摄像头（同步）使用说明

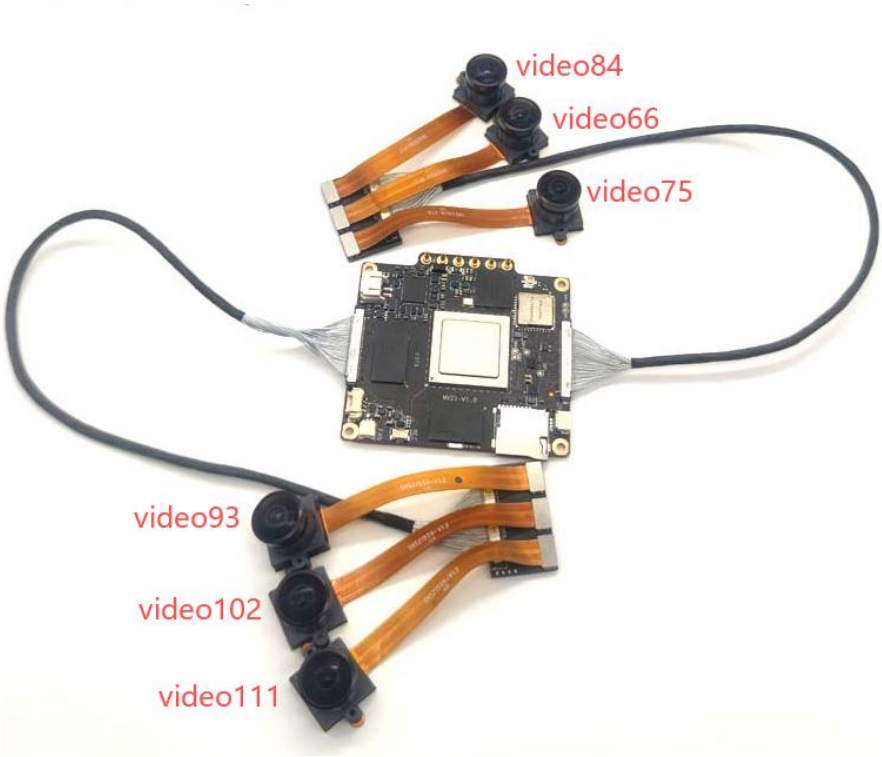
1. 摄像头输出节点

系统中共有六个同步摄像头，对应的输出节点如下：

摄像头	I2C	视频节点	控制节点
1	i2c3	/dev/video66	/dev/v4l-subdev2
2	i2c2	/dev/video75	/dev/v4l-subdev7
3	i2c4	/dev/video84	/dev/v4l-subdev12
4	i2c5	/dev/video93	/dev/v4l-subdev17
5	i2c7	/dev/video102	/dev/v4l-subdev22
6	i2c8	/dev/video111	/dev/v4l-subdev27

以上六个节点分别对应六路摄像头视频输出。

1.1 布局



---

2. 查看设置情况

可使用 v4l2-ctl 工具查看摄像头设置的情况。

2.1 测试命令示例

v4l2-ctl -d /dev/v4l-subdev2 -l

```
root@debian12:/# v4l2-ctl -d /dev/v4l-subdev2 -l

User Controls

                     exposure 0x00980911 (int)    : min=2 max=1216 step=1 default=1000 value=1000
                horizontal_flip 0x00980914 (bool)   : default=1 value=1
                  vertical_flip 0x00980915 (bool)   : default=1 value=1

Image Source Controls

              vertical_blanking 0x009e0901 (int)    : min=20 max=3020 step=1 default=20 value=20
            horizontal_blanking 0x009e0902 (int)    : min=4294965988 max=4294965988 step=1 default=42
94965988 value=-1308 flags=read-only
                 analogue_gain 0x009e0903 (int)    : min=32 max=512 step=1 default=350 value=350

Image Processing Controls

               link_frequency 0x009f0901 (intmenu): min=0 max=0 default=0 value=0 (450000000 0x1ad2
7480)
                    pixel_rate 0x009f0902 (int64)  : min=0 max=360000000 step=1 default=180000000 va
lue=180000000 flags=read-only
                  test_pattern 0x009f0903 (menu)   : min=0 max=15 default=0 value=0 (Disabled)
```

exposure 控制曝光，analogue_gain 曝光增益，启用 3A，曝光这些值可以不用管，horizontal_flip 控制水平翻转（0 正常，1 翻转），vertical_flip 控制垂直翻转（0 正常，1 翻转）

v4l2-ctl -d /dev/v4l-subdev2 --set-ctrl 'horizontal_flip=0'

控制摄像头水平翻转

---

3. 查看摄像头图像格式和分辨率

可使用 v4l2-ctl 工具查看图像格式和分辨率

3.1 获取摄像头支持的图像格式和分辨率

v4l2-ctl -d /dev/video66 --list-formats-ext

```
root@debian12:/# v4l2-ctl -d /dev/video66 --list-formats-ext
ioctl: VIDIOC_ENUM_FMT
        Type: Video Capture Multiplanar

        [0]: 'UYVY' (UYVY 4:2:2)
                Size: Stepwise 32x32 - 1920x1200 with step 8/8
        [1]: 'NV16' (Y/UV 4:2:2)
                Size: Stepwise 32x32 - 1920x1200 with step 8/8
        [2]: 'NV61' (Y/VU 4:2:2)
                Size: Stepwise 32x32 - 1920x1200 with step 8/8
        [3]: 'NV21' (Y/VU 4:2:0)
                Size: Stepwise 32x32 - 1920x1200 with step 8/8
        [4]: 'NV12' (Y/UV 4:2:0)
                Size: Stepwise 32x32 - 1920x1200 with step 8/8
        [5]: 'NM21' (Y/VU 4:2:0 (N-C))
                Size: Stepwise 32x32 - 1920x1200 with step 8/8
        [6]: 'NM12' (Y/UV 4:2:0 (N-C))
                Size: Stepwise 32x32 - 1920x1200 with step 8/8
```

3.2 获取当前摄像头图像格式和分辨率

v4l2-ctl -d /dev/video66 --get-fmt-video

```
root@debian12:/# v4l2-ctl -d /dev/video66 --get-fmt-video
Format Video Capture Multiplanar:
        Width/Height      : 1920/1200
        Pixel Format      : 'NV12' (Y/UV 4:2:0)
        Field             : None
        Number of planes  : 1
        Flags             :
        Colorspace        : sRGB
        Transfer Function : Rec. 709
        YCbCr/HSV Encoding: Rec. 709
        Quantization      : Full Range
        Plane 0           :
           Bytes per Line : 1920
           Size Image     : 3456000
```

4. 抓流测试方法

可使用 v4l2-ctl 工具对摄像头进行抓流测试。

4.1 测试命令示例

```
v4l2-ctl --verbose -d /dev/video66 \
  --set-fmt-video=width=1920,height=1200,pixelformat=UYVY \
  --stream-mmap=4
```

说明： -d /dev/video66：指定摄像头设备节点（其他摄像头请替换对应节点） - width / height：设置输出分辨率 - pixelformat=UYVY：设置像素格式 - --stream-mmap=4：使用 mmap 方式进行抓流，缓冲区数量为 4

5. 同步工作机制说明

本平台共接入 **六个摄像头**，其出流行为具有以下限制：

- **6*AR0234 摄像头都处于从模式，出流后需开启摄像头脉冲信号，要不然无法出流**

摄像头脉冲信号控制：

```
echo 30 > /dev/camera_sync_pwm  #输出 30fps
echo 60 > /dev/camera_sync_pwm  #输出 60fps
echo 0  > /dev/camera_sync_pwm   #停止输出
```

6. demo 程序

6.1 获取一组同步帧(C 语言)

代码位置 camera/demo/c/ar0234_sync.c

交叉编译指令

```
aarch64-none-linux-gnu-gcc -O2 -Wall -Wextra -pthread     -o
ar0234_sync ar0234_sync.c
```

示例：

```
./ar0234_sync 30 #以 30fps 获取同步帧
```

```
./ar0234_sync 60 #以 60fps 获取同步帧
```

```
root@debian12:/opt# ./ar0234_sync 30
=== Multi-Camera Real-time Synchronization System ===
Trigger: 30fps via /dev/camera_sync_pwm

[Camera 1] Format set: 1920x1200
[Camera 1] Buffer 0 mapped: 4608000 bytes
[Camera 1] Buffer 1 mapped: 4608000 bytes
[Camera 1] Buffer 2 mapped: 4608000 bytes
[Camera 1] Buffer 3 mapped: 4608000 bytes
[Camera 1] STREAMON done, armed and waiting for external trigger
[Camera 1] Flip control applied: horizontal=1, vertical=1
[Camera 2] Format set: 1920x1200
[Camera 2] Buffer 0 mapped: 4608000 bytes
[Camera 2] Buffer 1 mapped: 4608000 bytes
[Camera 2] Buffer 2 mapped: 4608000 bytes
[Camera 2] Buffer 3 mapped: 4608000 bytes
[Camera 2] STREAMON done, armed and waiting for external trigger
[Camera 2] Flip control applied: horizontal=1, vertical=1
[Camera 3] Format set: 1920x1200
[Camera 3] Buffer 0 mapped: 4608000 bytes
[Camera 3] Buffer 1 mapped: 4608000 bytes
[Camera 3] Buffer 2 mapped: 4608000 bytes
[Camera 3] Buffer 3 mapped: 4608000 bytes
[Camera 3] STREAMON done, armed and waiting for external trigger
[Camera 3] Flip control applied: horizontal=1, vertical=1
[Camera 4] Format set: 1920x1200
[Camera 4] Buffer 0 mapped: 4608000 bytes
[Camera 4] Buffer 1 mapped: 4608000 bytes
[Camera 4] Buffer 2 mapped: 4608000 bytes
[Camera 4] Buffer 3 mapped: 4608000 bytes
[Camera 4] STREAMON done, armed and waiting for external trigger
[Camera 4] Flip control applied: horizontal=1, vertical=1
[Camera 5] Format set: 1920x1200
[Camera 5] Buffer 0 mapped: 4608000 bytes
[Camera 5] Buffer 1 mapped: 4608000 bytes
[Camera 5] Buffer 2 mapped: 4608000 bytes
[Camera 5] Buffer 3 mapped: 4608000 bytes
[Camera 5] STREAMON done, armed and waiting for external trigger
[Camera 5] Flip control applied: horizontal=1, vertical=1
[Camera 6] Format set: 1920x1200
[Camera 6] Buffer 0 mapped: 4608000 bytes
[Camera 6] Buffer 1 mapped: 4608000 bytes
[Camera 6] Buffer 2 mapped: 4608000 bytes
[Camera 6] Buffer 3 mapped: 4608000 bytes
[Camera 6] STREAMON done, armed and waiting for external trigger
[Camera 6] Flip control applied: horizontal=1, vertical=1

=== All cameras armed (STREAMON complete) ===
[pwm] arming external trigger at 30fps via /dev/camera_sync_pwm

=== Starting Continuous Capture ===
Press 'q' to capture and save synchronized frames...
[Camera 1] Starting continuous capture...
[Camera 5] Starting continuous capture...
[Camera 2] Starting continuous capture...
[Camera 6] Starting continuous capture...
[Camera 4] Starting continuous capture...
[Camera 3] Starting continuous capture...
[Camera 4] Captured 30 frames
[Camera 1] Captured 30 frames
[Camera 2] Captured 30 frames
[Camera 5] Captured 30 frames
[Camera 3] Captured 30 frames
[Camera 6] Captured 30 frames
[Camera 4] Captured 60 frames
[Camera 1] Captured 60 frames
[Camera 2] Captured 60 frames
[Camera 5] Captured 60 frames
[Camera 3] Captured 60 frames
[Camera 6] Captured 60 frames
[Camera 4] Stopped. Total frames: 83
[Camera 1] Stopped. Total frames: 83
[Camera 2] Stopped. Total frames: 83
[Camera 5] Stopped. Total frames: 83
[Camera 6] Stopped. Total frames: 83
[Camera 3] Stopped. Total frames: 83
[pwm] disabling external trigger via /dev/camera_sync_pwm

=== Finding Best Synchronized Frame Set ===

Best synchronized frame set found:
Camera | Ring Index | Timestamp (ns)     | Total Frames Captured
-------|------------|--------------------|----------------------
  1    |       3    |     3510881834000 | 83
  2    |       3    |     3510881842000 | 83
  3    |       3    |     3510881846000 | 83
  4    |       3    |     3510881851000 | 83
  5    |       3    |     3510881856000 | 83
  6    |       3    |     3510881861000 | 83

Max timestamp delta: 27000 ns (0.027 ms)

=== Saving Synchronized Frames ===
Camera 1: Saved cam1_video66.yuv (4608000 bytes)
Camera 2: Saved cam2_video75.yuv (4608000 bytes)
Camera 3: Saved cam3_video84.yuv (4608000 bytes)
Camera 4: Saved cam4_video93.yuv (4608000 bytes)
Camera 5: Saved cam5_video102.yuv (4608000 bytes)
Camera 6: Saved cam6_video111.yuv (4608000 bytes)

Done!
```