

密级状态：绝密() 秘密() 内部资料() 公开(√)

RK PCBA Camera 移植说明

历史版本

版 本	日 期	描 述	作 者	审 核
V1.0	2017-7-24		黄春成	

目录

目录.....	2
1. 适用范围.....	3
2. 文件介绍.....	3
3. 移植说明.....	3
3.1. mk.....	3
3.2. 驱动文件.....	3
3.3. 主程序.....	6

1. 适用范围

目前该代码支持平台如下：

RK3368

支持 sensor 如下：

1、GC2155

2、GC2145

3、OV8858

2. 文件介绍

Android.mk: make file 文件

camera_test.c: PCBA camera 测试程序

camera_test.h: camera test 预处理信息

camsys_head.h: 与 RK kernel 交互的声明的信息

XX_CIF.c: 并口 sensor 驱动

XX_CIF_priv.h: 并口 sensor 驱动用的头文件

XX_MIPI.c: MIPI sensor 驱动

XX_MIPI_priv.h: MIPI sensor 驱动用的头文件

mrv_all_regs.h: RK ISP 寄存器信息

sys_core_ion.h: 调用 ion core 的头文件

3. 移植说明

移植新的驱动，应根据硬件设计来调整相应的配置，具体见下面详细描述。

3.1. mk

添加新的驱动要在 Android.mk 文件中添加编译相应的驱动文件，如下

```
LOCAL_SRC_FILES:=\  
    camera_test.c \  
    OV8858_MIPI.c \  
    GC2155_MIPI.c
```

3.2. 驱动文件

添加一个驱动，主要分为.c和.h两个文件。.h主要包含了与camera test文件中的交互接口信息，.c主要是驱动的具体实现，具体内容介绍如下：

1) sensor_test 数组

数组中主要包括了初始化序列以lane、分辨率等相关模块寄存器，注意：如果硬件有4lane情况下，建议使用4lane序列已达到对硬件连接的充分验证。

2) sensor_reg_init 函数

Sensor寄存器的初始化。以下是对初始化中的一些参数解释：

i2cinfo.bus_num = SENSOR_I2C_NUM;//I2C bus num

i2cinfo.slave_addr = SENSOR_I2C_ADDR;// I2C 从地址

```

i2cinfo.reg_addr = REG_SOFTWARE_RST;//I2C 寄存器地址
i2cinfo.reg_size = 2; //I2C 寄存地址位宽
i2cinfo.val = 0x01;//I2C 寄存器值
i2cinfo.val_size = 0x01;//I2C 寄存器值位宽
i2cinfo.i2cbuf_directly = 0;//I2C 方向
i2cinfo.speed = SENSOR_I2C_RATE;//I2C 速率

#define REG_SOFTWARE_RST          0x0103//soft reset 寄存器地址
//注意：当 sensor ID 寄存器只有 2byte，以下需调整为 2byte
#define REG_CHIP_ID_H             0x300a//sensor ID 寄存器高位地址
#define REG_CHIP_ID_M             0x300b//sensor ID 寄存器中位地址
#define REG_CHIP_ID_L             0x300c//sensor ID 寄存器低位地址

3) sensor_streamon 函数
   用于控制 Sensor stream 的开关，相关寄存器如下：
   #define REG_STREAM_ON          0x0100//stream 寄存器地址

4) get_SensorInfo 函数
   用于主程序获取驱动相关的配置信息，sensor 相关信息如下：
//phy 配置
   phy_type:   cif 和 mipi 区分
               CamSys_Phy_Cif: cif  □
               CamSys_Phy_Mipi:mipi □
//mipi 配置
   lane_num: lane 的数目
   bit_rate: mipi clk 速率，根据配置序列调整
   phy_index:
       0: phy0
       1: phy1
   mipi_img_data_sel:
       0x08...0x0F generic short packets
       0x12 embedded 8-bit data
       0x18 YUV 420 8-bit
       0x19 YUV 420 10-bit
       0x1A Legacy YUV 420 8-bit
       0x1C YUV 420 8-bit (CSPS)
       0x1D YUV 420 10-bit (CSPS)
       0x1E YUV 422 8-bit
       0x1F YUV 422 10-bit
       0x20 RGB 444
       0x21 RGB 555
       0x22 RGB 565
       0x23 RGB 666
       0x24 RGB 888
       0x28 RAW 6
       0x29 RAW 7

```

```

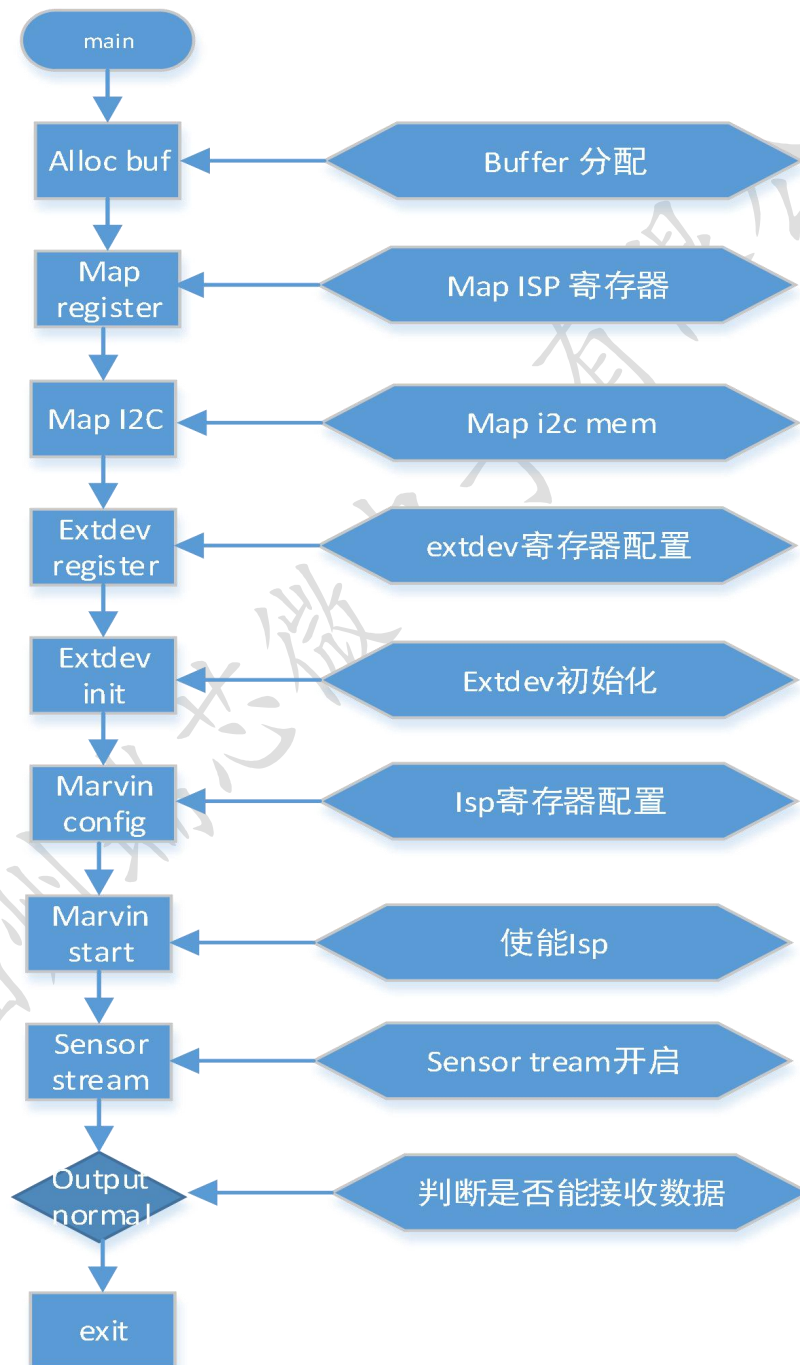
    0x2A RAW 8
    0x2B RAW 10
    0x2C RAW 12
    0x30...0x37 User Defined Byte-based data
//cif 配置
    cif_num: cif num 数目,一般为 0
    Fmt: cif formate
    Cifio: cif 管脚配置, 根据硬件设计配置
//sensor 属性配置
    Width: sensor width size
    Height: sensor heightsize
    Mode:
        000 - RAW picture with BT.601 sync (ISP bypass)
        001 - ITU-R BT.656 (YUV with embedded sync)
        010 - ITU-R BT.601 (YUV input with H and Vsync signals)
        011 - Bayer RGB processing with H and Vsync signals
        100 - data mode (ISP bypass, sync signals interpreted as data
            enable)
        101 - Bayer RGB processing with BT.656 synchronization
        110 - RAW picture with ITU-R BT.656 synchronization (ISP
            bypass)
        111 - reserved
    YCSequence:
        00- YCbYCr
        01- YCrYCb
        10- CbYCrY
        11- CrYCbY
    Conv422:
        00- co-sited color subsampling Y0Cb0Cr0 - Y1
        01- interleaved color subsampling Y0Cb0 - Y1Cr1 (not
            recommended)
        10- non-cosited color subsampling Y0Cb(0+1)/2 - Y1Cr(0+1)/2
        11- reserved
    BPat: //bayer pattern
        00- first line: RGRG..., second line: GBGB..., etc.
        01- first line: GRGR..., second line: BGBG..., etc.
        10- first line: GBGB..., second line: RGRG..., etc.
        11- first line: BGBG..., second line: GRGR..., etc.
    HPol: //horizontal polarity
        0: high active
        1: low active
    VPol: //vertical polarity
        0: high active
        1: low active
    Edge://sample edge
        0- negative edge sampling

```

1- positive edge sampling
dev_id:
CAMSYS_DEVID_SENSOR_1A:back sensor
CAMSYS_DEVID_SENSOR_1B:front
Sensorname: sensor name
Version: 驱动 version

3.3. 主程序

Camera_test 中 main 如主函数，流程示意图如下：



流程示意图

可根据上图理解整个 Camera_test 的流程，在此就不对此展开详细描述，主要对移植过程中需要注意的事项进行介绍：

- 1) 每次修改需要更新版本号，并且记录 note 信息；
- 2) 增加驱动，需要添加相应的头文件，如 `#include "GC2155_MIPI_priv.h"`；
- 3) ION_LEN 大小要根据分辨率来设定，目前为 0x1600000 可支持 1600M；
- 4) 更换 sensor 注意跟 sensor 相关名字要更改过来，如 `get_SensorInfo`、`sensor_streamon`、`sensor_reg_init`、`SENSOR_NAME` 等
- 5) 验证驱动是否正常，可先用保存图片的测试代码验证完全出图正确后，再移植用于 PCBA 的测试，`marvin_check_vsync` 仅能用作检测 sensor 能否输出数据，无法判断数据的准确性与否；
- 6) 目前不支持 2 个 sensor 同时打开，前后置可通过切换来实现；