

Save camera image

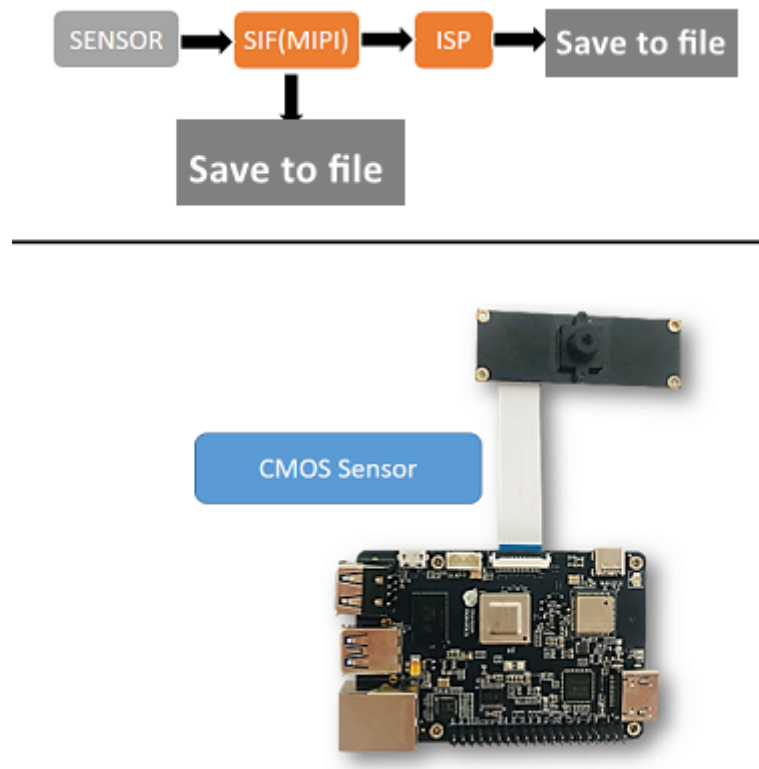
Save camera image

1. Preparation
2. Running method
3. Result

1. Preparation

This example `vio_capture` implements the `MIPI` camera image acquisition and local storage of images in two formats: `RAW` and `YUV`.

The example flow chart is as follows.



- When the development board is powered off, connect the MIPI camera to the development board (with the blue side facing up).
- Connect the development board and monitor via HDMI cable
- Power on the development board and log in to the system

2. Running method

The example code is provided in source code form and needs to be compiled and run using the 'make' command. The steps are as follows:

```
sunrise@ubuntu:~$ cd /app/cdev_demo/vio_capture/
sunrise@ubuntu:/app/cdev_demo/vio_capture$ sudo make
sunrise@ubuntu:/app/cdev_demo/vio_capture$ sudo ./capture -b 12 -c 10 -h 1080 -w 1920
```

Parameter description.

- -b: RAW image bit count, IMX477: 12, others: 10
- -c: Number of saved images
- -w: Save the width of the image
- -h: Save the height of the image

3. Result

After the program runs correctly, the current directory saves the specified number of image files.

```
sunrise@ubuntu:~$ cd /app/cdev_demo/vio_capture/
sunrise@ubuntu:/app/cdev_demo/vio_capture$ sudo make
make: Nothing to be done for 'all'.
sunrise@ubuntu:/app/cdev_demo/vio_capture$ sudo ./capture -b 12 -c 10 -h 1080 -w
1920
2024/05/27 11:26:29.849 !INFO [x3_cam_init_param][0099]Enable mipi host0 mclk
2024/05/27 11:26:29.850 !INFO [x3_cam_init_param][0099]Enable mipi host1 mclk
Camera: gpio_num=19, active=low, i2c_bus=1, mipi_host=0
Camera: gpio_num=19, active=low, i2c_bus=1, mipi_host=2
Camera 0:
    enable: 1
    i2c_bus: 1
    mipi_host: 0
Camera 1:
    enable: 1
    i2c_bus: 1
    mipi_host: 2
Camera 2:
    enable: 0
    i2c_bus: 0
    mipi_host: 0
cmd=i2ctransfer -y -f 1 w2@0x10 0x0 0x0 r1 2>&l, result=0x02

Found sensor:imx219 on i2c bus 1, use mipi host 0
Setting VPS channel-2: src_w:1920, src_h:1080; dst_w:1920, dst_h:1080;
Setting VPS channel-1: src_w:1920, src_h:1080; dst_w:1920, dst_h:1080;
capture time :0
capture time :1
capture time :2
capture time :3
capture time :4
```

The RAW format is named as raw_*.raw , while the YUV format is named as yuvv *.yuv.

The running log is as follows.

```
sunrise@ubuntu:/app/cdev_demo/vio_capture$ sudo ./capture -b 12 -c 10 -h 1080 -w
1920
Setting VPS channel-2: src_w:1920, src_h:1080; dst_w:1920, dst_h:1080;
Setting VPS channel-1: src_w:1920, src_h:1080; dst_w:1920, dst_h:1080;
jiale:start streaming...
capture time :0
capture time :1
capture time :2
capture time :3
capture time :4
capture time :5
capture time :6
capture time :7
capture time :8
capture time :9
```

```
sensor_name imx477, setting_size = 1
```

```
[ 701.213210]hb_isp_algo_stop@main_user.c:389 GENERIC(ERR) :g_mutex destroy.
```

```
sunrise@ubuntu:/app/cdev_demo/vio_capture$ ls
capture      raw_0.raw    raw_4.raw    raw_8.raw    yuv_2.yuv    yuv_6.yuv
capture.c     raw_1.raw    raw_5.raw    raw_9.raw    yuv_3.yuv    yuv_7.yuv
capture.o     raw_2.raw    raw_6.raw    yuv_0.yuv    yuv_4.yuv    yuv_8.yuv
Makefile      raw_3.raw    raw_7.raw    yuv_1.yuv    yuv_5.yuv    yuv_9.yuv
sunrise@ubuntu:/app/cdev_demo/vio_capture$
```