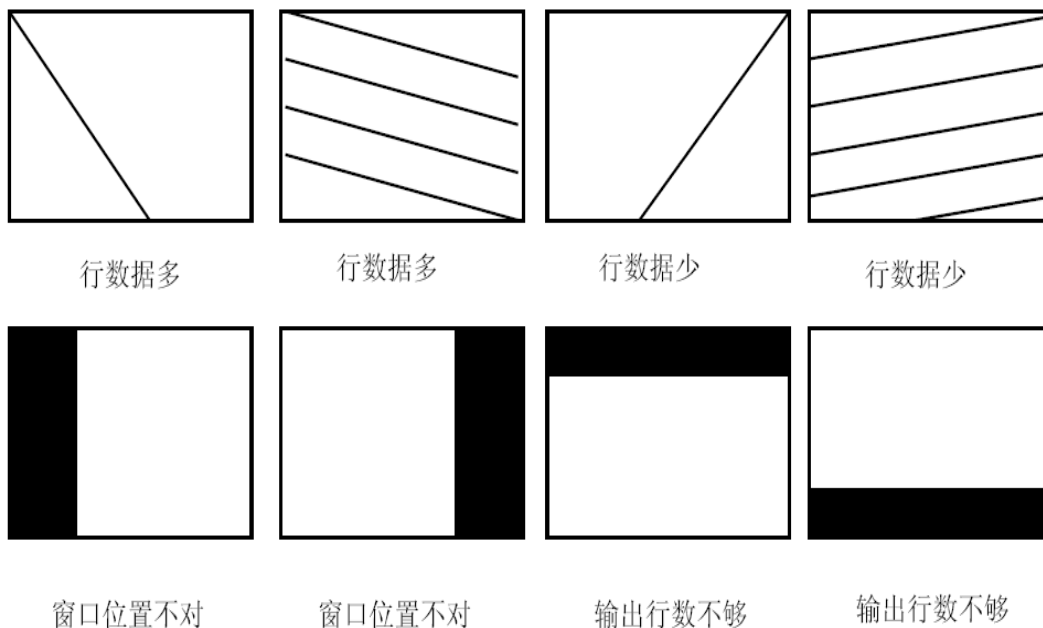
后端采样频率太高

后端采样频率太低



时钟与数据同步有问题

• 图像窗口位置



6 当前 RK29SDK/RK30SDK 支持的 SOC_CAMERA 型号

Camera Sensor 型号	Driver 状态
ov7675	纽曼 客户使用
ov2655	TCL 客户、SDK 使用
ov2659	SDK 使用
Ov2640	希科普
ov3640 (Auto focus AD5820/Fixed focus)	佳音客户 (手机) 使用
ov3660	海信
ov5642 (Auto focus A3907 /Fixed focus)	SDK 使用
Ov5640 (Auto focus)	福建实达
s5k6aa	万利达、FIH 客户使用
S5k5ca	华为 (手机组语音 pad, 还在调试中)
mt9d112	
mt9d113	TCL 客户
mt9t111	上海依天
Mt9p111	Fih
gc0307	
gc0308	力瑞客户
gc0329	华为 (手机组语音 pad, 还在调试中)
gt2005	
gc0309	天智伟业 客户使用
gc2015	天智伟业 客户使用
siv120b	Foxconn 客户使用

sid130B	
hi253	天智伟业 客户使用
hi704	天智伟业 客户使用
nt99250	易方 客户使用
Sp0838	
Sp0215	

以上列表所列出的 Sensor 为 sdk 中已经提供的 Sensor 驱动，如果客户需要选择其他模组的也可以，但是必须注意采用的 Sensor 模组必须符合以下条件：

- 1)、只支持集成 ISP 功能 Sensor 芯片，不支持直接输出 RAW 数据的 Sensor 芯片，目前 RK 芯片并未实现 ISP 功能，所以不能支持；
- 2)、只支持输出 YUV422 数据的 Sensor 芯片；
- 3)、Sensor 输出电平必须是 1.8v、3.3v 之一；
- 4)、Sensor AF 功能必须模组集成马达，集模组内置支持，RK 芯片未实现 ISP 功能无法通过控制马达实现 AF 的功能；















7. USB 摄像头支持说明

Usb 摄像头采用 linux 的 uvc 标准驱动，linux 编译时必须打开 uvc 驱动相关选项：
Device Drivers --->









```
<*> Multimedia support --->
[*] Video capture adapters --->
[*] V4L USB devices --->
<*> USB Video Class (UVC)
```

目前 android 支持自动识别 uvc 驱动以及 rk29camera 驱动，针对 usb 摄像头目前只支持 yuv422 (yu1v) 数据，由于 usb 摄像头在高分辨率情况下采用的是 mjpeg 数据格式，目前暂不支持；

以下为 uvc 驱动支持摄像头列表，选用时最好采用下表中，并且支持输出 yuv422(yu1v) 数据的摄像头：**(注释：以下列表中的 uvc sensor 不一定支持，sdk 板调试确认支持的只用罗技 C110)**

0402:5606	USB 2.0 Camera (VIT D2010 notebooks)	ALi Corporation	 [12]
0408:030c	HP Webcam (HP Pavilion DV6744 and DV6750)	Quanta Computer	
041e:4057	Creative Live! Cam Optia	Creative Labs	
041e:4058	Creative Live! Cam Optia AF	Creative Labs	
041e:4063	Creative Live! Cam Video IM Pro	Creative Labs	 [7]
041e:4065	Creative Live! Cam Optia Pro	Creative Labs	
041e:406a	Creative Live! Cam Notebook Ultra	Creative Labs	
041e:406b	Creative Live! Cam Chat IM	Creative Labs	
041e:406c	Creative Live! Cam Sync	Creative Labs	
041e:4080	Creative Live! Cam Socialize HD	Creative Labs	
041e:4088	Creative Live! Cam Chat HD	Creative Labs	
0458:505e	Genius iSlim 330	Genius	
0458:7055	Genius iSlim 2020AF	Genius	
0458:705d	Genius iSlim 2000AF	Genius	

0458:706e	Genius eFace 2025	Genius	✓
0458:7070	Genius FaceCam 310	Genius	⚠ [1,3]
0458:7071	Genius iSlim 1300 V2	Genius	✓
0458:707c	Genius eFace 1300	Genius	✓
045e:00f8	Microsoft LifeCam NX-6000	Microsoft	✓
045e:074a	Microsoft LifeCam VX-500	Microsoft	✓
045e:075d	Microsoft LifeCam Cinema	Microsoft	⚠ [1,3]
045e:0770	Microsoft LifeCam VX-700	Microsoft	✓
045e:0772	Microsoft LifeCam Studio	Microsoft	✓
046d:0802	Logitech Webcam C200	Logitech	✓
046d:0804	Logitech Webcam C250	Logitech	✓
046d:0805	Logitech Webcam C300	Logitech	✓
046d:0807	Logitech Webcam C500	Logitech	✓
046d:0808	Logitech Webcam C600	Logitech	✓
046d:0809	Logitech Webcam Pro 9000	Logitech	✓
046d:080a	Logitech Portable Webcam C905	Logitech	✓
046d:0819	Logitech Webcam C210	Logitech	✓
046d:081d	Logitech Webcam C310	Logitech	✓
046d:0821	Logitech Portable Webcam C910	Logitech	✓
046d:0825	Logitech HD Webcam C270	Logitech	✓
046d:08c1	Logitech Quickcam Fusion	Logitech	⚠ [1,2]
046d:08c2	Logitech Quickcam Orbit/Sphere MP	Logitech	⚠ [1,2]
046d:08c3	Logitech Quickcam for Notebooks Pro	Logitech	⚠ [1,2]
046d:08c5	Logitech Quickcam Pro 5000	Logitech	⚠ [1,2]
046d:08c6	Logitech Quickcam OEM Dell Notebook	Logitech	⚠ [1,2]
046d:08c7	Logitech Quickcam OEM Cisco VT Camera II	Logitech	⚠ [1,2]
046d:08c9	Logitech Quickcam Ultra Vision	Logitech	⚠ [1,2]
046d:08ca	Logitech Quickcam Fusion (2006 model)	Logitech	⚠ [1,2]
046d:08cb	Logitech Quickcam for Notebooks Pro (2006 model)	Logitech	⚠ [1,2]
046d:08cc	Logitech Quickcam Orbit/Sphere MP (2006 model)	Logitech	⚠ [1,2]
046d:08ce	Logitech Quickcam Pro 5000 (2006 model)	Logitech	⚠ [1,2]
046d:0990	Logitech Quickcam Pro 9000 for Business	Logitech	⚠ [6]
046d:0991	Logitech Quickcam Pro for Notebooks (2007 model)	Logitech	✓
046d:0992	Logitech Quickcam Communicate Deluxe	Logitech	✓
046d:0994	Logitech Quickcam Orbit/Sphere AF	Logitech	✓
046d:09a1	Logitech Quickcam Communicate MP/S5500	Logitech	✓
046d:09a2	Logitech Quickcam Communicate Deluxe/S7500	Logitech	✓
046d:09a4	Logitech Quickcam E 3500	Logitech	✓
046d:09a5	Logitech Quickcam 3000 for Business	Logitech	✓
046d:09a6	Logitech Quickcam Vision Pro	Logitech	✓
046d:09b0	Acer OrbiCam (Acer notebooks)	Logitech	✓
046d:09b2	Fujitsu Webcam (Fujitsu-Siemens notebooks)	Logitech	⚠ [3]
046d:09c0	Quickcam for Dell Notebooks (Dell notebooks)	Logitech	⚠ [1,2]

046d:09c1	Logitech Quickcam Deluxe for Notebooks Logitech Quickcam Deluxe for Notebooks for Business	for Logitech	 [1,2]
0471:0331	Philips SPC 1300NC	Philips	
0471:0332	Philips SPC 1000NC	Philips	
0471:0333	Philips SPC 620NC	Philips	
0471:0334	Philips SPC 520/525NC	Philips	
0471:2034	Philips SPC 530NC	Philips	
0471:2037	Philips SPC 1330NC	Philips	
0471:2038	Philips SPC 2050NC	Philips	
0474:02da	Sanyo Xacti HD2000	Sanyo Electric	
0474:0722	Sanyo W33SA	Sanyo Electric	
0474:0b0e	Sanyo VPC-CA102	Sanyo Electric	
0489:d00a	Traveler DC 8900	Schenker Inc.	
04cb:014c	Fujifilm FinePix A340	Fujifilm	
04cb:016f	Fujifilm FinePix S5500 Zoom	Fujifilm	
04cb:0172	Fujifilm FinePix E550	Fujifilm	
04da:2318	Panasonic NV-GS11/230/250 (webcam mode)	Camcorder Panasonic	
04da:231a	Panasonic NV-GS11/230/250 (DV mode)	Camcorder Panasonic	
04da:231d	Panasonic NV-GS27/37/320/500 (webcam mode)	Camcorder Panasonic	
04da:231e	Panasonic NV-GS27/37/320/500 (DV mode)	Camcorder Panasonic	
04f2:a133	Chicony USB 2.0 1.3MP UVC Camera (Maxell MaxCam MWC-1300D)	Chicony Electronics	
04f2:a13c	HP KQ246AA 8.0MP Deluxe Webcam	Chicony Electronics	
04f2:a13e	Panda 10C	Chicony Electronics	
04f2:a147	Medion P86004 2MP Webcam with Headset	Chicony Electronics	
04f2:b008	Chicony USB 2.0 Camera	Chicony Electronics	
04f2:b012	Chicony 1.3M UVC Webcam (Asus G1S notebooks)	Chicony Electronics	 [3]
04f2:b013	Chicony USB 2.0 Camera (Lenovo 3000 N200 notebooks)	Chicony Electronics	
04f2:b015	Chicony VGA 24fps UVC Webcam (HP notebooks)	Chicony Electronics	
04f2:b016	Chicony VGA 30fps UVC Webcam (HP notebooks)	Chicony Electronics	
04f2:b018	Chicony 2M UVC Webcam (Compaq notebooks)	Chicony Electronics	
04f2:b021	ViewSonic 1.3M, USB2.0 Webcam (ViewSonic VX2255WMB screens)	Chicony Electronics	 [5]
04f2:b022	Gateway USB 2.0 Webcam (One C34xx notebooks)	Chicony Electronics	
04f2:b023	Gateway USB 2.0 Webcam (HP Pavilion DV9560EG notebooks)	Chicony Electronics	
04f2:b024	USB 2.0 Webcam (Packard Bell notebooks)	Chicony Electronics	
04f2:b027	Gateway USB 2.0 Webcam (Gateway T-1616 notebooks)	Chicony Electronics	
04f2:b029	USB 2.0 1.3M UVC WebCam (Asus F6S notebooks)	Chicony Electronics	
04f2:b033	USB 2.0 1.3M UVC WebCam (Asus M70VM notebooks)	Chicony Electronics	
04f2:b044	Acer CrystalEye webcam (Acer Aspire 5535 notebooks)	Chicony Electronics	
04f2:b062	CNF7045 (Packard-Bell notebooks)	Chicony Electronics	
04f2:b070	Toshiba Satellite L350D notebooks	Chicony Electronics	
04f2:b071	CNF7129 (Asus N10JA2 and EeePC 1000HE netbooks, K50IN notebooks)	Chicony Electronics	 [3,14]
04f2:b073	CNF7231 (MSI MS-1722 ID1)	Chicony Electronics	

04f2:b082	notebooks) CKA7227 (HP EliteBook 2530p)	Chicony Electronics	✓
04f2:b083	notebooks) CKF7063 (HP Compaq 6830s)	Chicony Electronics	✓
04f2:b084	Unnamed (Acer Aspire One D150)	Chicony Electronics	✓
04f2:b105	Lenovo EasyCamera (Lenovo IdeaPad Y530 notebooks)	Chicony Electronics	✓
04f2:b106	CNF7246 (Asus G71V notebooks)	Chicony Electronics	✓
04f2:b107	CNF7070 (HP 2133 notebooks)	Chicony Electronics	✓
04f2:b1b9	(Asus U52F notebooks)	Chicony Electronics	✓ [3]
04f2:b1bb	(Asus N82JV notebooks)	Chicony Electronics	✓ [3]
04f2:b1be	USB2.0 0.3M UVC WebCam (Asus UL30JT notebooks)	Chicony Electronics	✓ [3]
04f2:b1e5	USB2.0 0.3M UVC WebCam (Asus K25JC and K52De notebooks)	Chicony Electronics	✓ [3]
058f:3820	Future Boy PC USB Webcam (Alcor Micro AU3820 chipset)	Alcor Micro	✓
05a9:2640	OmniVision OV2640 (Dell Inspiron 1420/1720 notebooks)	OmniVision	✓
05a9:2643	OmniVision Monitor Webcam (Dell SP2208WFP)	OmniVision	✓
05a9:2649	OmniVision Monitor Webcam (Dell SP2309W)	OmniVision	✓
05a9:7670	OmniVision OV7670 (Dell XPS m1330 notebooks)	OmniVision	✓
05ac:8502	Apple built-in iSight	Apple	✓ [4]
05c8:0103	FO13FF-65 PC-CAM	Foxlink	✓
05ca:18a1	Integrated Webcam (Dell Studio 1535 notebooks)	Ricoh	✓
05ca:18b7	Sony Visual Communication Camera (Sony VPCS12J1E notebooks)	Ricoh	✓
064e:8100	Integrated Webcam 2M (Dell Vostro 1088 notebooks)	SuYin	✓
064e:a100	Acer OrbiCam (Acer notebooks)	SuYin	✓
064e:a101	Acer CrystalEye webcam (Acer notebooks)	SuYin	✓
064e:a102	Webcam (Acer Timeline 1810T notebooks)	SuYin	✓
064e:a103	Acer OrbiCam (Acer Aspire 7730ZG-343G32Mn notebooks)	SuYin	✓
064e:a110	HP Webcam (HP TX2000 notebooks)	SuYin	✓
064e:a111	USB 2.0 Camera (Datron TW7A notebooks)	SuYin	✓
064e:a116	USB 2.0 UVC 1.3M WebCam (Asus N20A notebooks)	SuYin	⚠ [3]
064e:a117	Acer HD Crystal Eye webcam (Acer 4930 notebooks)	SuYin	✓
064e:a118	Integrated Webcam (Dell Mini 9 notebooks)	SuYin	✓
064e:a136	USB 2.0 UVC 0.3M Webcam (Asus UL50VT notebooks)	SuYin	✓
064e:a219	UVC 1.3M Webcam (Acer Aspire 5745G notebooks)	SuYin	✓
064e:d101	Acer Crystal Eye webcam (Acer Aspire One AOA150-Ab notebooks)	SuYin	✓
064e:e201	Integrated Webcam (Lenovo Thinkpad Edge 13" notebooks)	SuYin	✓
06f8:3005	Hercules Dualpix Exchange	Guillemot Corp.	✓
06f8:3007	Hercules Dualpix Chat and Show	Guillemot Corp.	✓
06f8:300a	Hercules Dualpix Infinite	Guillemot Corp.	✓
06f8:300c	Hercules Classic Silver	Guillemot Corp.	✓
06f8:3020	Hercules Webcam EC300 (Malata PC-81005 netbooks and clones)	Guillemot Corp.	✓

090c:37b3		Lenovo EasyCamera (Lenovo G560 notebooks)	Silicon Motion	✓
090c:b370	Silicon Motion SM370		Silicon Motion	✓
090c:b371	Silicon Motion SM371		Silicon Motion	✓
093a:2700	iSonic W002 A4Tech PK-635K Digital Innovations 1.3MP Webcam	Pixart Imaging		✓
093a:2800	DealExtreme USB 2.0 Camera	Pixart Imaging		✓
093a:2900	Agama V-315	Pixart Imaging		✓
0ac8:0336	Elecom UCAM-DLQ30 (Vimicro VC0336 chipset)	Solid Years		✓
0ac8:3313	TopSpeed USB 2.0 Camera B.	Vimicro		✓
0ac8:332d	Vega USB 2.0 Camera (AOC screens and Techsolo TCA-4900)	Vimicro		✓
0ac8:3330	Sirius USB 2.0 Camera (Xinyi Y867 LCD Prince)	Vimicro		✓
0ac8:3343	Sirius USB 2.0 Camera (Fujitsu A6110 notebook)	Vimicro		✓
0ac8:3410	Venus USB 2.0 Camera (Minoru3D)	Vimicro		✓ [8]
0ac8:3420	Venus USB 2.0 Camera (Tevion MD 85872 and Minoru3D)	Vimicro		✓ [8]
0ac8:3450	A4Tech PK-333E	A4Tech		✓
0ac8:3460	Kodak Dual Webcamera	Sakar Corp.		✓
0ac8:c302	Vega USB 2.0 Camera (Samsung Q45 notebook)	Vimicro		✓
0ac8:c303	Saturn USB 2.0 Camera (Samsung screens)	Vimicro		✓
0ac8:c315	HP Elite Autofocus Webcam	Vimicro		✓
0ac8:c338	Namuge 2MP Webcam	Namuga		✓
0c45:62c0	Sonix USB 2.0 Camera (Acer Aspire 5050 and HP Pavilion DV6000 notebooks) Trust SpotLight Webcam Pro Centrios 1.3MP auto focus	Sonix Technology		✓
0c45:62e0	MSI Starcam Racer Rosewill RCM-8163	Sonix Technology		✓
0c45:62f1	Avatec CMA-L688 HueHD	Sonix Technology HueHD		⚠ [11]
0c45:6310	USB 2.0 Camera (Trust Chat Webcam)	Sonix Technology		✓
0c45:63e0	Sonix Integrated Webcam (Dell notebooks)	Sonix Technology		✓
0c45:63ea	Laptop Integrated Webcam 2M (Dell Studio 1555 notebooks)	Sonix Technology		✓
0c45:6409	USB 2.0 Camera (Nokia Booklet 3G netbooks)	Sonix Technology		✓
0c45:6415	Laptop Integrated Webcam 1.3M (Dell Inspiron 13z notebooks)	Sonix Technology		✓
0e8d:0004	MediaTek MT6227 phone	MediaTek Inc		✓
13d3:509b	USB 2.0 Camera (Asus EeePC T91 netbooks)	Genesys Logic Technology		✓
13d3:5103	USB 2.0 Camera (Medion Akoya All-in-one PC)	Sonix Technology		✓
13d3:5122	USB 2.0 Camera (Asus U33JC netbooks)	Sonix Technology		⚠ [3]
13d3:5130	USB 2.0 Camera (Asus K40AE, K50IE and K52JT notebooks)	Sonix Technology		⚠ [3]
145f:013e	Trust Megapixel USB2 WB-5600R	Trust		✓
145f:013f	Trust Megapixel USB2 Auto Focus Webcam	Trust		✓

145f:0142	Trust WB-6250X Webcam	Trust	✓
145f:015b	Trust WB-8500X Webcam	Trust	✓
152d:0310	JMicron USB2.0 XGA WebCam	JMicron	✓
174f:1118	Syntek D-Max HP Webcam (HP Pavillon DV3 notebooks)	Syntek	✓
174f:5212	Syntek USB 2.0 UVC PC Camera (HP Spartan notebooks)	Syntek	✓
174f:5215	Syntek USB 2.0 UVC PC Camera (upcoming UMPC device)	Syntek	✓
174f:5271	Syntek USB 2.0 UVC PC Camera (upcoming UMPC device)	Syntek	✓
174f:5931	Syntek USB 2.0 UVC PC Camera (Samsung Q310 notebooks)	Syntek	✓
174f:5a11	Unknown (Asus A8Sc notebook)	Unknown	✓
174f:5a31	Sonix USB 2.0 Camera (Asus M50SV notebook)	Sonix	✓
174f:5a35	Sonix USB 2.0 Camera (Asus F3KE and G2S notebook)	Sonix	⚠ [3]
174f:8a12	Syntek USB 2.0 UVC PC Camera (Packard Bell Easynote MX52 notebooks)	Syntek	✓
174f:8a33	Syntek USB 2.0 UVC PC Camera (Asus U3S notebooks)	Syntek	✓
174f:8a34	Syntek USB 2.0 UVC PC Camera (JAOtech Smart Terminal)	Syntek	✓
177f:0060	Sweex	WC060 Series HD Webcam	✓
1778:0204	PEVO corp	IPEVO Point 2 View	✓
17dc:0202	Miricle 307K	Thermoteknix	✓
17ef:1004	Integrated Camera (Lenovo Thinkpad T61 notebooks)	Lenovo	✓
17ef:480b	Integrated Camera (Lenovo SL400 and SL500 notebooks)	Lenovo	✓
17ef:481c	Integrated Camera (Lenovo SL510 notebooks)	Lenovo	✓
1871:01f0	Aveo Technology USB 2.0 Camera	Aveo Technology	✓
1871:0306	Aveo Technology USB 2.0 Camera	Aveo Technology	✓
18cd:cafe	Pico iMage	Ecamm	✓
18ec:3188	Manta MM-353 Plako	ArkMicro	✓
18ec:3288	FSC WebCam V30S	ArkMicro	⚠ [9]
18ec:3290	USB 2.0 PC Camera (Sabrent WCM-6LNV)	ArkMicro	✓
18ec:3299	USB 2.0 PC Camera (model number QC3231)	ArkMicro	✓
199e:8101	DFx 21BU04	ImagingSource	✓
19ab:1000	Bodelin ProScope HR	Bodelin	✓
19ab:1020	Bodelin ProScope HR2	Bodelin	✓
19ff:0102	Dynex 1.3MP Webcam	Dynex	✓
1b3b:2951	MSI StarCam 370i	Unknown	✓
1c4f:3000	SiGma Micro USB Web Camera	SiGma Micro	✓
1cac:a332	Kinstone C8 webcam (Vimicro chipset)	Kinstone	✓
1cac:b288	Kinstone C18 webcam (Sonix chipset)	Kinstone	✓
1e4e:0100	USB 2.0 Camera	Etron Technologies	⚠ [10]
22b8:6006	Motorola MOTOROKR E6	Motorola	✓
5986:0100	Acer OrbiCam (Acer notebooks)	Bison Electronics	✓
5986:0101	USB2.0 Camera (Packard Bell Easynote SJ notebooks)	Bison Electronics	✓
5986:0102	Acer Crystal Eye webcam (Acer)	Bison Electronics	✓

	TravelMate 7720 notebooks)		
5986:0200	Acer OrbiCam (Acer notebooks)	Bison Electronics	✓
5986:0202	Bison (Fujitsu-Siemens Amilo S12636 notebooks)	Bison Electronics	✓
5986:0203	Bison (Advent 4211 and MSI Wind notebooks)	Bison Electronics	✓
5986:0205	Lenovo EasyCamera (Lenovo N500 and U330 notebooks)	Bison Electronics	⚠[9]
5986:0241	Bison (MSI Wind Top AE1900 nettop)	Bison Electronics	⚠[15]
5986:0314	BisonCam, NB Pro (MSI Wind U135DX netbook)	Bison Electronics	✓
eb1a:2571	eMPIA 27xx based camera (unbranded)	eMPIA Technology	✓
eb1a:2761	eMPIA 2761 based camera (unbranded)	eMPIA Technology	✓
eb1a:2771	eMPIA 2771 based camera (Intelbras iPlug netbook)	eMPIA Technology	✓

8 android camera 模块配置注意点

8.1 DV 分辨率设置

8.1.1 android 2.3 media_profile.xml

由于各个项目的前后 sensor 都采用不一，所以针对 DV 的分辨率设置需要客户根据自身样机的实际 sensor 来进行设置。现在针对修改说明如下（**注释部分**）：

文件目录：

```
out\target\product\rk29sdk\system\etc\media_profiles.xml
device\rockchip\rk29sdk\media_profiles.xml
```

```
<CamcorderProfiles cameraId="0"> // 后置 sensor ， 只有一个 sensor ， 默认为后置
```

```
<EncoderProfile quality="high" fileFormat="mp4" duration="60">
```

```
<Video codec="h264"
```

```
bitRate="3000000"
```

```
width="1280" // 高质量录像对应的分辨率
```

```
height="720"
```

```
frameRate="15" />
```

```
<Audio codec="amrnb"
```

```
bitRate="12200"
```

```
sampleRate="8000"
```

```
channels="1" />
```

```
</EncoderProfile>
```

```
<EncoderProfile quality="low" fileFormat="mp4" duration="30">
  <Video codec="h264"
    bitRate="256000"
    width="800" // 低质量录像对应的分辨率
    height="600"
    frameRate="15" />

  <Audio codec="amrnb"
    bitRate="12200"
    sampleRate="8000"
    channels="1" />

</EncoderProfile>

<ImageEncoding quality="90" />
<ImageEncoding quality="80" />
<ImageEncoding quality="70" />
<ImageDecoding memCap="20000000" />

<Camera previewFrameRate="0" />

</CamcorderProfiles>

<CamcorderProfiles cameraId="1"> //前置 sensor

  <EncoderProfile quality="high" fileFormat="mp4" duration="60">
    <Video codec="h264"
      bitRate="3000000"
      width="1280" // 高质量录像对应的分辨率
      height="720"
      frameRate="15" />

    <Audio codec="amrnb"
      bitRate="12200"
      sampleRate="8000"
      channels="1" />
  </EncoderProfile>

  <EncoderProfile quality="low" fileFormat="mp4" duration="30">
    <Video codec="h264"
      bitRate="256000"
      width="176" // 低质量录像对应的分辨率
      height="144"
```

```
frameRate="15" />

<Audio codec="amrnb"
    bitrate="12200"
    sampleRate="8000"
    channels="1" />

</EncoderProfile>

<ImageEncoding quality="90" />
<ImageEncoding quality="80" />
<ImageEncoding quality="70" />
<ImageDecoding memCap="20000000" />

<Camera previewFrameRate="0" />

</CamcorderProfiles>
```

8.1.2 android 4.0 media_profile.xml

1)、Sensor 帧率测试

1.1 将 camera_test 可执行文件通过 adb 直接 push 到 /system/bin 目录下或者将 camera_test 放到 /out/target/product/rk29_phone_sdk/system/bin 目录下然后执行 ./mkimage.sh, 烧写下固件就可以使用 camera_test 了。

1.2 在串口中断输入 camera_test -h 会输出各项测试的提示。

帧率测试是 camera_test -i 默认是测试后置摄像头
可加选项:

-d /dev/video0 前置摄像头是 video1, 后置摄像头是 video0。

-r 800x600 只测试当前指定的分辨率

-m 1280x720 指定测试的最大分辨率

1.3 测试结果说明 (以下为 K97 机器的测试结果)

camera test.. version: 1.0.0

Current camera driver version: 0.1.6

Now detect 176x144 framerate, Please wait...

Time interval between frame: 12504

Now detect 320x240 framerate, Please wait...

Time interval between frame: 12504

Now detect 352x288 framerate, Please wait...

Time interval between frame: 12504

Now detect 640x480 framerate, Please wait...

Time interval between frame: 12504

Now detect 720x480 framerate, Please wait...

Time interval between frame: 11587

WARNING: This resolution is crop from 800x600 at (0x00,0x00) point, so window is cutted out!!!

Now detect 800x600 framerate, Please wait...

Time interval between frame: 12504

Now detect 1280x720 framerate, Please wait...

Time interval between frame: 6256

WARNING: This resolution is crop from 1280x1024 at (0x00,0x00) point, so window is cutted out!!!

“Time interval between frame: 12504”表示的帧率*1000，实际帧率为 12.5fps。《4.4.2 3.0 kernel board 文件配置》章节说明的各个分辨率帧率宏直接填写该信息，即 12504；

红字注释表示的是当前分辨率并非 sensor 直接输出的（**sensor 驱动不具备当前分辨率的设置序列**），而是经过裁剪得到的，所以并非完整的视窗，所以**该分辨率不适合作为 dv 分辨率**。

以上红字说明在 Camera 驱动更新至 0.1.7 及其以上版本之后有所变更：

0.1.7 及其以上版本的 Camera 驱动对于 sensor 无法提供的分辨率采用 IPP 进行先 Crop+Scale 的方案进行处理，这样一来视窗的取景问题就不是很明显，可以忽略；但是由于有经过 IPP 进行处理，所以在带宽上还是有较大影响的，建议如下：

1、针对提示 WARNING 的分辨率，如果不是由 1280x1024 及其以上分辨率处理获得的（即 WARNING 提示信息中提到的 crop from 分辨率），那么可以作为 DV 的分辨率，即可以填写如 media_profiles.xml 文件；

Android4.0 中 dv 应用只识别 1080p、720p、480p、qvga、cif、qcif 6 种分辨率，只有在定义支持 1080p、720p、480p 后，dv 的菜单中才会出现相关分辨率切换菜单。如果 1080p、720p、480p 都不支持，那么就按照定义的 high 分辨率进行录像。Android 程序根据 media_profiles.xml 定义的各项分辨率来查找找到 high 和 low 2 中质量的分辨率选项，但是如果 xml 文件中自己定义了 high 或是 low 质量的分辨率选项，那么以用户定义的为准。其中各分辨率详细信息：

1080p: 1920x1080、 1920x1088
720p: 1280x720p
480p: 720x480、 640x480、 704x480
qvga: 320x240
cif: 352x288
qcif: 176x144

如上所示，以上分辨率支持的话，在 xml 文件中即可定义相应质量名称（1080p、720p、480p、qvga、cif、qcif）选项。如果要定义这几种分辨率以外的分辨率选项，那么请将其定义在 high、low 质量的分辨率选项中，但是这样 cts 测试的 CamcorderProfileTest—testGet 会检查失败。如果分辨率是都是符合 6 种定义要求，那么不需要定义 high、low 质量的分辨率选项，不然定义不符合要求也会出现 CamcorderProfileTest—testGet 测试失败。

如下，红色部分的配置信息可以通过 camera_test 来获取，framerate 在 camera_test 得到的非整数值在填入 media_profiles.xml 时取整：

```
<EncoderProfile quality="high" fileFormat="mp4" duration="30">
  <Video codec="h264"
    bitRate="256000"
    width="640"
    height="480"
    frameRate="15" />
  <Audio codec="amrnb"
    bitRate="12200"
    sampleRate="8000"
    channels="1" />
</EncoderProfile>
<EncoderProfile quality="timelapsehigh" fileFormat="mp4" duration="30">
  <Video codec="h264"
    bitRate="1200000"
    width="640"
    height="480"
    frameRate="15" />
  <!-- audio setting is ignored -->
  <Audio codec="aac"
    bitRate="96000"
    sampleRate="48000"
    channels="1" />
</EncoderProfile>
```

注意每个分辨率都要有 2 中配置，quality="XXX"和 quality="timelapseXXX"。

原则上我们选择分辨率时最好是 sensor 直接输出的，而不是 IPP 缩放得到的。因为如果选择的分辨率是 IPP 缩放得到的，拍出来的视频有可能会变形。

8.2 4.0. Panorama(全景拍照) and FaceLock (人脸解锁)

4.0 中 Panorama(全景拍照)针对的是后置摄像头，如果机器没有后置摄像头的话，那么该功能无效；

4.0 中 FaceLock (人脸解锁) 针对的是前置摄像头，如果机器没有前置摄像头的话，那么该功能无效；

摄像头的前置、后置配置详见《4.4.2 3.0 kernel board 文件配置》章节的说明

9 android camera 模块各项目 CTS 测试注意事项

testPreviewFpsRange 测试

9.1.1 android 2.3 版本 testPreviewFpsRange

该项测试主要检查 camera 的实际帧率是否与登记的帧率符合。各个项目中的 sensor 的实际帧率不一，所以针对该项测试，各个项目需要对登记帧率进行修改。针对 2.00sdk 以及 1.28sdk 打了 camera 相关补丁 20110826, 需要修改 hardware/rk29/camera/CameraHal.cpp 中:

```
594 /*frame per second setting*/
595 parameterString = "15000,15000";
596 params.set(CameraParameters::KEY_PREVIEW_FPS_RANGE, parameterString.string());
597 parameterString = "{15000,15000}";
598 params.set(CameraParameters::KEY_SUPPORTED_PREVIEW_FPS_RANGE, parameterString.string());
```

针对 drivers/media/video/rk29_camera_oneframe.c 驱动的 0.0.3 版本:

```
00132: //Configure Macro
00133: #define RK29_CAM_VERSION_CODE KERNEL_VERSION(0, 0, 3)
```

Android 中 hardware/rk29/camera/CameraHal.h 中具备以下配置宏:

```
00051:
00052: #define CONFIG_AUTO_DETECT_FRAMERATE 0
00053:
00054: #if CONFIG_AUTO_DETECT_FRAMERATE
00055: #define CAMERA_DEFAULT_PREVIEW_FPS_MIN 8000 //8 fps
00056: #define CAMERA_DEFAULT_PREVIEW_FPS_MAX 15000
00057: #else
00058: #define CAMERA_FRONT_PREVIEW_FPS_MIN 8000 //8 fps
00059: #define CAMERA_FRONT_PREVIEW_FPS_MAX 15000
00060: #define CAMERA_BACK_PREVIEW_FPS_MIN 8000
00061: #define CAMERA_BACK_PREVIEW_FPS_MAX 15000
00062: #endif
00063:
```

该版本支持 android 在启动时自动检测 camera 的所有支持预览分辨率的帧率，即将 CONFIG_AUTO_DETECT_FRAMERATE 宏打开，但是该项功能打开后，在系统启动时会耗时接近 30s 检测 2 个摄像头的帧率，在尚未检测完毕时，进入 camera 会黑屏等待。如果关闭自动检测功能，麻烦配置上图所示的各个宏:

```
#define CAMERA_FRONT_PREVIEW_FPS_MIN 8000 //前置 sensor 帧率最小值
#define CAMERA_FRONT_PREVIEW_FPS_MAX 15000
#define CAMERA_BACK_PREVIEW_FPS_MIN 8000 //后置 sensor 帧率最小值
#define CAMERA_BACK_PREVIEW_FPS_MAX 15000
```

9.1.2 android 4.0.3 版本 testPreviewFpsRange

4.0.3 版本测试该项时，Camera 硬件抽象层从 kernel 获取相应 camera 的各个分辨率的帧率信息。Kernel 中各个项目的帧率信息设置详见《4.4.2 3.0 kernel board 文件配置》章节。这边需要注意: rk29_camera_oneframe.c (camera 驱动) 的版本在 v0.x.5 时支持帧率测试(配

合 camera_test 工具), 但是测试出来的帧率存在 2fps 的误差, 所以在 testPreviewFpsRange 测试中各项目经常出现失败, 针对该项测试需要更新至 v0.x.6 版本以上的 camera 驱动, 然后配置 camera_test_v1.0 工具重新测试帧率, 该工具测试出来的帧率信息详见《8.1.2 android 4.0 media_profile.xml》第 1 小节, 输出的帧率信息直接填写到 kernel board (《4.4.2 3.0 kernel board 文件配置》章节说明) 文件中的各个帧率对应宏中。

CameraHal v0.2.8

CameraHal v0.2.2 版本支持该项测试通过, 但是每个预览分辨率 (KEY_SUPPORTED_PREVIEW_SIZES) 的帧率信息 (KEY_SUPPORTED_PREVIEW_FPS_RANGE) 由查询 kernel board (《4.4.2 3.0 kernel board 文件配置》章节说明) 文件中的各个帧率对应宏来确定。但是这些宏定义只适合 sensor 在各个分辨率输出时固定帧率的情况, 在某些 sensor 的配置中会将 sensor 的输出配置成动态帧率的, 这样在测试该项测试时会出现概率性的测试不通过现象。CameraHal v0.2.8 版本在填写 KEY_SUPPORTED_PREVIEW_FPS_RANGE 等帧率信息时, 修改成如下方式, 将 kernel board 文件中定义各个分辨率帧率的最大值和最小值作为 KEY_SUPPORTED_PREVIEW_FPS_RANGE 帧率信息的最大值和最小值。如果查询不到某项分辨率的帧率信息, 采用 CameraHal.h 文件中 CONFIG_CAMERA_FRONT_PREVIEW_FPS_MIN/CONFIG_CAMERA_FRONT_PREVIEW_FPS_MAX/CONFIG_CAMERA_BACK_PREVIEW_FPS_MIN/CONFIG_CAMERA_BACK_PREVIEW_FPS_MAX 来分别定义前后置摄像头的最大最小帧率。如上所述, 在 CameraHal v0.2.8 版本针对该项测试配置如下:

- 1、根据 CameraTest 测试帧率填写 kernel board 文件中各个帧率宏定义信息;
- 2、填写 CamerHal.h 文件中

CONFIG_CAMERA_FRONT_PREVIEW_FPS_MIN/CONFIG_CAMERA_FRONT_PREVIEW_FPS_MAX/CONFIG_CAMERA_BACK_PREVIEW_FPS_MIN/CONFIG_CAMERA_BACK_PREVIEW_FPS_MAX 这 4 个宏定义信息;

- 3、屏蔽 kernel board 文件中 CONFIG_SENSOR_QCIF_FPS_FIXED_XX 的宏定义, 这样 CameraHal .h 定义的宏信息就生效;

CameraHal v0.2.a

- 1、根据 CameraTest 测试帧率填写 kernel board 文件中各个帧率宏定义信息;
- 2、填写 CamerHal.h 文件中

CONFIG_CAMERA_FRONT_PREVIEW_FPS_MIN/CONFIG_CAMERA_FRONT_PREVIEW_FPS_MAX/CONFIG_CAMERA_BACK_PREVIEW_FPS_MIN/CONFIG_CAMERA_BACK_PREVIEW_FPS_MAX 这 4 个宏定义信息;

- 3、在 0.2.a 版本中, 第 2 点中的宏定义立即生效, 不需要屏蔽 kernel board 文件中的宏定义;

android.hardware.cts.CameraGLTest 测试

该项测试只有 4.0.3 版本 android 有要求, 2.3android 没有该类要求。需要注意事项如下:

1)、testCameraToSurfaceTextureMetadata

该项主要测试在打开 camera 之后，Surface 合成的速率是否能够符合 camera 的帧率要求，由于合成速率的问题，CameraHal 的预览缓冲需要开到 4 个（版本更新至 v0.2.1）。由于 cts 测试程序的要求，针对各个预览分辨率都必须能够支持 KEY_SUPPORTED_PREVIEW_FPS_RANGE，目前我们公司的 sensor 驱动针对各个分辨率的帧率可能没有不统一，这样就会导致该测试项测试不过。简单要求如下：

- (1)、各个分辨率的帧率一样；
- (2)、如果各分辨率的帧率不一样，那么必须与最低帧率成倍数关系，例如：5fps、10fps、15fps.....;

CameraHal v0.2.8 版本的配置与 testPreviewFpsRange 测试配置说明一致；

9.1、9.2 两个章节所提到的测试，如果出现测试不过，并且是提示 720p 分辨率时出错，麻烦各项目检查各自 kernel 的 sensor 驱动文件，如果 sensor 没有直接提供 720p 序列，注意 sensor_try_fmt 函数的实现是否有以下代码，如果没有，请参考 ov2659.c 的代码实现：

```
02021: static int sensor_try_fmt(struct v4l2_subdev *sd, struct v4l2_mbus_framefmt *mf)
02022: {
02023:     struct i2c_client *client = v4l2_get_subdevdata(sd);
02024:     struct sensor *sensor = to_sensor(client);
02025:     const struct sensor_datafmt *fmt;
02026:     int ret = 0, set_w, set_h;
02027:
02028:     fmt = sensor_find_datafmt(mf->code, sensor_colour_fmts,
02029:                             ARRAY_SIZE(sensor_colour_fmts));
02030:     if (fmt == NULL) {
02031:         fmt = &sensor->info_priv.fmt;
02032:         mf->code = fmt->code;
02033:     }
02034:
02035:     if (mf->height > SENSOR_MAX_HEIGHT)
02036:         mf->height = SENSOR_MAX_HEIGHT;
02037:     else if (mf->height < SENSOR_MIN_HEIGHT)
02038:         mf->height = SENSOR_MIN_HEIGHT;
02039:
02040:     if (mf->width > SENSOR_MAX_WIDTH)
02041:         mf->width = SENSOR_MAX_WIDTH;
02042:     else if (mf->width < SENSOR_MIN_WIDTH)
02043:         mf->width = SENSOR_MIN_WIDTH;
02044:
02045:     set_w = mf->width;
02046:     set_h = mf->height;
02047:
02048:     if (((set_w <= 176) && (set_h <= 144)) && sensor_qcif[0].reg)
02049:     {
02050:         set_w = 176;
02051:         set_h = 144;
02052:     }
02053:     else if (((set_w <= 320) && (set_h <= 240)) && sensor_qvga[0].reg)
02054:     {
02055:         set_w = 320;
```

```

02056:         set_h = 240;
02057:     }
02058:     else if (((set_w <= 352) && (set_h <= 288)) && sensor_cif[0].reg)
02059:     {
02060:         set_w = 352;
02061:         set_h = 288;
02062:     }
02063:     else if (((set_w <= 640) && (set_h <= 480)) && sensor_vga[0].reg)
02064:     {
02065:         set_w = 640;
02066:         set_h = 480;
02067:     }
02068:     else if (((set_w <= 800) && (set_h <= 600)) && sensor_svga[0].reg)
02069:     {
02070:         set_w = 800;
02071:         set_h = 600;
02072:     }
02073:     else if (((set_w <= 1280) && (set_h <= 720)) && sensor_720p[0].reg)
02074:     {
02075:         set_w = 1280;
02076:         set_h = 720;
02077:     }
02078:     else if (((set_w <= 1024) && (set_h <= 768)) && sensor_xga[0].reg)
02079:     {
02080:         set_w = 1024;
02081:         set_h = 768;
02082:     }
02083:     else if (((set_w <= 1280) && (set_h <= 1024)) && sensor_sxga[0].reg)
02084:     {
02085:         set_w = 1280;
02086:         set_h = 1024;
02087:     }
02088:     else if (((set_w <= 1600) && (set_h <= 1200)) && sensor_uxga[0].reg)
02089:     {
02090:         set_w = 1600;
02091:         set_h = 1200;
02092:     }
02093:     else
02094:     {
02095:         /* ddl@rock-chips.com : Sensor output smallest size if isn't support app */
02096:         set_w = SENSOR_INIT_WIDTH;
02097:         set_h = SENSOR_INIT_HEIGHT;
02098:     }
02099:     mf->width = set_w;

02100:     mf->height = set_h;
02101:     mf->colorspace = fmt->colorspace;
02102:     return ret;
02103: }
02104: }

```

2)、testSetPreviewTexturePreviewCallback

该项测试在 android 4.0 CameraHal v0.3.5 之前版本可能出现概率性不过，导致后续 camera 测试都失败的现象，麻烦更新至 v0.3.5.

android.hardware.cts. SystemFeaturesTest 测试

1)、testCameraFeatures

该项测试会检查 CameraHal 中支持的后置 sensor、后置 sensor auto focus、后置 sensor flash、前置 sensor 4 项配置与/etc/permissions 目录中各 xml 文件定义的 feaure 是否匹配。如果不匹配，麻烦各项目人员修改 device\rockchip\rk29sdk 目录下 device.mk 文件中对相关 xml 文件的拷贝定义；

android.hardware.camera.flash-autofocus.xml

后置 sensor、后置 sensor auto focus、后置 sensor flash

android.hardware.camera.autofocus

后置 sensor、后置 sensor auto focus

android.hardware.camera.front

前置 sensor

android.hardware.camera

后置 sensor

注意 handheld_core_hardware.xml 文件中有
<feature name="android.hardware.camera" />
表示的是后置摄像头，与 android.hardware.camera 文件一致

android.hardware.cts. CamcorderProfileTest 测试

1)、testGet

该项测试针对 media_profiles.xml 文件定义的合法性进行检查。麻烦详见《8.1.2 android 4.0 media_profile.xml》章节的“sensor 帧率测试小节”。

如果项目中只有一个 sensor，且该 sensor 在 kernel 在 board 文件中配置成前置 sensor，那么必须将 CameraHal 模块的版本更新至 v0.2.3 及其以上版本。

单个前置摄像头 CTS 测试注意事项

关于机器只有前置摄像头，没有后置摄像头的机器过 9.4 以及 9.3 这 2 项测试时注意点如下：

针对 android-cts-4.0.3_r2 以及 android-cts-4.0.3_r1:

1)、在 testGet 测试的是后置摄像头，所以在打上 Camera_Patch_v1.1(即 CameraHal 版本 v0.2.7 及其以上版本)，这一项才可以通过；

2)、打上 Camera_Patch_v1.1(即 CameraHal 版本 v0.2.7 及其以上版本)之后，机器其实在软件上认为是 2 个摄像头，但是打开后置时打开的其实是前置。所以 9.3 项中提到的 feaure 就必须包含后置摄像头才可以通过 9.3 项。Media_profiles.xml 文件中也必须包含后置摄像头的 dv 信息，前后置信息一致；

3)、v0.2.7 以及以后版本是否打开该项功能由 CameraHal.h 中的 CONFIG_CAMERA_SINGLE_SENSOR_FORCE_BACK_FOR_CTS 该宏来控制；

注意这样打开之后，camera 应用中就会出现 2 个摄像头，这个无法避免；

针对 android-cts-4.0.3_r3:

这个版本的 CTS，只要不需要按照 r2 以及 r1 步骤来操作，只需按照 9.4 以及 9.3 步骤操作即可。

1、CameraHal 版本 v0.2.7 及其以上版本必须保证 CONFIG_CAMERA_SINGLE_SENSOR_FORCE_BACK_FOR_CTS 宏定义为 0；

2、android /etc/ permissions 目录中各 xml 文件定义的 feaure，只能包含前置摄像头，不能包括后置摄像头；

10 android camera 摄像头模组方向说明

在 android 2.3 目录下 gingerbread\hardware\rk29\camera\CameraHal.cpp 中，如下代码为设置摄像头方向信息：

```

extern "C" void HAL_getCameraInfo(int cameraId, struct CameraInfo *cameraInfo)
{
    char property[PROPERTY_VALUE_MAX];
    int hwrotation;

    memcpy(cameraInfo, &cameraInfo[cameraId], sizeof(CameraInfo));
    if (screenIsTall == false) {
        property_get("ro.sf.hwrotation", property, "0");
        hwrotation = strtol(property, 0, 0);
        if ((hwrotation == 90) || (hwrotation == 270)) {
            if (cameraInfo->facing == CAMERA_FACING_BACK)
                cameraInfo->orientation = 90;
            else
                cameraInfo->orientation = 270;
        }
    } else {
        if (cameraInfo->facing == CAMERA_FACING_BACK)
            cameraInfo->orientation = 90;
        else
            cameraInfo->orientation = 270;
    }

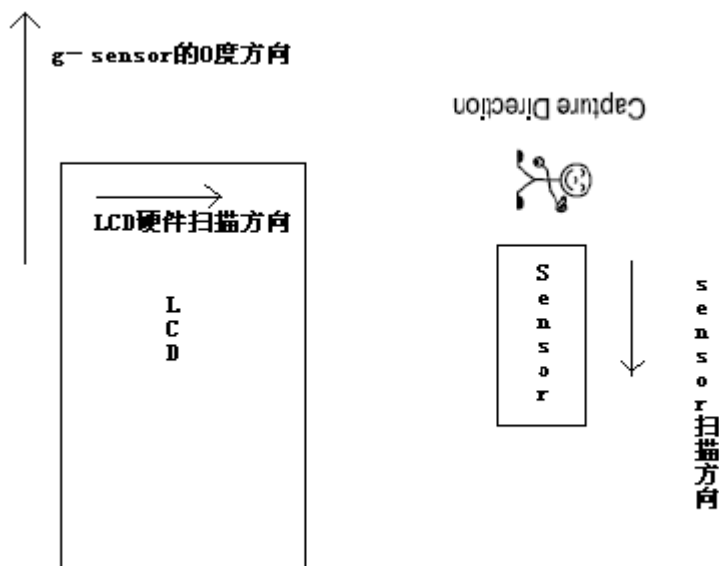
    LOGD("HAL_getCameraInfo:%d ro.sf.hwrotation:%d screenIsTall:%d", cameraInfo->orientation, hwrot
}

```

在 android 4.0 + kernel 3.0 版本中，已经将该定义移至 kernel 的 board 文件中，麻烦参考“4.4.2 3.0 kernel board 文件配置”章节。

在 android 中，camera 应用使用的是横屏显示，所以在硬件 PCB 设计时，摄像头模组的放置方向是和 LCD 屏的放置方向有关的。原则上必须符合如下规则：

- 1、摄像头模组扫描方向为横向，摄像头模组的长边必须与 LCD 屏的长边平行；如下图所示，（注释：按照模组厂一般规则，其模组规格书中，小人两手方向为 sensor 的长边方向，但是该规则不知是否所有模组厂都一致，这一点需要在硬件设计时与模组厂确认）



按上图所示，前置 sensor 在 CameraHal.cpp 中关于方向的设置应该为 270，如果朝向左边的 LCD 屏长边，那么设置成 90。

后置 sensor 的角度设置成 90 度，如果朝向左边的 LCD 屏长边，那么设置成 270。

按照上图放置 sensor 之后，最后 LCD 屏上显示图像与实际图像出现镜像效果：

- 1、竖屏时，图像上下颠倒，横屏时图像左右颠倒，那么请调整 Sensor 的 Mirror 寄存器（或称为 Flip_x/Flip_h）；-----左右镜像
- 2、竖屏时图像左右颠倒，横屏时图像上下颠倒，那么请调整 Sensor 的 Flip 寄存器（或称为 Flip_y/Flip_v）；-----垂直镜像

3、调整 sensor 的镜像时，修改的是 kernel 的 sensor 驱动，sensor 驱动中相关分辨率序列数组中有关寄存器内容都必须修改：

```
static struct reginfo sensor_init_data[] =
{
};

/* 720p 15fps @ 1280x720 */
static struct reginfo sensor_720p[] =
{
};

/* 1080p, 0x15fps, 0xyuv @1920x1080 */
static struct reginfo sensor_1080p[] =
{
};

/* 2592X1944 QSXGA */
static struct reginfo sensor_qsxga[] =
{
};

/* 2048*1536 QXGA */
static struct reginfo sensor_qxga[] =
{
};
```

4、各个 sensor 的 mirror 以及 flip 寄存器如下，**仅供参考，以实际 sensor datasheet 为准**

GC0308(i2c addr: 0x42):

P0:0x14	CISCTL_Mode1	8	0x00	RW	<p>[7] hsync_always 1: hsync always on 0: hsync output at active output</p> <p>[6] NA</p> <p>[5:4] CFA sequence, determined once color filter is determined</p> <p>[3:2] NA</p> <p>[1] upside down</p> <p>[0] mirror</p>
---------	--------------	---	------	----	--

Gc0309(i2c addr: 0x42):

P0:0x14	CISCTL_Mode1	8	0x00	RW	[7] hsync_always 1: hsync always on 0: hsync output at active output [6] NA [5:4] CFA sequence, determined once color filter is determined [3:2] NA [1] upside down [0] mirror
---------	--------------	---	------	----	---

Gc0329(i2c addr: 0x62):

Function	Register Address	Register Value
正常图像	0x17[1:0]	00
镜像翻转	0x17[1:0]	01
垂直翻转	0x17[1:0]	10
镜像垂直翻转	0x17[1:0]	11

Gc2015(i2c addr: 0x60):

P0:0x29	CISCTL_mode1	8	0X20	RW	[7] HSYNC always [6] Close 2 frame dbrow [5:4] CFA sequence [3:2] dark CFA sequence [1] Updown image [0] mirror image
---------	--------------	---	------	----	--

Gc2035(i2c addr: 0x78):

P0:0x17	Mirror_updn	8	0X00	RW	[7:3] Reserved [1] Flip [0] mirror
---------	-------------	---	------	----	--

Gt2005(i2c addr: 0x78):

0x0101	VREVON	1	0x00	RW	[1] VREVON Selection verical flip 0h : Normal 1h : vertical flip
	HREVON	1			[0] HREVON Selection Horizontal mirror 0h : Normal 1h : Horizontal mirror

Hi253(i2c addr: 0x40):

2.4. 方向调整



```
//A: Normal
write_cmos_sensor(0x03, 0x00); //page 00
write_cmos_sensor(0x11, 0x90); //B[1:0] set "0"

//B: X_flip
write_cmos_sensor(0x03, 0x00); //page 00
write_cmos_sensor(0x11, 0x91); //B[0] set "1"

//C: Y_Flip
write_cmos_sensor(0x03, 0x00); //page 00
write_cmos_sensor(0x11, 0x92); //B[1] set "1"

//D: X_Y flip
write_cmos_sensor(0x03, 0x00); //page 00
write_cmos_sensor(0x11, 0x93); //B[1:0] set "1"
```

Hi704(i2c addr: 0x60):

0x11 [page mode 0]: VDOCTL2 [default=0x90, r/w]

Bit	Function	Description	Default
B[7]	Windowing	User changes image size by setting WINROW[0x20,0x21:P0], WINGCOL [0x22,0x23:P0], WINHGT [0x24,0x25:P0] and WINWID[0x26,0x27:P0]. (0:OFF, 1:ON)	1b
B[6:4]	Bad Frame Skip	It is used to skip bad frames when image size is changed. 001: Skip 1frame. 010: Skip 2frames, 011: Skip 3frames Note) Do not set 0.	001b
B[3]	Fixed Frame Rate2	It is used to exclude VBLANK at frame rate when frame time to be constant. Refer to 5.13[Fixed Frame Rate Timing]	0b
B[2]	Fixed Frame Rate1	Set frame time to be constant, regardless of the change of exposure time. (0:OFF, 1:ON) Refer to 5.13[Fixed Frame Rate Timing]	0b
B[1]	Y Flip	Vertical Flip Function (0:OFF, 1:ON)	0b
B[0]	X Flip	Horizontal Flip Function (0:OFF, 1:ON)	0b

Mt9d112(i2c addr: 0x7a/0x78):

59 0x03B	1	0x0000	Vertical Flip 0 = Normal readout 1 = Readout is flipped (mirrored) vertically so that the row specified by y_addr_end_ is read out of the sensor first. Setting this bit will change the bayer pixel order (see Reg0x3024). The bit-order of bits [1:0] match the order in Reg0x301C but is reversed relative to earlier Micron Imaging sensors.
---------------------------	---	--------	---

Micron Confidential and Proprietary

Preliminary



MT9D112: 1/4-Inch 2-Mp SOC Digital Image Sensor Register Description

Table 35: 7: Mode Variables (continued)

Reg. #	Bits	Default	Name
	0	0x0000	Horizontal Mirror 0 = Normal readout 1 = Readout is mirrored horizontally so that the column specified by x_addr_end_ is read out of the sensor first. Setting this bit will change the bayer pixel order (see Reg0x3024).
context B shadow register R0x20:0. Changes take effect only after REFRESH_MODE command.			

Nt99250(i2c addr: 0x6c):

0x3022	Read_Mode_0	7:0	R/W	0x24	
	Y_Even_Inc	7:5	R/W	1	Increment applied to even addresses in Y (row) direction ↑ 1: Normal readout
	X_Even_Inc	4:2	R/W	1	Increment applied to even addresses in X (column) direction ↑ 1: Normal readout
	Mirror _Horizontal	1	R/W	0	Horizontal mirror 0: Normal readout 1: Mirror readout
	Flip_Vertical	0	R/W	0	Vertical flip 0: Normal readout 1: Flip readout

Ov2640(i2c addr: 0x60):

				(8 MSBS in VSTR1[7:0] (0x19))
04	REG04	20	RW	<p>Register 04</p> <p>Bit[7]: Horizontal mirror</p> <p>Bit[6]: Vertical flip</p> <p>Bit[4]: VREF bit[0]</p> <p>Bit[3]: HREF bit[0]</p> <p>Bit[2]: Reserved</p> <p>Bit[1:0]: AEC[1:0] (AEC[15:10] is in register REG45[5:0] (0x45), AEC[9:2] is in register AEC[7:0] (0x10))</p>

Ov2655(i2c addr: 0x60):

i2c_salve_Address = 0x60;

MIRROR
write_i2c(0x3090, 0x08);

FLIP
write_i2c(0x307c, 0x01);flip

MIRROR&FLIP
write_i2c(0x307c, 0x01)
write_i2c(0x3090, 0x08);

NORML
write_i2c(0x307c, 0x00);no mirror/flip
write_i2c(0x3090, 0x08);

Ov2659(i2c addr: 0x60):

MIRROR

20

OV2659 Camera Module Softwa

write_i2c(0x3821, 0x07)
write_i2c(0x3820, 0x81)

FLIP
write_i2c(0x3821, 0x01)
write_i2c(0x3820, 0x87)

MIRROR&FLIP
write_i2c(0x3821, 0x07)
write_i2c(0x3820, 0x87)

NORML
write_i2c(0x3821, 0x01)
write_i2c(0x3820, 0x81)

Ov3640(i2c addr: 0x78):

```
i2c_salve_Address = 0x78;

MIRROR
write_i2c(0x307c, 0x12);mirror
write_i2c(0x3090, 0xc8);
write_i2c(0x3023, 0x0a);

FLIP
write_i2c(0x307c, 0x11);flip
write_i2c(0x3023, 0x09);
write_i2c(0x3090, 0xc0);

MIRROR&FLIP
write_i2c(0x307c, 0x13);flip/mirror
write_i2c(0x3023, 0x09);
write_i2c(0x3090, 0xc8);

NORML
write_i2c(0x307c, 0x10);no mirror/flip
write_i2c(0x3090, 0xc0);
write_i2c(0x3023, 0x0a);
```

Ov5642(i2c addr: 0x78):

```
.....
i2c_salve_Address = 0x78;

MIRROR
write_i2c(0x3818, 0x81);
write_i2c(0x3621, 0xe7);

FLIP
write_i2c(0x3818, 0xe1);
write_i2c(0x3621, 0xc7);

MIRROR&FLIP
write_i2c(0x3818, 0xa1);
write_i2c(0x3621, 0xe7);
```

NORML

```
write_i2c(0x3818, 0xc1);
write_i2c(0x3621, 0xc7);
```

Ov5640: (该设置似乎有问题, 麻烦联系 ov FAE)

Ov7670:

OV7670/OV7171 CMOS VGA (OmniPixel®) CAMERACHIP™ Sensor

Table 5 Device Control Register List (Continued)

Address (Hex)	Register Name	Default (Hex)	R/W	Description
1E	MVFP	01	RW	Mirror/VFlip Enable Bit[7:6]: Reserved Bit[5]: Mirror 0: Normal image 1: Mirror image Bit[4]: VFlip enable 0: Normal image 1: Vertically flip image Bit[3]: Reserved Bit[2]: Black sun enable Bit[1:0]: Reserved

Ov7675(i2c addr: 0x78):

figure 4-1 mirror and flip samples

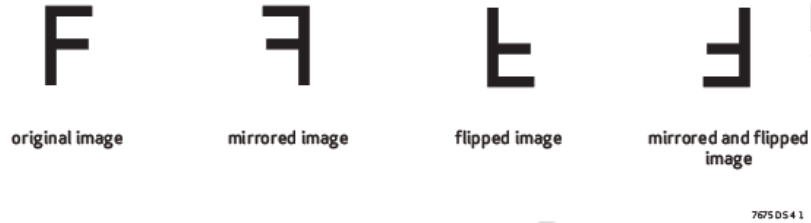


table 4-1 image windowing control functions

address	register name	default value	R/W	description
0x1E	MVFP	0x01	RW	Mirror/VFlip Enable Bit[5]: Mirror 0: Normal image 1: Mirror image Bit[4]: VFlip enable 0: Normal image 1: Vertically flip image

Sid103b(i2c addr: 0x37):

address	register name	default value	description	R/W
0x04	CNTR_B	0x00	1 Dynamic mode – Normal operation mode [7:6] Control fixed frame mode 00 Normal operation mode 01 Fixed frame mode @ exposure time <= frame size Normal operation mode @ exposure time > frame size 10/11 Fixed frame mode [4] Select register group between timing Group A & Group B 0 : Select Group_B Registers 1 : Select Group_A Registers [3:2] Clock (PCLK) divider – max 1/8 PCLK 00: PCLK 01: 1/2PCLK 10: 1/4PCLK 11: 1/8PCLK [1] Control vertical flip image 0 Normal image 1 Vertical flip image [0] Control horizontal flip (mirror) image 0 Normal image 1 Horizontal flip (mirror) image	R/W

11 Camera Digital Zoom

Rk2918 camera driver v0.0.4 及其以上版本开始支持数码变焦功能，

```
00136: //Configure Macro
00137: /*
00138: *           Driver Version Note
00139: *v0.0.1 : this driver first support rk2918;
00140: *v0.0.2 : fix this driver support v4l2 format is V4L2_PIX_FMT_M
00141: *           and V4L2_PIX_FMT_YUV422P;
00142: *v0.0.3 : this driver support VIDIOC_ENUM_FRAMEINTERVALS;
00143: *v0.0.4 : this driver support digital zoom;
00144: */
00145: #define RK29_CAM_VERSION_CODE KERNEL_VERSION(0, 0, 4)
00146:
```

针对数码变焦功能，驱动的实现方法是依旧 IPP 的裁剪缩放功能实现的，所以在实现上必须事先将 sensor 数据采集到内存中，然后经过 IPP 裁剪放大到用户指定的内存区域。同言之，为了支持数码变焦，必须预先分配足够的内存给 vip 采集，内存分配值见

[rk29_camera.c (drivers\media\video)] 文件中 PMEM_CAMIPP_NECESSARY:

```
00040: #if (PMEM_CAM_FULL_RESOLUTION == 0x500000)
00041: #define PMEM_CAM_NECESSARY 0x1200000 /* 1280*720*1.5*4(preview) + 7.5M(
00042: #define PMEM_CAMIPP_NECESSARY 0x800000
00043: #elif (PMEM_CAM_FULL_RESOLUTION == 0x300000)
00044: #define PMEM_CAM_NECESSARY 0xe00000 /* 1280*720*1.5*4(preview) + 4.5M(
00045: #define PMEM_CAMIPP_NECESSARY 0x500000
00046: #elif (PMEM_CAM_FULL_RESOLUTION == 0x200000) /* 1280*720*1.5*4(preview) + 3M(ca
00047: #define PMEM_CAM_NECESSARY 0xc00000
00048: #define PMEM_CAMIPP_NECESSARY 0x400000
00049: #elif ((PMEM_CAM_FULL_RESOLUTION == 0x100000) || (PMEM_CAM_FULL_RESOLUTION == 0
00050: #define PMEM_CAM_NECESSARY 0x800000 /* 800*600*1.5*4(preview) + 2M(cap
00051: #define PMEM_CAMIPP_NECESSARY 0x400000
00052: #elif (PMEM_CAM_FULL_RESOLUTION == 0x30000)
00053: #define PMEM_CAM_NECESSARY 0x400000 /* 640*480*1.5*4(preview) + 1M(cap
00054: #define PMEM_CAMIPP_NECESSARY 0x400000
00055: #else
00056: #define PMEM_CAM_NECESSARY 0x1200000
00057: #define PMEM_CAMIPP_NECESSARY 0x800000
00058: #endif
```

内核 menuconfig 中:

<*>RK29XX Camera Sensor Interface driver

RK29XX Camera Sensor Interface Work Mode (VIP OneFrame Mode) --->

RK29XX camera sensor interface work with IPP (VIP work with IPP) --->

RK29XX camera digital zoom with IPP (Digital zoom with IPP on) --->

红色标识的 config 配置决定了是否分配 PMEM_CAMIPP_NECESSARY 内存。

12 Camera Memory

CameraHal v0.2.4 + Camera Driver v0.1.8 版本及其以上版本支持 Camera 模块必需内存（预览内存 preview buffer、拍照内存 raw buffer、编码输出内存 JPEG buffer）从 ION 模块中动态分配得到，同时兼容原来 pmem 预留内存中分配的方式。采用何种分配方式由以下宏来决定：

Kernel : menuconfig:

Device Drivers --->

Multimedia support --->

Video capture adapters --->

RK29XX Camera Sensor Interface driver

RK29XX Camera Sensor Interface Work Mode (VIP OneFrame Mode) --->

RK29XX camera sensor interface work with IPP (VIP work with IPP) --->

RK29XX camera digital zoom with IPP (Digital zoom with IPP on) --->

RK29XX camera memory (Camera memory from pmem) --->

Android CameraHal:

hardware\rk29\camera\CameraHal_Mem.h 文件中

```
00001: /*
00002: *Author: zyc@rock-chips.com
00003: */
00004:
00005: #define CAMERA_MEM_ION 0
00006: #define CAMERA_MEM_PMEM 1
00007: /*
00008: *NOTE:
00009: * configuration macro
00010: *
00011: */
00012: #define CONFIG_CAMERA_MEM CAMERA_MEM_PMEM
```

hardware\rk29\camera\Android.mk(增加链接 libion 库)

LOCAL_SHARED_LIBRARIES:= \

libgui\

libjpeghwenc\

libjpeg\

libyuvtorgb\

libion

Camera 内存从 ion 模块动态分配的话，那么 Camera 内存、UI 内存、VPU 内存都可以共享，达到节省内存的目的。但是这个必需要 ION 模块的支持，确认 ION 模块是否支持的方式：

Kernel 启动时有如下信息：

Rockchip ion module(version: 1.0) is successfully loaded

确认以上信息后，打开以上 camera 提到的宏配置，重新编译代码后，打开 camera 模块，android 的 log 中有以下信息：

Ion(version: 1.0) is successfully opened by camera

这时表示 camera 内存是从 ion 设备中分配的。如果发现以下信息：

`/dev/pmem_cam isn't registered, CameraHal_Mem current configuration isn't support ION memory`

那表示 CameraHal_Mem.h 中 CONFIG_CAMERA_MEM 宏配置成 pmem 模式，kernel menuconfig 中又配置成 ion 模式。

13 Camera 模块各源码版本规则说明

1、源码版本获取方式：

在 shell 命令中输入以下命令：

```
#getprop  
[sys_graphic.cam_driver.ver]: [0.1.a]  
[sys_graphic.cam_hal.ver]: [0.2.7]
```

Apk 版本获取方式：

打开 camera 应用时，android LOG 中包含版本打印：

```
sys_graphic.camera.apk.ver: 0.0.1
```

2、版本号规则说明

CamerHal v0.2.d，Kernel v0.2.b 及其以后版本，版本号尾数偶数为正式版本（即经过品质部专项测试的版本），奇数为开发版本(只经过自测)。

3、源码版本历史记录

源码版本历史记录可以分别在 CameraHal.h 以及 rk30_camera_oneframe.c，rk29_camera_oneframe.c 中查询。

14 android4.0 预览垂直及水平镜像问题说明

由于模组关系，可能会造成同样一份 sensor 驱动代码在某些产品上看起来图像是左右镜像，或者上下镜像的。配置 board 中的 sensor 旋转角度也许可以解决 camera 应用图像镜像问题，但是其他应用出来的图像可能还是不正常的，如 POCO 相机等应用（这些应用没有用到 board 中定义的相关角度信息）。为了解决此类问题，首先要确定模组是否与驱动相匹配。即在不修改 board 中旋转角度（前置 270，后置 90）情况下，打开 camera 应用查看图像是否正常（也可通过 camera_test 测试程序查看，参照 15 章说明），如果出现镜像问题则需要序列中加入镜像处理，即需要重新定义 sensor 相关序列，操作方法参照前述第 4.4.2.2 节。

15 Camera_test 测试程序使用说明

该测试程序可验证 camera 的基本功能，特别是可用来测试 sensor 各分辨率的帧率。1.1 版本兼容 29 和 30 平台。使用方法如下：

- 1、将 camera_test 文件 push 到/system/bin 目录下
- 2、执行 su 命令
- 3、chmod 777 /system/bin/camera_test
- 4、可执行 camera_test -h 获取使用帮助信息

常用参数有：

- i 测试帧率，具体使用请参照 8.1.2 节
- d 指定哪个 sensor，参数为 /dev/videoX，X 为 0 或 1
- r 指定要测试的分辨率，格式如 800x600
- z 数码变焦测试

其他还有 -f,-F,-e,-L 等参数，具体使用可通过 -h 查询。此外，使用 camera_test 测试程序也可判断出模组的方向，只要输入 camera_test -z -i 640x480 查看图像状态即可。

16 针对视频通话远端图像镜像问题说明

在 android 中，前置摄像头在本地预览显示时增加了水平镜像 mirror 处理，该处理是放在显示端进行，所以针对某些视频通话 apk（skype 等）在传送给远端的数据如果直接通过 camera 的 preview 接口获得的话，就会出现远端图像相对于本地预览图像颠倒的问题。该问题只在前置摄像头情况下存在。

CameraHal v0.3.17 版本及其以上版本支持针对特定视频通话 apk，在发送预览数据前先进行一次镜像操作，配置步骤如下：

```
00177:
00178: #define CONFIG_CAMERA_SINGLE_SENSOR_FORCE_BACK_FOR_CTS 0
00179: #define CONFIG_CAMERA_FRAME_DV_PROC_STAT 0
00180: #define CONFIG_CAMERA_FRONT_MIRROR_MDATA_CB 1
00181: #define CONFIG_CAMERA_FRONT_MIRROR_MDATA_CB_ALL 0
00182: #define CONFIG_CAMERA_FRONT_MIRROR_MDATA_CB_APK "<com.skype.raider>,"
00183: #define CONFIG_CAMERA_PREVIEW_BUF_CNT 4
00184: #define CONFIG_CAMERA_UVC_INVALID_FRAMECNT 5
00185: #define CONFIG_CAMERA_ORIENTATION_SKYPE 0
00186: #define CONFIG_CAMERA_FRONT_ORIENTATION_SKYPE 0
00187: #define CONFIG_CAMERA_BACK_ORIENTATION_SKYPE 0
00188:
```

- 1、CONFIG_CAMERA_FRONT_MIRROR_MDATA_CB 该宏配置是否打开该项功能；
- 2、CONFIG_CAMERA_FRONT_MIRROR_MDATA_CB_ALL 该宏配置是否针对所有 apk 都执行镜像动作；
- 3、在 CONFIG_CAMERA_FRONT_MIRROR_MDATA_CB_ALL 配置关闭的情况下，可以由 CONFIG_CAMERA_FRONT_MIRROR_MDATA_CB_APK 来配置需要进行镜像动作的 apk，将 apk 名称用 <> 填写入该宏即可；
- 4、针对 apk 名称的获取，可以用该 apk 打开 camera，查看 camera 以下 LOG 获得：

D/CameraHal(100): Calling process is: com.android.gallery3d

17 Camera 插值说 明

当摄像头最大分辨率较小，而又想提高分辨率时可使用插值功能。在此，以 GC0308 为例作为说明。GC0308 是 30 万摄像头，如要插值为更高像素可做如下改动。

主要修改文件为：

```
drivers/media/video/Kconfig
driver/media/video/gc0308.c
arch/arm/plat-rk/include/plat/rk_camera.h
```

使用方法：

如果要插值为 500 万，则在 menuconfig 中选中该宏 CONFIG_SOC_CAMERA_GC0308_INTERPOLATION_5M，插值到 300 万或 200 万以此类似。

其他 sensor 如果要做插值处理，可参考 GC0308 的修改，主要修改文件为以上三个文件，各文件的修改说明如下：

(1) drivers/media/video/Kconfig

在 menuconfig 中增加对应 sensor 的插值选项

```
1033 config OV5642_USER_DEFINED_SERIES
1034     depends on SOC_CAMERA_OV5642
1035     bool "OV5642 user defined init series"
1036     default n
1037
1038 config SOC_CAMERA_OV5642_INTERPOLATION
1039     bool "support sensor interpolation for higher resolution"
1040     depends on SOC_CAMERA_OV5642
1041 choice
1042     prompt "OV5642 interpolation resolution"
1043     depends on SOC_CAMERA_OV5642_INTERPOLATION
1044     default SOC_CAMERA_OV5642_INTERPOLATION_8M
1045     ---help---
1046     OV5642 interpolation resolution
1047
1048 config SOC_CAMERA_OV5642_INTERPOLATION_8M
1049     bool "8 megapixel 3264x2448"
1050 endchoice
```

(2) arch/arm/plat-rk/include/plat/rk_camera.h

根据 menuconfig 修改 sensor 的最大分辨率定义，该文件中搜索 CONFIG_SOC_CAMERA_GC0308_INTERPOLATION_5M 宏即可得知需要需要的地方。

```
00107:
00108: #define ov7675_FULL_RESOLUTION    0x30000    // 0.3 megapixel
00109: #define ov9650_FULL_RESOLUTION    0x130000    // 1.3 megapixel
00110: #define ov2640_FULL_RESOLUTION    0x200000    // 2 megapixel
00111: #define ov2655_FULL_RESOLUTION    0x200000    // 2 megapixel
00112: #define ov2659_FULL_RESOLUTION    0x200000    // 2 megapixel
00113: #define ov7690_FULL_RESOLUTION    0x300000    // 2 megapixel
00114: #define ov3640_FULL_RESOLUTION    0x300000    // 3 megapixel
00115: #define ov3660_FULL_RESOLUTION    0x300000    // 3 megapixel
00116: #define ov5640_FULL_RESOLUTION    0x500000    // 5 megapixel
00117: #if defined(CONFIG_SOC_CAMERA_OV5642_INTERPOLATION_8M)
00118:     #define ov5642_FULL_RESOLUTION    0x800000    // 8 megapixel
00119: Directive
00120:     #define ov5642_FULL_RESOLUTION    0x500000    // 5 megapixel
00121: #endif
00122: #define s5k6aa FULL_RESOLUTION    0x130000    // 1.3 megapixel
```

(3) driver/media/video/gc0308.c

主要修改最大分辨率定义， sensor_s_fmt 及 sensor_try_fmt 函数， 搜索 CONFIG_SOC_CAMERA_GC0308_INTERPOLATION_5M 宏即可得知需要需要的地方。修改地方如下图所示：

```
00047: #define SENSOR_NAME RK29_CAM_SENSOR GC0308
00048: #define SENSOR_V4L2_IDENT V4L2_IDENT_GC0308
00049: #define SENSOR_ID 0x9b
00050: #define SENSOR_MIN_WIDTH 640//176
00051: #define SENSOR_MIN_HEIGHT 480//144
00052: #define SENSOR_MAX_WIDTH_REAL 640
00053: #define SENSOR_MAX_HEIGHT_REAL 480
00054: #if defined(CONFIG_SOC_CAMERA_GC0308_INTERPOLATION_5M)
00055: #define SENSOR_MAX_WIDTH 2592
00056: #define SENSOR_MAX_HEIGHT 1944
00057: #elif defined(CONFIG_SOC_CAMERA_GC0308_INTERPOLATION_3M)
00058: #define SENSOR_MAX_WIDTH 2048
00059: #define SENSOR_MAX_HEIGHT 1536
00060: #elif defined(CONFIG_SOC_CAMERA_GC0308_INTERPOLATION_2M)
00061: #define SENSOR_MAX_WIDTH 1600
00062: #define SENSOR_MAX_HEIGHT 1200
00063: #else
00064: #define SENSOR_MAX_WIDTH SENSOR_MAX_WIDTH_REAL
00065: #define SENSOR_MAX_HEIGHT SENSOR_MAX_HEIGHT_REAL
00066: #endif
00067: #define SENSOR_INIT_WIDTH 640 /* Sensor pixel size for sensor_init_data array */
```

static int sensor_s_fmt(struct v4l2_subdev *sd, struct v4l2_mbus_framefmt *mf):

```
01843: }
01844: #if defined(CONFIG_SOC_CAMERA_GC0308_INTERPOLATION)
01845: else if ((set_w <= SENSOR_MAX_WIDTH) && (set_h <= SENSOR_MAX_HEIGHT)) )
01846: {
01847:     winsize set addr = sensor vga;
01848:     set_w = SENSOR_MAX_WIDTH_REAL;
01849:     set_h = SENSOR_MAX_HEIGHT_REAL;
01850: }
01851: #endif
01852:
```

static int sensor_try_fmt(struct v4l2_subdev *sd, struct v4l2_mbus_framefmt *mf)

```
01929: static int sensor_try_fmt(struct v4l2_subdev *sd, struct v4l2_mbus_framefmt *mf)
01930: {
01931:     struct i2c_client *client = v4l2_get_subdevdata(sd);
01932:     struct sensor *sensor = to_sensor(client);
01933:     const struct sensor_datafmt *fmt;
01934:     int ret = 0, set_w, set_h;
01935:
01936:     fmt = sensor_find_datafmt(mf->code, sensor_colour_fmts,
01937:                               ARRAY_SIZE(sensor_colour_fmts));
01938:     if (fmt == NULL) {
01939:         fmt = &sensor->info_priv.fmt;
01940:         mf->code = fmt->code;
01941:     }
01942:
01943:     /* ddl@rock-chips.com : It is query max resolution only. */
01944:     if (mf->reserved[6] == 0xfefe5a5aOperator
01945:         mf->height = SENSOR_MAX_HEIGHT;
01946:         mf->width = SENSOR_MAX_WIDTH;
01947:         ret = 0;
01948:         goto sensor_try_fmt_end;
01949: }
```

```
01988: }
01989: else if ((set_w <= 1280) && (set_h <= 1024) && sensor_sxga[0].reg)
01990: {
01991:     set_w = 1280;
01992:     set_h = 1024;
01993: }
01994: #if defined(CONFIG_SOC_CAMERA_GC0308_INTERPOLATION)
01995: else if ((set_w <= SENSOR_MAX_WIDTH) && (set_h <= SENSOR_MAX_HEIGHT)) )
01996: {
01997:     set_w = SENSOR_MAX_WIDTH_REAL;
01998:     set_h = SENSOR_MAX_HEIGHT_REAL;
01999: }
02000: #endif
```

提交内容参考如下:

Kernel:

commit 850d247dc3301521ce797ce6086473541448d268

Author: zyc <zyc@rock-chips.com>

Date: Sat Nov 3 10:29:54 2012 +0800

camera: add interpolation for gc0308 and hi253.

commit 7336bd1117d69c25ad79caf7ac79608de7819c21

Author: ddl <ddl@rock-chips.com>

Date: Wed Nov 7 11:53:13 2012 +0800

camera: add support interpolate to 8Mega, version v0.2.0x17

Android(如果插值后分辨率<=5Mega, 以下提交不需要更新):

1、 packages/apps/Camera:

commit f7428f4d1c585eae2a09066c3be16ee30b53ea87

Author: ddl <ddl@rock-chips.com>

Date: Wed Nov 7 10:03:01 2012 +0800

camera : add 8Mega menu and strings resource

2、 hardware/rk29/camera:

commit d62d244b579643aa787843832df8bbcc090c57da

Author: ddl <ddl@rock-chips.com>

Date: Wed Nov 7 10:39:37 2012 +0800

camera: add support 8Mega picture, version v0.3.0x19