



MPP sample 使用说明

版本号: 1.4

发布日期: 2022.03.20

版本历史

| 版本号 | 日期 | 制/修订人 | 内容描述 |
|-----|------------|---------|---|
| 1.0 | 2021.06.15 | AWA1771 | 初版，MPP sample 使用说明。 |
| 1.1 | 2021.08.12 | AWA1771 | 添加 Tina 和 Melis 上 MPP sample 的配置方法； 添加 Tina 和 Melis 各平台方案上 MPP sample 支持情况。 |
| 1.2 | 2021.08.28 | AWA1771 | 添加 sample_QPMAP 和 sample_venc_QPMAP。 |
| 1.3 | 2022.02.12 | AWA1771 | 移除 sample_QPMAP。 重命名 sample_venc_QPMAP 为 sample_vencQpMap。 添加以下 sample： 【视频】 sample_OnlineVenc、 sample_wbYuv、 sample_vencGdcZoom、 sample_takePicture； 【UVC 和 UAC】 sample_uac、 sample_usbcamera； 【AI demo】 sample_odet_demo、 sample_RegionDetect； 【其他】 sample_MotionDetect、 sample_PersonDetect、 sample_directIORead。 |
| 1.4 | 2022.03.20 | AWA1771 | 移除 sample_wbYuv。 更新各 sample 对测试目的、参数配置和预期结果的描述。 新增 sample 的编译和测试章节。 新增音视频编解码常用功能检查方法。 |

目 录

| | |
|-----------------------------------|-----------|
| 1 前言 | 1 |
| 1.1 概述 | 1 |
| 1.2 读者对象 | 1 |
| 1.3 约定 | 1 |
| 1.3.1 符号约定 | 1 |
| 2 简介 | 2 |
| 3 配置 | 3 |
| 3.1 Tina 上 MPP sample 的配置方法 | 3 |
| 3.2 Melis 上 MPP sample 的配置方法 | 3 |
| 4 编译 | 4 |
| 5 测试 | 5 |
| 5.1 在板端用串口测试 | 5 |
| 5.2 在 PC 端用 adb shell 测试 | 5 |
| 5.3 测试指令 | 5 |
| 6 类别 | 7 |
| 7 状态 | 11 |
| 7.1 Tina 各平台方案上 MPP sample 支持情况 | 11 |
| 7.2 Melis 各平台方案上 MPP sample 支持情况 | 13 |
| 8 正文 | 16 |
| 8.1 视频 | 16 |
| 8.1.1 sample_driverVipp | 16 |
| 8.1.2 sample_virvi | 17 |
| 8.1.3 sample_virvi2vo | 19 |
| 8.1.4 sample_virvi2vo_zoom | 21 |
| 8.1.5 sample_vi_reset | 22 |
| 8.1.6 sample_isposd | 23 |
| 8.1.7 sample_vin_isp_test | 24 |
| 8.1.8 sample_region | 25 |
| 8.1.9 sample_venc | 28 |
| 8.1.10 sample_venc2muxer | 29 |
| 8.1.11 sample_virvi2venc | 31 |
| 8.1.12 sample_timelapse | 34 |
| 8.1.13 sample_virvi2venc2muxer | 35 |
| 8.1.14 sample_multi_vi2venc2muxer | 38 |
| 8.1.15 sample_rtsp | 40 |
| 8.1.16 sample_CodecParallel | 43 |

| | | |
|--------|------------------------------|----|
| 8.1.17 | sample_vdec | 45 |
| 8.1.18 | sample_demux2vdec | 46 |
| 8.1.19 | sample_demux2vdec_saveFrame | 47 |
| 8.1.20 | sample_demux2vdec2vo | 48 |
| 8.1.21 | sample_vencQpMap | 49 |
| 8.1.22 | sample_OnlineVenc | 50 |
| 8.1.23 | sample_vencGdcZoom | 52 |
| 8.1.24 | sample_takePicture | 55 |
| 8.1.25 | sample_recorder | 57 |
| 8.1.26 | sample_vencRecreate | 58 |
| 8.2 | 音频 | 61 |
| 8.2.1 | sample_ai | 61 |
| 8.2.2 | sample_ao | 62 |
| 8.2.3 | sample_aoSync | 64 |
| 8.2.4 | sample_ao_resample_mixer | 65 |
| 8.2.5 | sample_ao2ai_aec | 66 |
| 8.2.6 | sample_ao2ai_aec_rate_mixer | 68 |
| 8.2.7 | sample_aec | 69 |
| 8.2.8 | sample_aenc | 71 |
| 8.2.9 | sample_ai2aenc | 72 |
| 8.2.10 | sample_ai2aenc2muxer | 73 |
| 8.2.11 | sample_select | 74 |
| 8.2.12 | sample_adec | 75 |
| 8.2.13 | sample_adec2ao | 76 |
| 8.2.14 | sample_demux2adec | 77 |
| 8.2.15 | sample_demux2adec2ao | 78 |
| 8.3 | ISE 和 EIS | 79 |
| 8.3.1 | sample_fish | 79 |
| 8.3.2 | sample_virvi2fish2venc | 80 |
| 8.3.3 | sample_virvi2fish2vo | 82 |
| 8.3.4 | sample_virvi2eis2venc | 83 |
| 8.3.5 | sample_ise_dzoom | 85 |
| 8.3.6 | sample_gdc_dzoom | 86 |
| 8.4 | 视频显示 | 88 |
| 8.4.1 | sample_vo | 88 |
| 8.4.2 | sample_UILayer | 89 |
| 8.5 | G2D | 90 |
| 8.5.1 | sample_g2d | 90 |
| 8.5.2 | sample_vi_g2d | 91 |
| 8.6 | CE | 92 |
| 8.6.1 | sample_twinchn_virvi2venc2ce | 92 |
| 8.6.2 | sample_virvi2venc2ce | 94 |
| 8.7 | UVC 和 UAC | 95 |

| | | |
|----------|-------------------------------|------------|
| 8.7.1 | sample_uvc2vdec_vo | 95 |
| 8.7.2 | sample_uvc2vdenc2vo | 97 |
| 8.7.3 | sample_uvc2vo | 98 |
| 8.7.4 | sample_uvc_vo | 99 |
| 8.7.5 | sample_uvcout | 100 |
| 8.7.6 | sample_uac | 102 |
| 8.7.7 | sample_usbcamera | 104 |
| 8.8 | 多媒体文件 | 105 |
| 8.8.1 | sample_demux | 105 |
| 8.8.2 | sample_file_repair | 107 |
| 8.9 | AI demo | 107 |
| 8.9.1 | sample_odet_demo | 107 |
| 8.9.2 | sample_RegionDetect | 109 |
| 8.10 | 其他 | 110 |
| 8.10.1 | sample_glog | 110 |
| 8.10.2 | sample_hello | 111 |
| 8.10.3 | sample_pthread_cancel | 112 |
| 8.10.4 | sample_motor | 113 |
| 8.10.5 | sample_sound_controler | 113 |
| 8.10.6 | sample_ai_Demo_0.9.2 | 115 |
| 8.10.7 | sample_ai_Demo_0.9.6 | 115 |
| 8.10.8 | sample_thumb | 115 |
| 8.10.9 | sample_nna | 116 |
| 8.10.10 | sample_MotionDetect | 117 |
| 8.10.11 | sample_PersonDetect | 118 |
| 8.10.12 | sample_directIORead | 119 |
| 9 | FAQ | 121 |
| 9.1 | 某些方案上开机后执行某些 MPP sample 会运行失败 | 121 |
| 9.2 | MPP sample 测试时 SD 卡识别异常 | 121 |
| 9.3 | 视频编解码常用功能检查方法 | 121 |
| 9.3.1 | 常用检测工具介绍 | 121 |
| 9.3.1.1 | 码流播放软件 | 121 |
| 9.3.1.2 | 码流分析软件 | 122 |
| 9.3.1.3 | YUV 文件分析软件 | 124 |
| 9.3.1.4 | H264 码流分析工具 | 125 |
| 9.3.2 | 典型的视频编解码功能检测示例 | 126 |
| 9.3.2.1 | 编码格式 | 126 |
| 9.3.2.2 | 彩转灰 | 126 |
| 9.3.2.3 | 旋转编码 | 126 |
| 9.3.2.4 | P 帧帧内刷新 | 128 |
| 9.3.2.5 | 视频编码 OSD | 130 |
| 9.4 | 音频编解码常用功能检查方法 | 132 |

9.4.1 常用检测工具介绍 132



插图

| | |
|--|----|
| 8-1 MPP_sample 测试通路-sample_driverVipp | 16 |
| 8-2 MPP_sample 测试通路-sample_virvi | 17 |
| 8-3 MPP_sample 测试通路-sample_virvi2vo | 19 |
| 8-4 MPP_sample 测试通路-sample_virvi2vo_zoom | 21 |
| 8-5 MPP_sample 测试通路-sample_vi_reset | 22 |
| 8-6 MPP_sample 测试通路-sample_isposd | 23 |
| 8-7 MPP_sample 测试通路-sample_vin_isp_test | 25 |
| 8-8 MPP_sample 测试通路-sample_region | 26 |
| 8-9 MPP_sample 测试通路-sample_venc | 28 |
| 8-10 MPP_sample 测试通路-sample_venc2muxer | 29 |
| 8-11 MPP_sample 测试通路-sample_virvi2venc | 31 |
| 8-12 MPP_sample 测试通路-sample_timelapse | 34 |
| 8-13 MPP_sample 测试通路-sample_virvi2venc2muxer | 35 |
| 8-14 MPP_sample 测试通路-sample_multi_vi2venc2muxer | 38 |
| 8-15 MPP_sample 测试通路-sample_rtsp | 41 |
| 8-16 MPP_sample 测试通路-sample_CodecParallel | 43 |
| 8-17 MPP_sample 测试通路-sample_vdec | 45 |
| 8-18 MPP_sample 测试通路-sample_demux2vdec | 46 |
| 8-19 MPP_sample 测试通路-sample_demux2vdec_saveFrame | 47 |
| 8-20 MPP_sample 测试通路-sample_demux2vdec2vo | 48 |
| 8-21 MPP_sample 测试通路-sample_OnlineVenc | 50 |
| 8-22 MPP_sample 测试通路-sample_vencGdcZoom | 53 |
| 8-23 MPP_sample 测试通路-sample_takePicture | 56 |
| 8-24 MPP_sample 测试通路-sample_recorder | 57 |
| 8-25 MPP_sample 测试通路-sample_vencRecreate | 59 |
| 8-26 MPP_sample 测试通路-sample_ai | 62 |
| 8-27 MPP_sample 测试通路-sample_ao | 63 |
| 8-28 MPP_sample 测试通路-sample_aoSync | 64 |
| 8-29 MPP_sample 测试通路-sample_ao_resample_mixer | 65 |
| 8-30 MPP_sample 测试通路-sample_ao2ai_aec | 67 |
| 8-31 MPP_sample 测试通路-sample_ao2ai_aec_rate_mixer | 68 |
| 8-32 MPP_sample 测试通路-sample_aec | 70 |
| 8-33 MPP_sample 测试通路-sample_aenc | 71 |
| 8-34 MPP_sample 测试通路-sample_ai2aenc | 72 |
| 8-35 MPP_sample 测试通路-sample_ai2aenc2muxer | 73 |
| 8-36 MPP_sample 测试通路-sample_select | 74 |
| 8-37 MPP_sample 测试通路-sample_adec | 75 |
| 8-38 MPP_sample 测试通路-sample_adec2ao | 76 |
| 8-39 MPP_sample 测试通路-sample_demux2adec | 77 |
| 8-40 MPP_sample 测试通路-sample_demux2adec2ao | 78 |
| 8-41 MPP_sample 测试通路-sample_virvi2fish2venc | 81 |

| | | |
|------|--|-----|
| 8-42 | MPP_sample 测试通路-sample_virvi2fish2vo | 82 |
| 8-43 | MPP_sample 测试通路-sample_virvi2eis2venc | 84 |
| 8-44 | MPP_sample 测试通路-sample_vo | 88 |
| 8-45 | MPP_sample 测试通路-sample_UIlayer | 89 |
| 8-46 | MPP_sample 测试通路-sample_vi_g2d | 91 |
| 8-47 | MPP_sample 测试通路-sample_twinchn_virvi2venc2ce | 93 |
| 8-48 | MPP_sample 测试通路-sample_virvi2venc2ce | 94 |
| 8-49 | MPP_sample 测试通路-sample_uvc2vdec_vo | 96 |
| 8-50 | MPP_sample 测试通路-sample_uvc2vdenc2vo | 97 |
| 8-51 | MPP_sample 测试通路-sample_uvc2vo | 98 |
| 8-52 | MPP_sample 测试通路-sample_uvc_vo | 99 |
| 8-53 | MPP_sample 测试通路-sample_uvcout | 100 |
| 8-54 | MPP_sample 测试通路-sample_demux | 106 |
| 8-55 | MPP_sample 测试通路-sample_sound_controler | 114 |
| 8-56 | MPP_sample 测试通路-sample_thumb | 115 |
| 9-1 | VLC 播放码流 | 122 |
| 9-2 | MediaInfo 分析码流 | 123 |
| 9-3 | YUView 软件配置 ffmpeg 动态库 | 125 |
| 9-4 | 彩转灰效果 | 126 |
| 9-5 | 旋转-1 | 127 |
| 9-6 | 旋转-2 | 128 |
| 9-7 | H264-开启 P 帧帧内刷新 | 129 |
| 9-8 | H264-关闭 P 帧帧内刷新 | 129 |
| 9-9 | H265-开启 P 帧帧内刷新 | 130 |
| 9-10 | H265-关闭 P 帧帧内刷新 | 130 |
| 9-11 | 编码 OSD-1 | 131 |
| 9-12 | 编码 OSD-2 | 131 |
| 9-13 | OSD 坐标和大小示意图 | 132 |
| 9-14 | 通道波形对比图 | 133 |

1 前言

1.1 概述

整理 MPP sample 使用说明文档的目的是：**使 MPP sample 更好用。**

1.2 读者对象

本文档（本指南）主要适用于以下人员：

- 技术支持工程师
- 软件开发工程师

1.3 约定

1.3.1 符号约定

本文中可能出现的符号如下：

警告

警告

技巧

1. 技巧
2. 小常识

说明

说明

2 简介

MPP sample 一般存放在 MPP Middleware 的 sample 目录下。此外，MPP Framework 的 demo 目录下也有一些 sample。

本文档主要介绍 MPP Middleware 各 sample 的基本使用方法：配置、编译、测试以及 sample 类别、各平台方案上的支持情况和测试方法等。

文末 FAQ 部分对音视频编解码功能的测试方法和测试工具的使用做了详细介绍。



3 配置

本章节主要介绍 MPP sample 的配置方法。目前 Tina 和 Melis 上存在差异。

3.1 Tina 上 MPP sample 的配置方法

Tina V853、V833 和 V536 方案上，MPP sample 支持从 menuconfig 中配置。选中 [*] select mpp_sample 之后，当前支持测试的 MPP sample 就会显示出来，默认都是未选中的状态。此时，可以勾选想要测试的 sample，然后保存配置，重新编译 MPP 即可。

```
$ make menuconfig
Allwinner --->
  eyesee-mpp --->
    [*] select mpp sample
```

Tina 其他方案（如：V316 等）上，目前 MPP sample 不支持从 menuconfig 中配置。若想开启或关闭 sample，只能修改 tina.mk 文件，然后重新编译 MPP。

⚠ 警告

1. 各 sample 对组件的依赖关系详情，请参考文件 tina\package\allwinner\eyesee-mpp\middleware\Config.ino。
2. 因为每个 MPP sample 编译时，依赖的 MPP 组件不一样，所以，只有当支持的 MPP 组件打开后，相关的 MPP sample 才会显示出来。否则，不可见。
3. 由于在编译 MPP 基础库 libaw_mpp.a 或 libmedia_mpp.so 时，mpi region 是默认开启的，且 mpi region 依赖 mpi_vi 和 mpi_venc，所以 **mpi_vi 和 mpi_venc 是运行 MPP sample 的基础组件，需要保持常开。**

3.2 Melis 上 MPP sample 的配置方法

Melis 上各方案，目前 MPP sample 不支持从 menuconfig 中配置。若想开启或关闭 sample，只能修改 Makefile 文件，然后重新编译 MPP。

4 编译

本章节主要介绍 MPP sample 的编译方法。

以下以 tina 为例。

编译命令

```
cleanmpp && mkmpp
```

编译后 MPP sample 测试程序和配置文件存放位置

有两个位置：

(1) 每个 MPP sample 的源码目录下

```
tina\external\eyeseer-mpp\middleware\sun8iw21\sample\sample_xxx\
```

(2) 统一存放到bin目录下

```
tina\external\eyeseer-mpp\middleware\sun8iw21\sample\bin\
```

PS：由于 MPP sample 测试程序个数较多，且每个测试程序大小约 10M 左右，故不方便打包到文件系统中，测试时需要接 SD 卡，同时把测试程序和配置文件放到 SD 卡中进行测试。

5 测试

本章节主要介绍 MPP sample 的测试方法。

5.1 在板端用串口测试

1. 将测试程序和配置拷贝到 SD 卡，然后将 SD 卡接到板端。
2. 将 SD 卡 mount 到/mnt/extsd 目录，检查测试程序和配置是否在路径/mnt/extsd 下，然后准备测试。

```
mkdir -p /mnt/extsd  
mount -t vfat /dev/mmcblk0p1 /mnt/extsd
```

5.2 在 PC 端用 adb shell 测试

前提，需要配置 sdk 支持 adb（一般默认都支持）。

1. 在 PC 端启动 Windows PowerShell 终端，然后切换到测试程序和配置所在路径。
2. 在 Windows PowerShell 终端，使用 adb push 命令将测试程序和配置推送到板端。

```
> adb push .\sample_virvi2venc\sample_virvi2venc /mnt/extsd/  
.\sample_virvi2venc\sample_virvi2venc: 1 file pushed. 3.4 MB/s (8699676 bytes in 2.442s)  
>  
> adb push .\sample_virvi2venc\sample_virvi2venc.conf /mnt/extsd/  
.\sample_virvi2venc\sample_virvi2venc.conf: 1 file pushed. 0.1 MB/s (693 bytes in 0.005s)
```

3. 在 Windows PowerShell 终端，使用 adb shell 命令登录到板端，然后准备测试。

5.3 测试指令

以下分别以 sample_driverVipp 和 sample_virvi2venc 为例介绍没有配置文件和有配置文件两种情况下，MPP sample 的测试指令。

【Tina】

```
# cd /mnt/extsd/  
# ./sample_driverVipp # 需要逐个指定参数。  
或者一次性指定所有测试参数  
# ./sample_driverVipp 0 1920 1080 8 60 0 10 60 /mnt/extsd/
```

【Melis】

Melis上该sample的位置：
melis\source\ekernel\subsys\avframework\v4l2\drivers\media\platform\sunxi-vin\
sample_driverVipp
测试方法：
sample_driverVipp 0 1920 1080 8 60 0 10 60 /mnt/E/

【Tina】

```
# cd /mnt/extsd/  
# ./sample_virvi2venc -path ./sample_virvi2venc.conf
```

【Melis】

```
# sample_virvi2venc -path /mnt/E/sample_virvi2venc.conf
```



6 类别

本章节主要介绍 MPP sample 的分类情况。

参考 MPP Middleware sample 目录下的文档 SampleManual.md ，目前主要分成以下几类：

- 视频
- 音频
- ISE 和 EIS
- 视频显示
- G2D
- CE
- UVC 和 UAC
- 多媒体文件
- AI demo
- 其他

【视频】

- **sample_driverVipp** 演示直接调用 linux 内核驱动获取 frame
- **sample_virvi** 视频采集
- **sample_virvi2vo** 演示视频采集预览
- **sample_virvi2vo_zoom** 演示缩放
- **sample_vi_reset** 演示 mpi_vi 组件的 reset 流程
- **sample_isposd** 测试 isp 相关
- **sample_vin_isp_test** 通过调用 isp 接口抓图的 demo
- **sample_region** 测试 osd
- **sample_venc** 视频编码
- **sample_venc2muxer** 演示视频编码和封装 mp4
- **sample_virvi2venc** 演示采集到编码
- **sample_timelapse** 演示缩时录影
- **sample_virvi2venc2muxer** 演示采集到编码到封装 mp4
- **sample_multi_vi2venc2muxer** 演示多路编码
- **sample_rtsp** 演示视频编码后的 rtsp 传输
- **sample_CodecParallel** 演示同编同解
- **sample_vdec** 视频解码
- **sample_demux2vdec** 演示解封装和解码

- `sample_demux2vdec_saveFrame` 从原始文件中分离出视频数据帧并解码生成 yuv 文件
- `sample_demux2vdec2vo` 演示解码显示
- `sample_vencQpMap` 演示视频编码 QPMAP 模式
- `sample_OnlineVenc` 在线编码
- `sample_vencGdcZoom` 编码 GDC 数字变焦
- `sample_takePicture` 演示单拍和连拍
- `sample_recorder` 演示四路录制编码封装或者预览显示
- `sample_vencRecreate` 演示动态配置编码格式、帧率、码率等功能

【音频】

- `sample_ai` 演示音频采集
- `sample_ao` 演示音频输出
- `sample_aoSync` 演示采用同步的方式 send pcm frame。而 `sample_ao` 是采用异步的方式
- `sample_ao_resample_mixer` 演示读取 pcm 数据，然后播放声音，从耳机口输出声音
- `sample_ao2ai_aec` 回声消除，初版
- `sample_ao2ai_aec_rate_mixer` 回声消除，初版基础上测试打开音频输出重采样和混音功能后，回声消除是否正常
- `sample_aec` 回声消除，修改版
- `sample_aenc` 音频编码
- `sample_ai2aenc` 音频采集和编码
- `sample_ai2aenc2muxer` 音频采集、编码和封装
- `sample_select` 演示多路音频编码，用 `select()` 方式获取编码码流
- `sample_adec` 音频解码
- `sample_adec2ao` 音频解码输出
- `sample_demux2adec` 从原始文件中分离出音频数据帧并解码
- `sample_demux2adec2ao` 从原始文件中分离出音频数据帧并解码输出音频

【ISE 和 EIS】

- `sample_fish` 演示单目鱼眼功能
- `sample_virvi2fish2venc` 演示鱼眼图像畸形校正后做编码
- `sample_virvi2fish2vo` 演示鱼眼图像畸形校正后预览
- `sample_virvi2eis2venc` 演示防抖功能
- `sample_ise_dzoom` 演示 ise 缩放功能
- `sample_gdc_dzoom` 演示 gdc 缩放功能

【视频显示】

- `sample_vo` 演示视频 YUV 预览
- `sample_UILayer` 验证 UILayer 的格式

【G2D】

- `sample_g2d` 演示直接用 G2D 处理 YUV 文件做透明叠加、旋转和镜像、缩放等
- `sample_vi_g2d` 演示从 VI 获取视频帧，用 G2D 做旋转、裁剪、缩放等处理后送 VO 显示

【CE】

- `sample_twinchn_virvi2venc2ce` 两路 ce 加解密测试
- `sample_virvi2venc2ce` 单路 ce 加解密测试

【UVC 和 UAC】

- `sample_uvc2vdec_vo` 做主，mjpeg 解码显示
- `sample_uvc2venc2vo` 做主，mjpeg 解码显示
- `sample_uvc2vo` 做主，yuv 显示
- `sample_uvc_vo` 做主，yuv 显示
- `sample_uvcout` 做从，mjpeg 编码输出
- `sample_uac` 演示 uac 测试
- `sample_usbcamera` 测试 UVC、UAC 复合设备

【多媒体文件】

- `sample_demux` 解复用 mp4 文件
- `sample_file_repair` 修复 mp4 文件

【AI demo】

- `sample_odet_demo` 目标检测演示示例
- `sample_RegionDetect` 演示区域检测

【其他】

- `sample_glog` 演示 glog 库的用法
- `sample_hello` 演示 MPP helloworld 程序
- `sample_pthread_cancel` 测试 pthread_cancel()
- `sample_motor` 电机测试
- `sample_sound_controler` 语音识别
- `sample_ai_Demo_0.9.2` 智能算子 demo
- `sample_ai_Demo_0.9.6` 智能算子 demo

- `sample_thumb` 缩略图
- `sample_nna` 人形检测和人脸检测
- `sample_MotionDetect` 移动侦测
- `sample_PersonDetect` 人形检测
- `sample_directIORead` 演示使用 directIO 方式读文件



7 状态

本章节主要介绍当前各 MPP sample 的状态，即：分别在 Tina 和 Melis 各平台方案上的支持情况。

7.1 Tina 各平台方案上 MPP sample 支持情况

说明

标注“Y”则表示该 **sample** 支持在该平台方案上测试，未标注则表示不支持。

| MPP sample | V853 | V833 | V536 | V533 | V316 |
|-----------------------------|------|------|------|------|------|
| 总数 | 61 | 54 | 39 | 45 | 33 |
| 视频 | 26 | 19 | 13 | 17 | 11 |
| sample_driverVipp | Y | Y | | Y | |
| sample_virvi | Y | Y | Y | Y | Y |
| sample_virvi2vo | Y | Y | Y | Y | Y |
| sample_virvi2vo_zoom | Y | Y | | | |
| sample_vi_reset | Y | Y | | Y | |
| sample_isposd | Y | Y | | Y | |
| sample_vin_isp_test | Y | Y | | Y | |
| sample_region | Y | Y | Y | Y | Y |
| sample_venc | Y | Y | Y | Y | Y |
| sample_venc2muxer | Y | Y | Y | Y | Y |
| sample_virvi2venc | Y | Y | Y | Y | Y |
| sample_timelapse | Y | Y | Y | Y | |
| sample_virvi2venc2muxer | Y | Y | Y | Y | Y |
| sample_multi_vi2venc2muxer | Y | Y | | | |
| sample_rtsp | Y | Y | Y | Y | |
| sample_CodecParallel | Y | Y | | | |
| sample_vdec | Y | Y | Y | Y | Y |
| sample_demux2vdec | Y | Y | Y | Y | Y |
| sample_demux2vdec_saveFrame | Y | Y | Y | Y | Y |
| sample_demux2vdec2vo | Y | Y | Y | Y | Y |
| sample_vencQpMap | Y | | | | |
| sample_OnlineVenc | Y | | | | |
| sample_vencGdcZoom | Y | | | | |

| MPP sample | V853 | V833 | V536 | V533 | V316 |
|------------------------------|------|------|------|------|------|
| sample_takePicture | Y | | | | |
| sample_recorder | Y | | | | |
| 音频 | 14 | 14 | 9 | 10 | 9 |
| sample_ai | Y | Y | Y | Y | Y |
| sample_ao | Y | Y | Y | Y | Y |
| sample_aoSync | Y | Y | | | |
| sample_ao_resample_mixer | Y | Y | | | |
| sample_ao2ai_aec | Y | Y | | Y | |
| sample_ao2ai_aec_rate_mixer | Y | Y | | | |
| sample_aec | Y | Y | | | |
| sample_aenc | Y | Y | Y | Y | Y |
| sample_ai2aenc | Y | Y | Y | Y | Y |
| sample_ai2aenc2muxer | Y | Y | Y | Y | Y |
| sample_select | Y | Y | Y | Y | Y |
| sample_adeac | Y | Y | Y | Y | Y |
| sample_demux2adeac | Y | Y | Y | Y | Y |
| sample_demux2adeac2ao | Y | Y | Y | Y | Y |
| ISE 和 EIS | 0 | 4 | 6 | 4 | 5 |
| sample_fish | | Y | Y | Y | Y |
| sample_virvi2fish2venc | | Y | Y | Y | Y |
| sample_virvi2fish2vo | | Y | Y | Y | Y |
| sample_virvi2eis2venc | | Y | Y | Y | Y |
| sample_ise_dzoom | | | Y | | |
| sample_gdc_dzoom | | | Y | | Y |
| 视频显示 | 2 | 2 | 1 | 2 | 1 |
| sample_vo | Y | Y | Y | Y | Y |
| sample_UILayer | Y | Y | | Y | |
| G2D | 2 | 2 | 1 | 1 | 1 |
| sample_g2d | Y | Y | Y | Y | Y |
| sample_vi_g2d | Y | Y | | | |
| CE | 2 | 2 | 0 | 2 | 0 |
| sample_twinchn_virvi2venc2ce | Y | Y | | Y | |
| sample_virvi2venc2ce | Y | Y | | Y | |
| UVC 和 UAC | 7 | 5 | 5 | 5 | 5 |
| sample_uvc2vdec_vo | Y | Y | Y | Y | Y |
| sample_uvc2vdenc2vo | Y | Y | Y | Y | Y |
| sample_uvc2vo | Y | Y | Y | Y | Y |
| sample_uvc_vo | Y | Y | Y | Y | Y |
| sample_uvcout | Y | Y | Y | Y | Y |
| sample_uac | Y | | | | |
| sample_usbcamera | Y | | | | |

| MPP sample | V853 | V833 | V536 | V533 | V316 |
|------------------------|------|------|------|------|------|
| 多媒体文件 | 2 | 2 | 2 | 1 | 1 |
| sample_demux | Y | Y | Y | Y | Y |
| sample_file_repair | Y | Y | Y | | |
| AI demo | 2 | 0 | 0 | 0 | 0 |
| sample_odet_demo | Y | | | | |
| sample_RegionDetect | Y | | | | |
| 其他 | 6 | 3 | 2 | 4 | 0 |
| sample_glog | Y | Y | Y | Y | |
| sample_hello | Y | Y | Y | Y | |
| sample_pthread_cancel | Y | Y | | | |
| sample_motor | | | | Y | |
| sample_sound_controler | | | | Y | |
| sample_ai_Demo_0.9.2 | | | | | |
| sample_ai_Demo_0.9.6 | | | | | |
| sample_thumb | | | | | |
| sample_nna | | | | | |
| sample_MotionDetect | Y | | | | |
| sample_PersonDetect | Y | | | | |
| sample_directIORead | Y | | | | |

7.2 Melis 各平台方案上 MPP sample 支持情况

说明

标注“Y”则表示该 **sample** 支持在该平台方案上测试，未标注则表示不支持。

| MPP sample | V459 |
|----------------------|------|
| 总数 | 21 |
| 视频 | 9 |
| sample_driverVipp | Y |
| sample_virvi | Y |
| sample_virvi2vo | Y |
| sample_virvi2vo_zoom | Y |
| sample_vi_reset | |
| sample_isposd | |
| sample_vin_isp_test | |
| sample_region | Y |
| sample_venc | |
| sample_venc2muxer | |
| sample_virvi2venc | Y |

| | |
|------------------------------|------|
| MPP sample | V459 |
| sample_timelapse | |
| sample_virvi2venc2muxer | Y |
| sample_multi_vi2venc2muxer | Y |
| sample_rtsp | |
| sample_CodecParallel | |
| sample_vdec | |
| sample_demux2vdec | |
| sample_demux2vdec_saveFrame | |
| sample_demux2vdec2vo | Y |
| 音频 | 9 |
| sample_ai | Y |
| sample_ao | Y |
| sample_ao_resample_mixer | |
| sample_ao2ai_aec | Y |
| sample_ao2ai_aec_rate_mixer | |
| sample_aec | Y |
| sample_aenc | Y |
| sample_ai2aenc | |
| sample_ai2aenc2muxer | Y |
| sample_select | |
| sample_adec | Y |
| sample_demux2adec | |
| sample_demux2adec2ao | Y |
| sample_aoSync | Y |
| ISE 和 EIS | 1 |
| sample_fish | |
| sample_virvi2fish2venc | |
| sample_virvi2fish2vo | Y |
| sample_virvi2eis2venc | |
| sample_ise_dzoom | |
| sample_gdc_dzoom | |
| 视频显示 | 0 |
| sample_vo | |
| sample_UILayer | |
| G2D | 1 |
| sample_g2d | |
| sample_vi_g2d | Y |
| CE | 0 |
| sample_twinchn_virvi2venc2ce | |
| sample_virvi2venc2ce | |
| UVC | 0 |

| | |
|------------------------|------|
| MPP sample | V459 |
| sample_uvc2vdec_vo | |
| sample_uvc2vdenc2vo | |
| sample_uvc2vo | |
| sample_uvc_vo | |
| sample_uvcout | |
| 多媒体文件 | 0 |
| sample_demux | |
| sample_file_repair | |
| 其他 | 1 |
| sample_glog | |
| sample_hello | |
| sample_pthread_cancel | |
| sample_motor | |
| sample_sound_controler | |
| sample_ai_Demo_0.9.2 | |
| sample_ai_Demo_0.9.6 | |
| sample_thumb | Y |
| sample_nna | |

8 正文

本章节具体介绍 MPP 各 sample 的测试方法。主要包括：测试目的、组件依赖、测试通路、源文件、目标文件、参数配置、测试指令、退出测试、预期结果等。

8.1 视频

8.1.1 sample_driverVipp

测试目的：

演示直接调用vin驱动接口获取视频帧。
该sample没有使用任何mpp组件。

组件依赖：

无

测试通路：

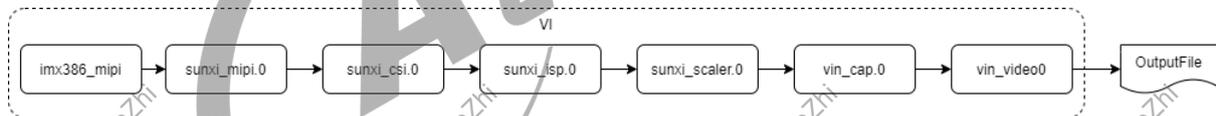


图 8-1: MPP_sample 测试通路-sample_driverVipp

源文件：

无

目标文件：

每隔若干帧保存一帧到指定的output目录。

参数配置：

video_device: 指定vipp设备节点，通常取值0、1、2、3（对于V853，通常取值0、4、8、12）。

capture_width: 指定camera采集的图像宽度。

capture_height: 指定camera采集的图像高度。

pixel_format: 指定camera采集的图像格式。

fps: 指定camera采集的帧率。

test_frame_count: 指定测试采集的frame总数，0表示无限。

store_count: 指定保存的图像数量。

store_interval: 指定保存图像的周期，即每n帧图像保存1帧。

frame_saving_path: 指定保存图像的目录, 该目录要确保存在。

测试指令:

【Tina】

```
# cd /mnt/extsd/
# ./sample_driverVipp # 需要逐个指定参数。
或者一次性指定所有测试参数
# ./sample_driverVipp 0 1920 1080 8 60 0 10 60 /mnt/extsd/
```

【Melis】

Melis上该sample的位置:

```
melis\source\kernel\subsys\avframework\v4l2\drivers\media\platform\sunxi-vin\
sample_driverVipp
```

测试方法:

```
sample_driverVipp 0 1920 1080 8 60 0 10 60 /mnt/E/
```

退出测试:

不自动退出测试, 需要按 “ctrl + c” 退出测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 使用YUVview 软件分析生成的YUV文件正常。

8.1.2 sample_virvi

测试目的:

演示采集图像帧, 最多可支持4路vipp采集。
该sample测试mpi_vi组件。

组件依赖:

mpi_vi

测试通路:

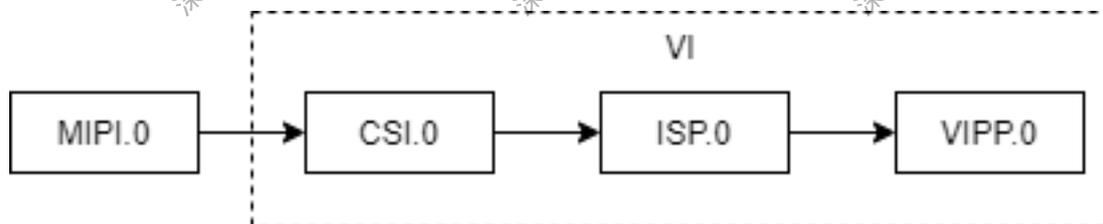


图 8-2: MPP_sample 测试通路-sample_virvi

源文件:

无

目标文件：

1. 连续YUV文件
/mnt/extsd/test.yuv
2. 单帧YUV文件
存放在指定路径/mnt/extsd 下面。

参数配置：

最多支持4路vipp采集，这4路的参数可单独配置。

第0路vipp的参数配置

dev_num_0: 指定vipp设备ID, 通常取值0, 表示vipp0 (对于V853, 通常也取值0, 表示vipp0)。
isp_dev_num_0: 指定isp设备ID, 通常取值0, 表示isp0。
pic_width_0: 指定vipp采集图像的宽度。
pic_height_0: 指定vipp采集图像的高度。
frame_rate_0: 指定vipp采集图像的帧率。
pic_format_0: 指定vipp采集图像的像素格式。
color_space_0: 指定vipp采集图像的颜色空间格式。
enable_wdr_mode_0: 指定vipp采集图像是否开启WDR。
drop_frm_num_0: 指定vipp开始前丢弃的帧数。

第1路vipp的参数配置

dev_num_1: 指定vipp设备ID, 通常取值1, 表示vipp1 (对于V853, 通常取值4, 表示vipp4)。
isp_dev_num_1: 指定isp设备ID, 通常取值0, 表示isp0。
pic_width_1: 指定vipp采集图像的宽度。
pic_height_1: 指定vipp采集图像的高度。
frame_rate_1: 指定vipp采集图像的帧率。
pic_format_1: 指定vipp采集图像的像素格式。
color_space_1: 指定vipp采集图像的颜色空间格式。
enable_wdr_mode_1: 指定vipp采集图像是否开启WDR。
drop_frm_num_1: 指定vipp开始前丢弃的帧数。

第2路vipp的参数配置

dev_num_2: 指定vipp设备ID, 通常取值2, 表示vipp2 (对于V853, 通常取值8, 表示vipp8)。
isp_dev_num_2: 指定isp设备ID, 通常取值0, 表示isp0。
pic_width_2: 指定vipp采集图像的宽度。
pic_height_2: 指定vipp采集图像的高度。
frame_rate_2: 指定vipp采集图像的帧率。
pic_format_2: 指定vipp采集图像的像素格式。
color_space_2: 指定vipp采集图像的颜色空间格式。
enable_wdr_mode_2: 指定vipp采集图像是否开启WDR。
drop_frm_num_2: 指定vipp开始前丢弃的帧数。

第3路vipp的参数配置

dev_num_3: 指定vipp设备ID, 通常取值3, 表示vipp3 (对于V853, 通常取值12, 表示vipp12)。
isp_dev_num_3: 指定isp设备ID, 通常取值0, 表示isp0。
pic_width_3: 指定vipp采集图像的宽度。
pic_height_3: 指定vipp采集图像的高度。
frame_rate_3: 指定vipp采集图像的帧率。
pic_format_3: 指定vipp采集图像的像素格式。
color_space_3: 指定vipp采集图像的颜色空间格式。
enable_wdr_mode_3: 指定vipp采集图像是否开启WDR。
drop_frm_num_3: 指定vipp开始前丢弃的帧数。

指定需要保存哪一路vipp的YUV数据, 以及保存的帧数和文件

save_pic_dev: 保存指定vipp的YUV数据。
yuv_frm_count: 指定保存的图像帧数。
yuv_file: 指定保存的路径和文件名。

保存 save_pic_dev 指定vipp的单帧YUV数据的总帧数、间隔和位置

raw_store_count: 指定保存单帧YUV数据的总帧数。

raw_store_interval: 指定保存单帧YUV数据的间隔, 单位: 帧。

store_dir: 指定保存单帧YUV数据的位置。

指定保存 YUV 的 buffer 长度和个数

save_pic_buffer_len: 指定保存 YUV 的 buffer 长度, 默认值0, 表示根据像素格式和分辨率自动换算 buffer 长度。

save_pic_buffer_num: 指定保存 YUV 的 buffer 个数。

test_duration: 测试时间, 单位: s。

测试指令:

【Tina】

```
# cd /mnt/extsd/
# ./sample_virvi -path ./sample_virvi.conf
```

【Melis】

```
# sample_virvi -path ./mnt/E/sample_virvi.conf
```

退出测试:

测试达到设定的时间后自动退出测试, 或者按 “ctrl + c” 提前结束测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 使用 YUVView 软件查看测试生成的 YUV 文件正常。

8.1.3 sample_virvi2vo

测试目的:

演示采集+预览。

该sample测试mpi_vi和mpi_vo组件的绑定组合。

创建mpi_vi和mpi_vo, 将它们绑定, 再分别启动。mpi_vi采集图像, 直接传输给mpi_vo显示。

组件依赖:

```
mpp_vi
mpp_hw_display
mpp_vo
```

测试通路:

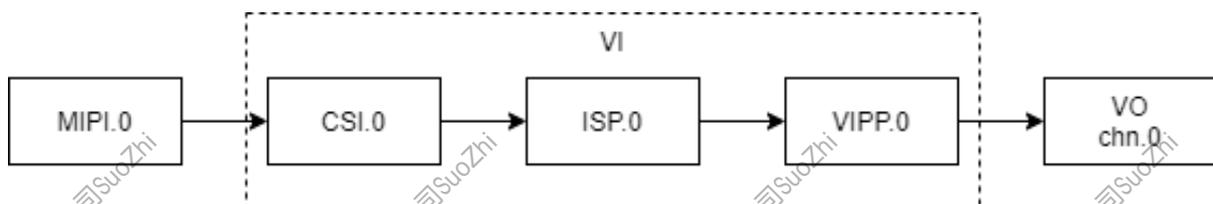


图 8-3: MPP_sample 测试通路-sample_virvi2vo

源文件：

无

目标文件：

无

参数配置：

以下配置支持测试：3路vipp采集+vo不同layer同时预览，再加上ui显示。

vipp0采集 + vo layer0预览的配置

capture_width: 指定vipp0采集图像的宽度。

capture_height: 指定vipp0采集图像的高度。

display_x: 指定vo显示输出图像的x坐标。

display_y: 指定vo显示输出图像的y坐标。

display_width: 指定vo显示输出图像的宽度。

display_height: 指定vo显示输出图像的高度。

layer_num: 指定vo显示的层数，通常取值为0，表示vo的layer0。

dev_num: 指定vipp设备号，通常取值为0，表示vipp0（对于V853，通常取值为0，表示vipp0）。

vipp4采集 + vo layer4预览的配置

capture_width2: 指定vipp0采集图像的宽度。

capture_height2: 指定vipp0采集图像的高度。

display_x2: 指定vo显示输出图像的x坐标。

display_y2: 指定vo显示输出图像的y坐标。

display_width2: 指定vo显示输出图像的宽度。

display_height2: 指定vo显示输出图像的高度。

layer_num2: 指定vo显示的层数，通常取值为4，表示vo的layer4。

dev_num2: 指定vipp设备号，通常取值为1，表示vipp1（对于V853，通常取值为4，表示vipp4）。

vipp8采集 + vo layer8预览的配置

capture_width3: 指定vipp0采集图像的宽度。

capture_height3: 指定vipp0采集图像的高度。

display_x3: 指定vo显示输出图像的x坐标。

display_y3: 指定vo显示输出图像的y坐标。

display_width3: 指定vo显示输出图像的宽度。

display_height3: 指定vo显示输出图像的高度。

layer_num3: 指定vo显示的层数，通常取值为8，表示vo的layer8。

dev_num3: 指定vipp设备号，通常取值为2，表示vipp2（对于V853，通常取值为8，表示vipp8）。

add_ui_layer: 是否增加UI层，默认0，表示不增加。

ui_test_layer: 指定UI显示的layer。

ui_display_width: 指定UI显示的宽度。

ui_display_height: 指定UI显示的高度。

disp_type: 指定显示设备类型，比如：lcd、hdmi、cvbs。

pic_format: 指定像素格式。

frame_rate: 指定帧率，单位：fps。

test_duration: 测试时间，单位：s。

测试指令：

【Tina】

```
# cd /mnt/extsd/  
# ./sample_virvi2vo -path ./sample_virvi2vo.conf
```

【Melis】

```
# sample_virvi2vo -path ./mnt/E/sample_virvi2vo.conf
```

退出测试：

测试达到设定的时间后自动退出测试，或者按“ctrl + c”提前结束测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 屏幕实时显示视频图像正常，无卡顿、花屏等。

8.1.4 sample_virvi2vo_zoom

测试目的：

演示直接使用pi_vi和mpi_vo组件实现图像的缩放，可支持4K图像缩放。
该sample测试mpi_vi和mpi_vo组件的非绑定。
mpi_vi采集图像，经过缩放算法处理后，把指定区域送给mpi_vo显示。

组件依赖：

```
mpp_vi  
mpp_hw_display  
mpp_vo
```

测试通路：

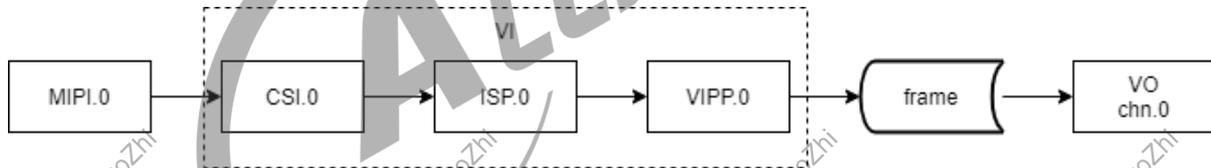


图 8-4: MPP_sample 测试通路-sample_virvi2vo_zoom

源文件：

无

目标文件：

无

参数配置：

```
capture_width: 指定camera采集图像的宽度。
capture_height: 指定camera采集图像的高度。
display_width: 指定vo显示输出图像的宽度。
display_height: 指定vo显示输出图像的高度。
dev_num: 指定vipp设备号。
disp_type: 指定vo显示设备类型 (hdmi, lcd, cvbs)。
pic_format: 指定camera采集的图像像素格式。
frame_rate: 指定camera采集的帧率，单位：fps。
```

test_duration: 指定测试时间, 单位: s。
 zoom_speed: 缩放的速度, 一般建议设置为10。值越小, 速度越快。
 zoom_max_cnt: 缩放的次数。

测试指令:

【Tina】

```
# cd /mnt/extsd/  
# ./sample_virvi2vo_zoom -path ./sample_virvi2vo_zoom.conf
```

【Melis】

```
# sample_virvi2vo_zoom -path /mnt/E/sample_virvi2vo_zoom.conf
```

结束测试:

达到缩放次数或指定测试时间后自动退出测试, 或者按 “ctrl + c” 提前结束测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 屏幕实时显示视频图像缩放效果正常, 无卡顿、花屏等。

8.1.5 sample_vi_reset

测试目的:

该sample 演示vi 组件的reset 流程。

组件依赖:

无

测试通路:

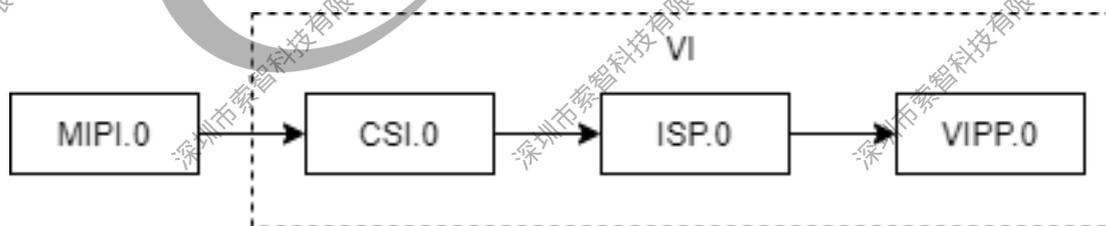


图 8-5: MPP_sample 测试通路-sample_vi_reset

源文件:

无

目标文件:

无

参数配置：

test_count：指定测试的次数。

指定测试vipp id的起止范围

vipp_id_start：指定测试起始的vipp id，通常取值0，表示vipp0（对于V853，表示vipp0）。

vipp_id_end：指定测试结束的vipp id，通常取值3，表示vipp3（对于V853，表示vipp12）。对于V853，由于当前vipp12被LDC（直方图均衡）功能占用，无法测试，故实测时取值2。即测试前3个vipp。

注：通过修改isp配置，关闭LDC功能后，可以测试全部4个vipp。

测试指令：

【Tina】

```
# cd /mnt/extsd/
# ./sample_vi_reset
```

【Melis】

暂不支持。

退出测试：

到达测试次数后自动退出测试，或者按“ctrl + c”提前结束测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。

8.1.6 sample_isposd

测试目的：

该sample测试mpi_vi和mpi_venc组件绑定。

创建mpi_vi和mpi_venc，将它们绑定，再分别启动。mpi_vi采集图像，调用mpi_vi相关接口获取实时的ISP参数添加overlay到VENC通道。

组件依赖：

```
mpp_vi
mpp_venc
mpp_system_rgb_ctrl
```

测试通路：

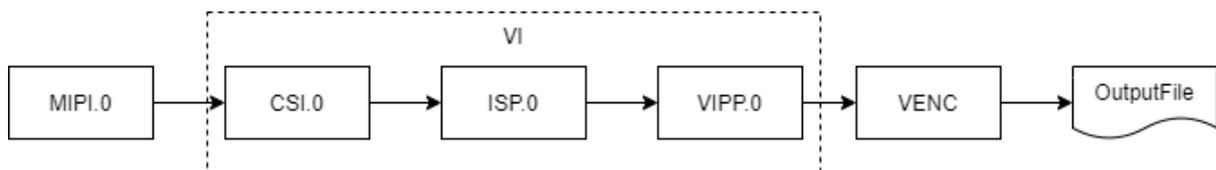


图 8-6: MPP sample 测试通路-sample_isposd

源文件：

无

目标文件：

/mnt/extsd/testRegion.h264

参数配置：

capture_width: 指定camera采集图像的宽度。
capture_height: 指定camera采集图像的高度。
pic_format: 指定camera采集图像的像素格式。
frame_rate: 指定camera采集的帧率。
test_duration: 指定测试时间, 单位: s。
overlay_x, overlay_y, overlay_w, overlay_h: 指定overlay类型的region参数。
cover_x, cover_y, cover_w, cover_h: 指定cover类型的region参数。
add_venc_channel: 指定是否选择VENC通道, 默认0, 表示不选择编码通道。
encoder_count: 保存编码视频的帧数。
bit_rate: 指定编码的码率, 单位: bps。
encoder_type: 指定编码类型 (H.265, H.264、MJPEG)。
output_file_path: 指定编码视频的保存路径。

测试指令：

【Tina】

```
# cd /mnt/extsd/  
# ./sample_isposd -path ./sample_isposd.conf
```

【Melis】

暂不支持。

退出测试：

到达测试时间后自动退出测试，或者按“ctrl + c”提前结束测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 生成的编码文件 testRegion.h264 的osd效果正常，且位置正确。

8.1.7 sample_vin_isp_test

测试目的：

该sample 用于测试ISP通路，是通过调用 isp 接口抓图的 demo。

组件依赖：

无

测试通路：

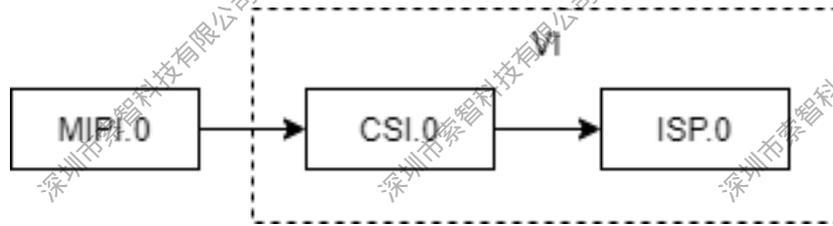


图 8-7: MPP_sample 测试通路-sample_vin_isp_test

源文件:

无

目标文件:

无

参数配置:

参数1: 选择通道
 参数2: 分辨率宽
 参数3: 分辨率高
 参数4: 保存路径
 参数5: 保存帧数
 参数6: 测试次数
 参数7: FPS
 参数8: WDR开关

测试指令:

【Tina】

```
# cd /mnt/extsd/
# ./sample_vin_isp_test
```

【Melis】

暂不支持。

退出测试:

测试达到指定次数后自动退出测试。

预期结果:

1. 测试程序运行正常，测试过程没有异常打印。
2. 使用YUVview 软件分析生成的YUV文件正常。

8.1.8 sample_region

测试目的:

演示vipp和venc的osd功能。对于V459和V833, vipp只支持orl, venc只支持overlay和cover。
 该sample测试mpi_vi和mpi_vo组件或者mpi_vi和mpi_venc组件绑定。
 创建mpi_vi和mpi_vo, 将它们绑定, 再分别启动。mpi_vi采集图像, 添加overlay和cover不同的region到VI通道, 传输给mpi_vo显示。
 也可以选择加入mpi_vi和mpi_venc组件绑定, 把region添加到VENC通道上, 保存编码后的视频数据文件, 可以用VLC播放。

组件依赖:

无

测试通路:

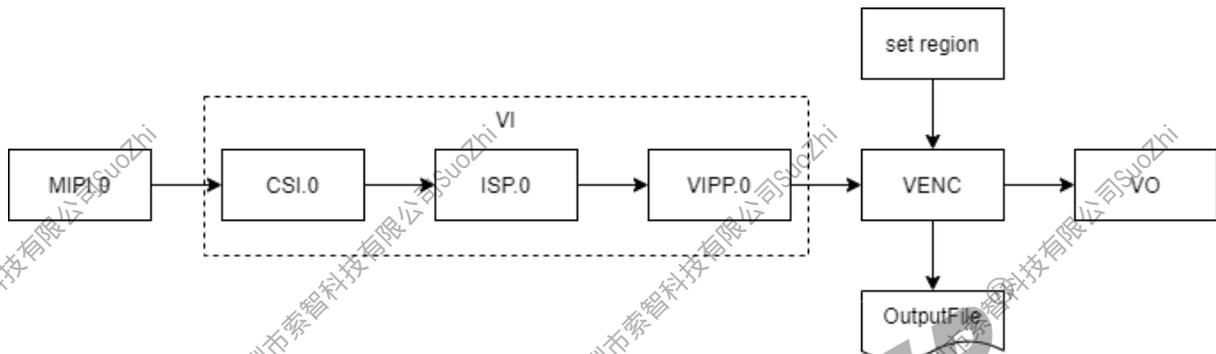


图 8-8: MPP_sample 测试通路-sample_region

源文件:

无

目标文件:

/mnt/extsd/testRegion.h264

参数配置:

online_en: 指定是否开启在线编码, 默认0, 表示不开启在线编码, 即当前为离线编码。
 online_share_buf_num: 指定在线编码的共享buffer个数 (1或2), 默认2, 表示2个buffer。注意该配置仅当配置在线模式时才生效。

vipp_dev: 指定vipp设备号, 默认0, 表示vipp0。

ve_ch_id: 指定编码通道, 默认0, 表示venc ch0。

rgn_attach_to_vi: 指定是否测试vipp的osd, 默认1, 表示测试vipp的osd, 其实主要是测试vipp orl功能。

add_venc_channel: 指定是否进行编码, 默认yes, 表示vipp采集后图像后送给venc编码。

rgn_attach_to_ve: 指定是否测试编码osd, 默认1, 表示测试编码osd。注意该配置仅当配置add_venc_channel=yes时才生效。

change_disp_attr_enable: 指定开启/关闭改变osd显示位置测试, 默认1, 表示开启该测试。

以下编码相关参数仅当配置add_venc_channel=yes时生效

encoder_count: 指定编码的总帧数。

bit_rate: 指定编码的码率, 单位: bps。

encoder_type: 指定编码格式 (H.264, H265)。

output_file_path: 指定编码文件的存放路径。

配置Vipp采集参数

capture_width: 指定camera采集图像的宽度。
capture_height: 指定camera采集图像的高度。

pic_format: 指定像素格式 (nv21, nv12, yu12, yv12; aw_lbc_2_5x, aw_lbc_2_0x, aw_lbc_1_5x, aw_lbc_1_0x)。

frame_rate: 指定帧率, 单位: fps。

注: 当设置以下显示的宽度和高度为0时, 表示不显示。

disp_width: 指定显示的宽度。
disp_height: 指定显示的高度。

bitmap_format: 指定位图的格式 (ARGB8888、ARGB1555)。

注: 当设置以下overlay、cover、orl的宽度和高度为0时, 表示不进行该项测试。

overlay_x: 指定overlay的x坐标。
overlay_y: 指定overlay的y坐标。
overlay_w: 指定overlay的宽度。
overlay_h: 指定overlay的高度。

cover_x: 指定cover的x坐标。
cover_y: 指定cover的y坐标。
cover_w: 指定cover的宽度。
cover_h: 指定cover的高度。

orl_x: 指定orl矩形的x坐标。
orl_y: 指定orl矩形的y坐标。
orl_w: 指定orl矩形的宽度。
orl_h: 指定orl矩形的高度。
orl_thick: 指定orl线条的宽度, 单位: 像素个数, 默认2, 表示2个像素的宽度。

test_duration: 指定测试时间, 单位: s。

测试指令:**【Tina】**

```
# cd /mnt/extsd/  
# ./sample_region -path ./sample_region.conf
```

【Melis】

```
# sample_region -path ./mnt/E/sample_region.conf
```

退出测试:

测试达到设定的时间后自动退出测试, 或者按 “ctrl + c” 提前结束测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 生成的编码文件 testRegion.h264 的osd效果正常, 且位置正确。
3. 屏幕实时显示Camera 采集的视频图像, 同时, 显示一个绿色的框。

8.1.9 sample_venc

测试目的：

该sample 演示从yuv(sample中格式限定为yuv420p)原始数据文件xxx.yuv中读取视频帧，编码，将取得的编码往输出文件里面直接写。生成裸码流视频文件。

注意：

如果是h264或h265编码sample会自动在目标文件的开始加上spssps信息，其他格式则不加。

组件依赖：

mpp_venc

测试通路：

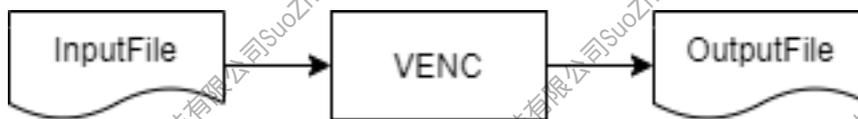


图 8-9: MPP_sample 测试通路-sample_venc

源文件：

/mnt/extsd/test.yuv

目标文件：

/mnt/extsd/test.raw

参数配置：

src_file: 指定原始yuv文件的路径。
 src_width: 指定原始视频的宽度。
 src_height: 指定原始视频的高度。
 dst_file: 指定生成的裸码流视频文件路径。
 dst_width: 指定生成的裸码流视频的宽度。
 dst_height: 指定生成的裸码流视频的高度。

src_pixfmt: 指定像素格式 (YUV类型: nv21(yuv420sp), yu12(yuv420p), yv12, nv12。LBC压缩类型: aw_lbc_2_0x, aw_lbc_2_5x, aw_lbc_1_5x, aw_lbc_1_0x)。
 color_space: 指定颜色空间 (jpeg, rec709, rec709_part_range)。
 encoder: 指定编码格式 (H.264、H.265、MJPEG)。
 profile: 指定编码质量, 对于H264, 建议配置High (2), 对于H265, 建议配置Main (0)。
 framerate: 指定帧率, 单位: fps。
 bitrate: 指定编码的码率, 单位: bps。

rotate: 指定编码旋转角度 (0, 90, 180, 270), 顺时针方向。
 rc_mode: 指定码率控制模式 (0:CBR, 1:VBR, 2:FIXQP, 3:QPMAP)。
 gop_mode: 指定GOP模式 (0:NormalP, 1:DualP, 2:SmartP)。
 gop_size: 指定GOP大小, 当前只针对H265有效, 取值范围[1, 63]。
 product_mode: 指定产品类型 (0:Normal, 1:IPC)。
 sensor_type: 指定sensor类型 (0:DisWdr, 1:EnWdr)。
 key_frame_interval: 指定I帧间隔。

VBR、CBR模式, 编码参数

init_qp: 指定初始QP值, 取值范围(0, 51)。
 min_i_qp: 指定I帧最小QP值, 取值范围(0, 51)。
 max_i_qp: 指定I帧最大QP值, 取值范围(0, 51)。
 min_p_qp: 指定P帧最小QP值, 取值范围(0, 51)。
 max_p_qp: 指定P帧最大QP值, 取值范围(0, 51)。

FIXQP模式, 编码参数

i_qp: 指定I帧的QP值, 取值范围(0, 51)

p_qp: 指定P帧的QP值, 取值范围(0, 51)

VBR模式, 编码参数

moving_th: 指定mb16x16中MAD的阈值, 取值范围[0, 31]。

quality: 指定静态P帧的位系数, 取值范围[1, 20]。

p_bits_coef: 指定移动P帧的位系数, 取值范围[1, 50]。

i_bits_coef: 指定I帧的位系数, 取值范围[1, 20]。

test_duration: 测试时间, 单位: s。

测试指令:

【Tina】

```
# cd /mnt/extsd/
# ./sample_venc -path ./sample_venc.conf
```

【Melis】

暂不支持。

退出测试:

测试达到设定的时间后自动退出测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 使用VLC 软件播放裸码流文件test.raw正常。

8.1.10 sample_venc2muxer

测试目的:

从yuv原始数据文件xxx.yuv中读取视频帧, 编码, 并由muxer进行封装生成相应的视频输出文件。

组件依赖:

```
mpp_venc
mpp_muxer
```

测试通路:

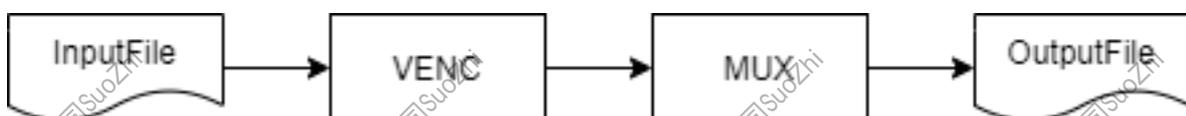


图 8-10: MPP_sample 测试通路-sample_venc2muxer

源文件：

```
/mnt/extsd/1080p.yuv
```

目标文件：

```
/mnt/extsd/1080p.mp4
```

参数配置：

```
yuv_src_file: 指定原始yuv文件的路径。
yuv_src_size: 指定原始视频文件的视频大小，如1080p。

视频源的region区域，当设置region_w=0时，不开启该功能。
以下宽高和坐标要求16对齐。
region_x: 指定region区域的x坐标。
region_y: 指定region区域的y坐标。
region_w: 指定region区域的宽度。
region_h: 指定region区域的高度。

yuv_src_pixfmt: 指定像素格式 (nv21) 。

video_dst_file: 指定生成的视频文件路径。
video_size: 指定生成的视频文件视频大小，如1080p。

video_encoder: 指定视频编码格式 (H.264、H.265、MJPEG) 。
profile: 指定编码质量，对于H264，建议配置High (2)，对于H265，建议配置Main (0)。
rc_mode: 指定码率控制模式 (0:CBR, 1:VBR, 2:FIXQP, 3:QPMP) 。

video_framerate: 指定生成视频文件的帧率，单位：fps。
video_bitrate: 指定生成视频文件的码率，单位：bps。

video_duration: 指定生成一个视频文件的最大持续时间（如：每个视频文件长度一分钟），单位：s。
media_file_format: 指定视频文件的封装格式，支持mp4和ts。

test_duration: 指定测试时间，单位：s。
```

测试指令：

【Tina】

```
# cd /mnt/extsd/
# ./sample_venc2muxer -path ./sample_venc2muxer.conf
```

【Melis】

```
暂不支持。
```

退出测试：

```
测试达到设定的时间后自动退出测试。
```

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 使用VLC\软件播放生成的mp4文件正常。

8.1.11 sample_virvi2venc

测试目的：

该sample测试mpi_vi和mpi_venc组件的绑定组合。创建mpi_vi和mpi_venc，将它们绑定，再分别启动。mpi_vi采集图像，直接传输给mpi_venc进行编码。

组件依赖：

mpp_vi
mpp_venc

测试通路：

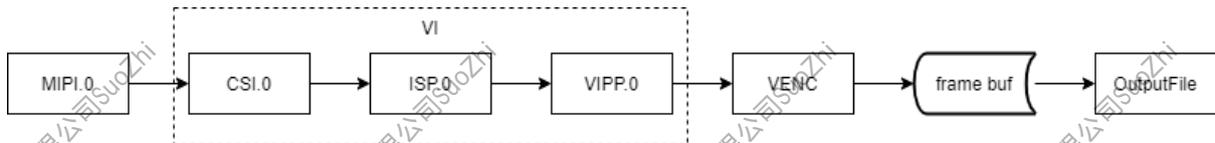


图 8-11: MPP_sample 测试通路-sample_virvi2venc

源文件：

无

目标文件：

/mnt/extsd/test.raw

参数配置：

online_en: 指定是否开启在线编码，默认0，表示不开启在线编码，即当前为离线编码。
 online_share_buf_num: 指定在线编码的共享buffer个数（1或2），默认2，表示2个buffer。注意该配置仅当配置在线模式时才生效。
 vipp_id: 指定vipp设备号，默认0，表示vipp0。
 wdr_en: 指定是否开启WDR，默认0，表示不开WDR。
 drop_frm_num: 指定丢帧个数，对于离线编码是vipp丢帧，对于在线编码是ve编码前丢帧。
 src_width: 指定原始视频的宽度。
 src_height: 指定原始视频的高度。
 vi_buffer_num: 指定VI的buffer个数，仅对离线编码有效。
 saturation_change: 指定饱和度改变的值，默认0，表示不调整。取值范围[-256, 512]。
 src_pixfmt: 指定像素格式（YUV类型：nv21(yvu420sp), yu12(yuv420p), yv12, nv12。LBC压缩类型：aw_lbc_2_0x, aw_lbc_2_5x, aw_lbc_1_5x, aw_lbc_1_0x）。
 color_space: 指定颜色空间（jpeg, rec709, rec709_part_range）。
 venc_ch_id: 指定编码通道，默认0，表示venc ch0。
 video_dst_file: 指定编码文件的存放路径。
 video_framerate: 指定帧率，单位：fps。
 video_bitrate: 指定编码的码率，单位：bps。

video_width: 指定生成的裸码流视频的宽度。
video_height: 指定生成的裸码流视频的高度。

video_encoder: 指定编码格式 (H.264、H.265、MJPEG)。
profile: 指定编码质量, 对于H264, 建议配置High (2), 对于H265, 建议配置Main (0)。

ve_freq: 指定VE的频率, 默认0, 表示300MHz, 单位: MHz。

product_mode: 指定产品类型 (0:Normal, 1:IPC)。
sensor_type: 指定sensor类型 (0:DisWdr, 1:EnWdr)。
key_frame_interval: 指定I帧间隔。

enable_gdc: 指定是否开启GDC功能, 默认0, 表示不开GDC。
rc_mode: 指定码率控制模式 (0:CBR, 1:VBR, 2:FIXQP, 3:QPMP)。

VBR、CBR模式, 编码参数

init_qp: 指定初始QP值, 取值范围(0, 51)。
min_i_qp: 指定I帧最小QP值, 取值范围(0, 51)。FIXQP模式时, 取其值为i_qp。
max_i_qp: 指定I帧最大QP值, 取值范围(0, 51)。
min_p_qp: 指定P帧最小QP值, 取值范围(0, 51)。FIXQP模式时, 取其值为p_qp。
max_p_qp: 指定P帧最大QP值, 取值范围(0, 51)。
mb_qp_limit_en: 指定是否开启mb qp限制, 默认0, 表示不开启。

VBR模式, 编码参数

moving_th: 指定mb16x16中MAD的阈值, 取值范围[0, 31]。
quality: 指定静态P帧的位系数, 取值范围[1, 20]。
p_bits_coef: 指定移动P帧的位系数, 取值范围[1, 50]。
i_bits_coef: 指定I帧的位系数, 取值范围[1, 20]。

gop_mode: 指定GOP模式 (0:NormalP, 1:DualP, 2:SmartP)。
gop_size: 指定GOP大小, 当前只针对H265有效, 取值范围[1, 63]。

高级跳帧参数

AdvancedRef_Base: 设置该值大于0表示开启高级跳帧功能, 等于0表示关闭高级跳帧功能。
AdvancedRef_Enhance: 设置Enhance为5。
AdvancedRef_RefBaseEn: 设置该值为1表示开启参考帧, 等于0表示关闭参考帧。

enable_fast_enc: 指定是否开启快速编码, 默认0, 表示不开启。

encode_rotate: 指定编码旋转角度 (0, 90, 180, 270), 顺时针方向。
mirror: 指定编码镜像是否开启, 默认0, 表示不开启。取值范围[0, 1]。

video_duration: 指定生成一个视频文件的最大持续时间 (如: 每个视频文件长度一分钟), 单位: s。
test_duration: 指定测试时间, 单位: s。

color2grey: 指定是否开启彩转灰功能, 默认"no", 表示不开启。取值范围["no", "yes"]。

2D降噪参数配置

2dnr_en: 编码器2DNR使能, 默认1, 表示开启。取值范围[0, 1]。
2dnr_strength_y: 亮度降噪强度系数。该值越大, 滤波强度越高。默认127, 取值范围[0, 255]。
2dnr_strength_c: 色度降噪强度系数。该值越大, 滤波强度越高。默认127, 取值范围[0, 255]。
2dnr_threshold_y: 亮度邻域像素降噪开关阈值。该值越大, 越容易触发2d滤波。默认7, 取值范围[0, 15]。
2dnr_threshold_c: 色度邻域像素降噪开关阈值。该值越大, 越容易触发2d滤波。默认7, 取值范围[0, 15]。

3D降噪参数配置

3dnr_en: 编码器3DNR使能, 默认1, 表示开启。取值范围[0, 1]。
3dnr_pix_level_en: 3d滤波权重系数自适应修正开关。默认0, 表示不开启。取值范围[0, 1]。
3dnr_smooth_en: 像素级3x3平滑滤波使能。默认1, 表示开启。取值范围[0, 1]。
3dnr_pix_diff_th: 仅当3d_adjust_pix_level_enable为1时生效, 自适应修正幅度阈值。该值越大, 滤波强度越高。默认6, 取值范围[0, 31]。

3dnr_max_mv_th: 源图像素块运动矢量最大阈值, 单个像素块仅当其水平和垂直MV皆小于该阈值, 才会进行3d滤波。该值越大, 越容易触发3d滤波。默认2, 取值范围[0, 63]。

3dnr_max_mad_th: 源图与参考图之间像素块的MAD最大阈值, 单个像素块仅当其MAD值小于该阈值, 才会进行3d滤波。该值越大, 越容易触发3d滤波。默认11, 取值范围[0, 63]。

3dnr_min_coef: 3d滤波权重系数的最小阈值。该值越小, 自适应3d滤波的强度上限越高。默认14, 取值范围[0, 3dnr_max_coef]。

3dnr_max_coef: 3d滤波权重系数的最大阈值。该值越大, 自适应3d滤波的强度下限越低。默认16, 取值范围[3dnr_min_coef, 16]。

ROI测试参数配置

roi_num: 指定ROI的个数, 默认0, 表示不开启ROI测试。取值范围[0, 8]。

roi_qp: 指定ROI区域的QP值, 默认50, 表示最差质量。取值范围(0, 51)。

roi_BgFrameRateEnable: 指定是否开启非ROI区域低帧率编码, 默认0, 表示不开启。

roi_BgFrameRateAttenuation: 指定非ROI区域的帧率衰减比例, 默认3, 表示非ROI区域的帧率是正常帧率的1/3。

IntraRefresh_BlockNum: 指定P帧帧内刷新的block个数, 默认0, 表示不开启P帧帧内刷新功能。

orl_num: 指定ORL(Object Rectangle Label)的个数, 取值范围[0,16]。

配置vbv buffer大小和vbv buffer阈值大小

vbvBufferSize: 指定vbv buffer大小, 默认0, 表示由mpp middleware自行计算并设置给编码器, 大于0, 表示由app设置。

vbvThreshSize: 指定vbv buffer阈值大小, 默认0, 表示由mpp middleware自行计算并设置给编码器, 大于0, 表示由app设置。

裁剪编码参数设置

crop_en: 指定是否开启裁剪编码, 默认0, 表示不开启。

crop_rect_x: 指定裁剪区域的x坐标。

crop_rect_y: 指定裁剪区域的y坐标。

crop_rect_w: 指定裁剪区域的宽度。

crop_rect_h: 指定裁剪区域的高度。

vui_timing_info_present_flag: 指定是否添加VUI信息, 默认0, 表示不添加。

encpp_disable: 指定是否禁用encpp新通路, 默认0, 表示不禁用。

super_frm_mode: 指定超级帧模式, 默认0, 表示不启用。(0:none, 1:discarded, 2:re-encode)。

测试指令:

【Tina】

```
# cd /mnt/extsd/  
# ./sample_virvi2venc -path ./sample_virvi2venc.conf
```

【Melis】

```
# sample_virvi2venc -path ./mnt/E/sample_virvi2venc.conf
```

退出测试:

测试达到设定的时间后自动退出测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 使用VLC 软件播放生成的裸码流文件/mnt/extsd/test.raw正常。

8.1.12 sample_timelapse

测试目的：

演示缩时录影。

从camera节点取vi输入数据，对venc组件设置取帧间隔，设置编码帧率，并对数据进行编码封装，生成对应的视频输出文件。

组件依赖：

mpp_vi
mpp_venc
mpp_muxer

测试通路：

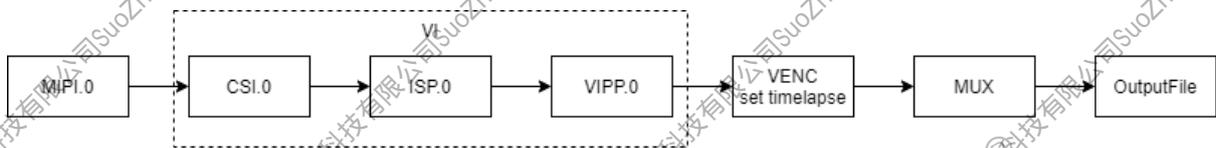


图 8-12: MPP_sample 测试通路-sample_timelapse

源文件：

无

目标文件：

/mnt/extsd/timelapse.mp4

参数配置：

vipp_index: 获取视频数据的VIPP设备号，默认0，表示vipp0。通常取值0/1/2/3，（对于V853，通常取值0/4/8/12）。

vipp_width: vipp输出视频图像的宽度。

vipp_height: vipp输出视频图像的高度。

vipp_frame_rate: vipp的采集帧率，单位：fps。

timelapse: 缩时录影模式下取帧的帧间隔，单位：us。

video_frame_rate: 编码帧率，即设置编码文件的播放帧率。

video_duration: 编码文件的播放时长。

video_bitrate: 编码码率，单位：Mbps。

video_file_path: 编码文件的存储路径。

测试指令：

【Tina】

```
# cd /mnt/extsd/
# ./sample_timelapse -path ./sample_timelapse.conf
```

【Melis】

暂不支持。

退出测试：

最终自动退出测试，默认测试时间有点长，可以按“ctrl + c”退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 使用VLC 软件播放生成的mp4文件/mnt/extsd/timelapse.mp4正常。

8.1.13 sample_virvi2venc2muxer

测试目的：

从camera 节点取vi 输入数据，并对数据进行编码封装，生成对应的视频输出文件。

组件依赖：

```
mpi_vi
mpp_venc
mpp_muxer
```

测试通路：

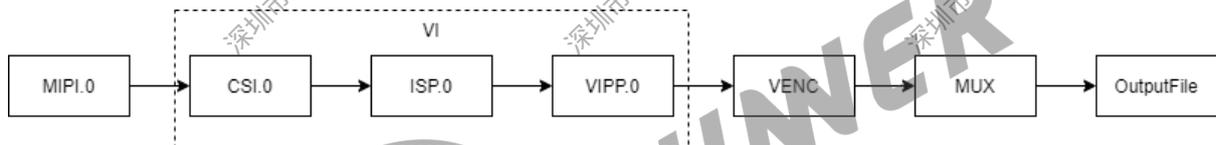


图 8-13: MPP_sample 测试通路-sample_virvi2venc2muxer

源文件：

无

目标文件：

/mnt/extsd/test.mp4

参数配置：

online_en: 指定是否开启在线编码，默认0，表示不开启在线编码，即当前为离线编码。
 online_share_buf_num: 指定在线编码的共享buffer个数（1或2），默认2，表示2个buffer。注意该配置仅当配置在线模式时才生效。

vipp_id: 指定vipp设备号，默认0，表示vipp0。
 wdr_en: 指定是否开启WDR，默认0，表示不开WDR。

drop_frm_num: 指定丢帧个数，对于离线编码是vipp丢帧，对于在线编码是ve编码前丢帧。

src_width: 指定原始视频的宽度。
 src_height: 指定原始视频的高度。

vi_buffer_num: 指定VI的buffer个数，仅对离线编码有效。
 saturation_change: 指定饱和度改变的值，默认0，表示不调整。取值范围[-256, 512]。

src_pixfmt: 指定像素格式（YUV类型：nv21(yuv420sp), yu12(yuv420p), yv12, nv12。LBC压缩类型：
 aw_lbc_2_0x, aw_lbc_2_5x, aw_lbc_1_5x, aw_lbc_1_0x）。

color_space: 指定颜色空间 (jpeg, rec709, rec709_part_range)。

venc_ch_id: 指定编码通道, 默认0, 表示venc ch0。

video_dst_file: 指定编码文件的存放路径。

add_repair_info: 指定是否为mp4文件添加文件修复信息。默认0, 表示不添加。

frmsTag_backup_interval: 指定mp4文件修复的frame tag备份的时间间隔, 单位:us。

dst_file_max_cnt: 指定muxer封装mp4文件, 最多保留的文件个数。默认3, 表示最多只允许保留3个mp4文件。

video_framerate: 指定帧率, 单位: fps。

video_bitrate: 指定编码的码率, 单位: bps。

video_width: 指定生成的裸码流视频的宽度。

video_height: 指定生成的裸码流视频的高度。

video_encoder: 指定编码格式 (H.264、H.265、MJPEG)。

profile: 指定编码质量, 对于H264, 建议配置High (2), 对于H265, 建议配置Main (0)。

ve_freq: 指定VE的频率, 默认0, 表示300MHz, 单位: MHz。

product_mode: 指定产品类型 (0:Normal, 1:IPC)。

sensor_type: 指定sensor类型 (0:DisWdr, 1:EnWdr)。

key_frame_interval: 指定I帧间隔。

enable_gdc: 指定是否开启GDC功能, 默认0, 表示不开GDC。

rc_mode: 指定码率控制模式 (0:CBR, 1:VBR, 2:FIXQP, 3:QMAP)。

VBR、CBR模式, 编码参数

init_qp: 指定初始QP值, 取值范围(0, 51)。

min_i_qp: 指定I帧最小QP值, 取值范围(0, 51)。FIXQP模式时, 取其值为i_qp。

max_i_qp: 指定I帧最大QP值, 取值范围(0, 51)。

min_p_qp: 指定P帧最小QP值, 取值范围(0, 51)。FIXQP模式时, 取其值为p_qp。

max_p_qp: 指定P帧最大QP值, 取值范围(0, 51)。

mb_qp_limit_en: 指定是否开启mb qp限制, 默认0, 表示不开启。

VBR模式, 编码参数

moving_th: 指定mb16x16中MAD的阈值, 取值范围[0, 31]。

quality: 指定静态P帧的位系数, 取值范围[1, 20]。

p_bits_coef: 指定移动P帧的位系数, 取值范围[1, 50]。

i_bits_coef: 指定I帧的位系数, 取值范围[1, 20]。

gop_mode: 指定GOP模式 (0:NormalP, 1:DualP, 2:SmartP)。

gop_size: 指定GOP大小, 当前只针对H265有效, 取值范围[1, 63]。

高级跳帧参数

AdvancedRef_Base: 设置该值大于0表示开启高级跳帧功能, 等于0表示关闭高级跳帧功能。

AdvancedRef_Enhance: 设置Enhance为5。

AdvancedRef_RefBaseEn: 设置该值为1表示开启参考帧, 等于0表示关闭参考帧。

enable_fast_enc: 指定是否开启快速编码, 默认0, 表示不开启。

encode_rotate: 指定编码旋转角度 (0, 90, 180, 270), 顺时针方向。

mirror: 指定编码镜像是否开启, 默认0, 表示不开启。取值范围[0, 1]。

video_duration: 指定生成一个视频文件的最大持续时间 (如: 每个视频文件长度一分钟), 单位: s。

test_duration: 指定测试时间, 单位: s。

color2grey: 指定是否开启彩转灰功能, 默认"no", 表示不开启。取值范围["no", "yes"]。

2D降噪参数配置

2dnr_en: 编码器2DNR使能, 默认1, 表示开启。取值范围[0, 1]。
 2dnr_strength_y: 亮度降噪强度系数。该值越大, 滤波强度越高。默认127, 取值范围[0, 255]。
 2dnr_strength_c: 色度降噪强度系数。该值越大, 滤波强度越高。默认127, 取值范围[0, 255]。
 2dnr_threshold_y: 亮度邻域像素降噪开关阈值。该值越大, 越容易触发2d滤波。默认7, 取值范围[0, 15]。
 2dnr_threshold_c: 色度邻域像素降噪开关阈值。该值越大, 越容易触发2d滤波。默认7, 取值范围[0, 15]。

3D降噪参数配置

3dnr_en: 编码器3DNR使能, 默认1, 表示开启。取值范围[0, 1]。
 3dnr_pix_level_en: 3d滤波权重系数自适应修正开关。默认0, 表示不开启。取值范围[0, 1]。
 3dnr_smooth_en: 像素级3x3平滑滤波使能。默认1, 表示开启。取值范围[0, 1]。
 3dnr_pix_diff_th: 仅当3d_adjust_pix_level_enable为1时生效, 自适应修正幅度阈值。该值越大, 滤波强度越高。默认6, 取值范围[0, 31]。
 3dnr_max_mv_th: 源图像素块运动矢量最大阈值, 单个像素块仅当其水平和垂直MV皆小于该阈值, 才会进行3d滤波。该值越大, 越容易触发3d滤波。默认2, 取值范围[0, 63]。
 3dnr_max_mad_th: 源图与参考图之间像素块的MAD最大阈值, 单个像素块仅当其MAD值小于该阈值, 才会进行3d滤波。该值越大, 越容易触发3d滤波。默认11, 取值范围[0, 63]。
 3dnr_min_coef: 3d滤波权重系数的最小阈值。该值越小, 自适应3d滤波的强度上限越高。默认14, 取值范围[0, 3dnr_max_coef]。
 3dnr_max_coef: 3d滤波权重系数的最大阈值。该值越大, 自适应3d滤波的强度下限越低。默认16, 取值范围[3dnr_min_coef, 16]。

ROI测试参数配置

roi_num: 指定ROI的个数, 默认0, 表示不开启ROI测试。取值范围[0, 8]。
 roi_qp: 指定ROI区域的QP值, 默认50, 表示最差质量。取值范围(0, 51]。
 roi_BgFrameRateEnable: 指定是否开启非ROI区域低帧率编码, 默认0, 表示不开启。
 roi_BgFrameRateAttenuation: 指定非ROI区域的帧率衰减比例, 默认3, 表示非ROI区域的帧率是正常帧率的1/3。

IntraRefresh_BlockNum: 指定P帧帧内刷新的block个数, 默认0, 表示不开启P帧帧内刷新功能。

orl_num: 指定ORL(Object Rectangle Label)的个数, 取值范围[0,16]。

配置vbv buffer大小和vbv buffer阈值大小

vbvBufferSize: 指定vbv buffer大小, 默认0, 表示由mpp middleware自行计算并设置给编码器, 大于0, 表示由app设置。

vbvThreshSize: 指定vbv buffer阈值大小, 默认0, 表示由mpp middleware自行计算并设置给编码器, 大于0, 表示由app设置。

裁剪编码参数设置

crop_en: 指定是否开启裁剪编码, 默认0, 表示不开启。

crop_rect_x: 指定裁剪区域的x坐标。

crop_rect_y: 指定裁剪区域的y坐标。

crop_rect_w: 指定裁剪区域的宽度。

crop_rect_h: 指定裁剪区域的高度。

vui_timing_info_present_flag: 指定是否添加VUI信息, 默认0, 表示不添加。

encpp_disable: 指定是否禁用encpp新通路, 默认0, 表示不禁用。

super_frm_mode: 指定超级帧模式, 默认0, 表示不启用。(0:none, 1:discarded, 2:re-encode)。

测试指令:

[Tina]

```
# cd /mnt/extsd/
# ./sample_virvi2venc2muxer -path ./sample_virvi2venc2muxer.conf
```

[Melis]

```
# sample_virvi2venc2muxer -path ./mnt/E/sample_virvi2venc2muxer.conf
```

退出测试：

测试达到设定的时间后自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 使用VLC 软件播放生成的mp4文件正常。

8.1.14 sample_multi_vi2venc2muxer

测试目的：

最多允许4路编码加1路jpeg拍照。每个编码通道都可以选择vipp，以及timelapse模式等。

默认的配置如下：

VIPP0->venc0->muxerGrp0: bufferNum=5, 采集分辨率1920x1080, 格式LBC2.5, h264编码帧率30, 码率1Mbit/s, threshSize: w*h/10, 普通NormalP编码, vbv缓存1秒。

VIPP1->venc1->muxerGrp1: bufferNum=5, 采集分辨率640x480, 格式NV21, h264编码帧率30, 码率500Kbit/s, threshSize: w*h/10, 普通NormalP编码, vbv缓存1秒。

VIPP1->venc2->muxerGrp2: h264编码帧率30, 码率500Kbit/s, threshSize: w*h/10, 普通NormalP编码, vbv缓存1秒, timelapse = 200ms。

VIPP1->venc3->muxerGrp3: h264编码帧率30, 码率500Kbit/s, threshSize: w*h/10, 普通NormalP编码, vbv缓存1秒, timelapse = 1000ms。

VIPP0->venc_jpeg: 每隔30秒拍照一张。

组件依赖：

mpp_vi
mpp_venc
mpp_muxer

测试通路：

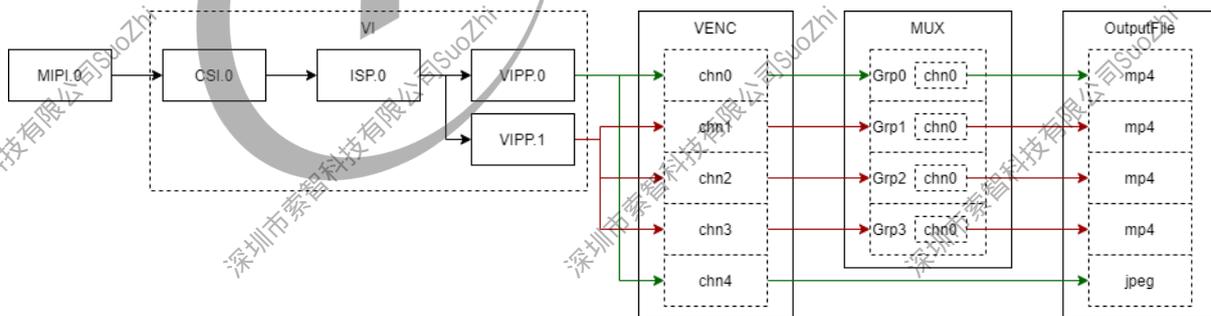


图 8-14: MPP_sample 测试通路-sample_multi_vi2venc2muxer

源文件：

无

目标文件：

/mnt/extsd/1080p_1.mp4
/mnt/extsd/1080p_2.mp4
/mnt/extsd/1080p_3.mp4

```
/mnt/extsd/1080p_4.mp4
/mnt/extsd/360p_1.mp4
/mnt/extsd/360p_2.mp4
/mnt/extsd/360p_3.mp4
/mnt/extsd/360p_4.mp4
/mnt/extsd/timelapse_first.mp4
/mnt/extsd/timelapse_second.mp4
/mnt/extsd/1080p_1.jpeg
/mnt/extsd/1080p_2.jpeg
/mnt/extsd/1080p_3.jpeg
```

参数配置：

vipp0: 第1条vpp通路的vipp设备号。
vipp0_format: 图像采集格式。
vipp0_capture_width: 图像采集宽度。
vipp0_capture_height: 图像采集高度。
vipp0_framerate: 图像采集帧率, 单位: fps。

vipp1: 第2条vpp通路的vipp设备号。
vipp1_format: 图像采集格式。
vipp1_capture_width: 图像采集宽度。
vipp1_capture_height: 图像采集高度。
vipp1_framerate: 图像采集帧率, 单位: fps。

主码流

videoA_vipp: 视频编码通路连接的vipp设备号。
videoA_file: 录制的视频文件路径。
videoA_file_cnt: 循环录制允许的最大文件数目, 超过就开始删除最早的文件。
videoA_framerate: 编码目标帧率, 单位: fps。
videoA_bitrate: 编码码率, 单位: bps。
videoA_width: 编码输出宽度。
videoA_height: 编码输出高度。
videoA_encoder: 编码格式。
videoA_rc_mode: 码率模式, 0:CBR 1:VBR。
videoA_duration: 文件时长。
videoA_timelapse: timelapse模式, -1:禁止timelapse; 0:慢摄影; >0:timelapse模式, 数值为采集帧的帧间隔, 单位us。

子码流

videoB_vipp: 视频编码通路连接的vipp设备号。
videoB_file: 录制的视频文件路径。
videoB_file_cnt: 循环录制允许的最大文件数目, 超过就开始删除最早的文件。
videoB_framerate: 编码目标帧率, 单位: fps。
videoB_bitrate: 编码码率, 单位: bps。
videoB_width: 编码输出宽度。
videoB_height: 编码输出高度。
videoB_encoder: 编码格式。
videoB_rc_mode: 码率模式, 0:CBR 1:VBR。
videoB_duration: 文件时长。
videoB_timelapse: timelapse模式, -1:禁止timelapse; 0:慢摄影; >0:timelapse模式, 数值为采集帧的帧间隔, 单位us。

缩时录影1

videoC_vipp: 视频编码通路连接的vipp设备号。
videoC_file: 录制的视频文件路径。
videoC_file_cnt: 循环录制允许的最大文件数目, 超过就开始删除最早的文件。
videoC_framerate: 编码目标帧率, 单位: fps。
videoC_bitrate: 编码码率, 单位: bps。
videoC_width: 编码输出宽度。

videoC_height: 编码输出高度。
videoC_encoder: 编码格式。
videoC_rc_mode; 码率模式, 0:CBR 1:VBR。
videoC_duration: 文件时长。
videoC_timelapse: timelapse模式, -1:禁止timelapse; 0:慢摄影; >0:timelapse模式, 数值为采集帧的帧间隔, 单位us。

缩时录影2

videoD_vipp: 视频编码通路连接的vipp设备号。
videoD_file: 录制的视频文件路径。
videoD_file_cnt: 循环录制允许的最大文件数目, 超过就开始删除最早的文件。
videoD_framerate: 编码目标帧率, 单位: fps。
videoD_bitrate: 编码码率, 单位: bps。
videoD_width; 编码输出宽度。
videoD_height: 编码输出高度。
videoD_encoder: 编码格式。
videoD_rc_mode; 码率模式, 0:CBR 1:VBR。
videoD_duration: 文件时长。
videoD_timelapse: timelapse模式, -1:禁止timelapse; 0:慢摄影; >0:timelapse模式, 数值为采集帧的帧间隔, 单位us。

JPEG拍照

videoE_vipp: 拍照通路连接的vipp设备号。
videoE_file_jpeg: 拍照的图片路径。
videoE_file_jpeg_cnt: 拍照的图片张数。
videoE_width: 编码输出宽度。
videoE_height: 编码输出高度。
videoE_encoder: 编码格式。
videoE_photo_interval: 拍照时间间隔, 单位: s。

test_duration: 测试时间, 单位: s。0表示无限时长。

测试指令:

[Tina]

```
# cd /mnt/extsd/  
# ./sample_multi_vi2venc2muxer -path ./sample_multi_vi2venc2muxer.conf
```

[Melis]

```
# sample_multi_vi2venc2muxer -path ./mnt/E/sample_multi_vi2venc2muxer.conf
```

退出测试:

到达设定的测试时长后会自动退出测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 使用VLC 软件播放生成的mp4文件正常。

8.1.15 sample_rtsp

测试目的:

演示视频编码后的rtsp传输的场景。

组件依赖：

```
mpp_vi
mpp_venc
mpp_system_rtsp
```

测试通路：

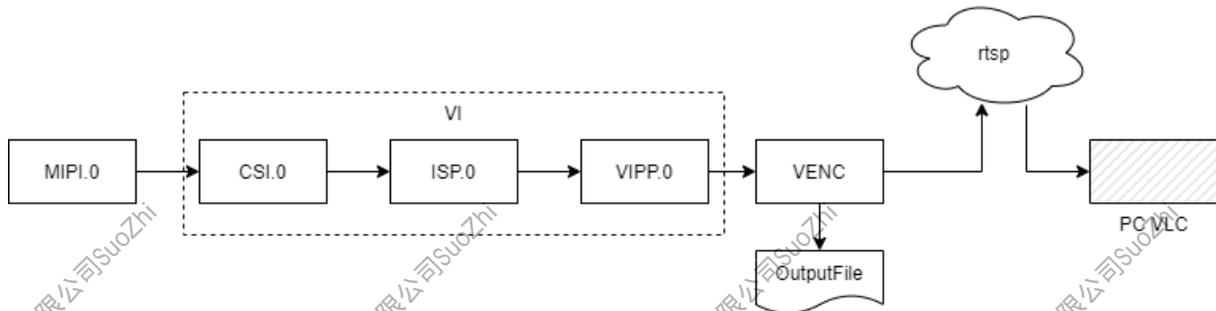


图 8-15: MPP_sample 测试通路-sample_rtsp

源文件：

无

目标文件：

AW_VirviEncoder.H264

参数配置：

主码流

main_vipp: 指定vipp设备号，默认0，表示vipp0。

main_src_width: 指定原始视频的宽度。

main_src_height: 指定原始视频的高度。

main_pixel_format: 指定像素格式（YUV类型：nv21(yvu420sp), yu12(yuv420p), yv12, nv12。LBC压缩类型：aw_lbc_2_0x, aw_lbc_2_5x, aw_lbc_1_5x, aw_lbc_1_0x）。

main_wdr_enable: 指定是否开启WDR，默认0，表示不开WDR。

main_vi_buf_num: 指定VI的buffer个数，仅对离线编码有效。

main_src_frame_rate: 指定vipp采集的帧率，单位：fps。

main_viChn: 指定vi虚通道号，默认0，表示vi ch0。配置为-1表示不测试主码流。

main_venc_chn: 指定ve通道号，默认0，表示ve ch0。配置为-1表示不测试主码流。

main_encode_type: 指定编码格式（H.264、H.265）。

main_encode_width: 指定生成的裸码流视频的宽度。

main_encode_height: 指定生成的裸码流视频的高度。

main_encode_frame_rate: 指定编码视频的帧率，单位：fps。

main_encode_bitrate: 指定编码的码率，单位：bps。

main_file_path: 指定编码裸码文件的存放路径。若不指定，则不会保存到文件。

main_online_en: 指定是否开启在线编码，默认0，表示不开启在线编码，即当前为离线编码。

main_online_share_buf_num: 指定在线编码的共享buffer个数（1或2），默认2，表示2个buffer。注意该配置仅当配置在线模式时才生效。

main_encpp_disable: 指定是否禁用encpp新通路，默认0，表示不禁用。

子码流

sub_vipp: 指定vipp设备号，默认1，表示vipp1（对于V853，表示vipp4）。

sub_vipp_crop_en: 指定是否开启vipp crop功能。默认1，表示开启。

sub_vipp_crop_rect_x: 指定vipp crop的x坐标。
 sub_vipp_crop_rect_y: 指定vipp crop的y坐标。
 sub_vipp_crop_rect_w: 指定vipp crop的宽度。
 sub_vipp_crop_rect_h: 指定vipp crop的高度。
 sub_src_width: 指定原始视频的宽度。
 sub_src_height: 指定原始视频的高度。
 sub_pixel_format: 指定像素格式 (YUV类型: nv21(yvu420sp), yu12(yuv420p), yv12, nv12。LBC压缩类型: aw_lbc_2_0x, aw_lbc_2_5x, aw_lbc_1_5x, aw_lbc_1_0x)。
 sub_wdr_enable: 指定是否开启WDR, 默认0, 表示不开WDR。
 sub_vi_buf_num: 指定VI的buffer个数, 仅对离线编码有效。
 sub_src_frame_rate: 指定vipp采集的帧率, 单位: fps。

sub_viChn: 指定vi虚通道号, 默认0, 表示vi ch0。配置为-1表示不测试子码流。
 sub_venc_chn: 指定ve通道号, 默认1, 表示ve ch1。配置为-1表示不测试子码流。
 sub_encode_type: 指定编码格式 (H.264、H.265)。
 sub_encode_width: 指定生成的裸码流视频的宽度。
 sub_encode_height: 指定生成的裸码流视频的高度。
 sub_encode_frame_rate: 指定编码视频的帧率, 单位: fps。
 sub_encode_bitrate: 指定编码的码率, 单位: bps。
 sub_file_path: 指定编码裸流文件的存放路径。若不指定, 则不会保存到文件。
 sub_encpp_disable: 指定是否禁用encpp新通路, 默认0, 表示不禁用。

缩时录影

lapse_viChn: 指定vi虚通道号, 默认1, 表示vi ch1。配置为-1表示不测试子码流。
 lapse_venc_chn: 指定ve通道号, 默认2, 表示ve ch2。配置为-1表示不测试子码流。
 lapse_encode_type: 指定编码格式 (H.264、H.265)。
 lapse_encode_width: 指定生成的裸码流视频的宽度。
 lapse_encode_height: 指定生成的裸码流视频的高度。
 lapse_encode_frame_rate: 指定编码视频的帧率, 单位: fps。
 lapse_encode_bitrate: 指定编码的码率, 单位: bps。
 lapse_file_path: 指定编码裸流文件的存放路径。若不指定, 则不会保存到文件。
 lapse_time: 指定缩时录影的帧间隔时间, 单位: ms。
 lapse_encpp_disable: 指定是否禁用encpp新通路, 默认0, 表示不禁用。

ISP2VE联动机制参数配置

isp_ve_linkage_enable: 指定是否开启ISP2VE联动机制。默认0, 表示不开启。
 isp_ve_linkage_test_stream: 指定ISP2VE联动机制应用到哪一路, 默认0, 表示主码流。(0:main stream, 1:sub stream, 2:lapse stream)
 ve2isp_param_update_interval: 指定ISP2VE联动参数更新的时间间隔, 单位: ms。

rtsp_net_type: 指定RTSP网络类型。默认1, 表示以太网。(0: "lo", 1: "eth0", 2: "br0", 3: "wlan0")

test_duration: 测试时间, 单位: s。0表示无限时长。

测试指令:

【Tina】

1. 将sample_rtsp和sample_rtsp.conf放到sd卡根目录。
2. 将sd卡挂载到机器上。
3. 执行测试指令

```
cd /mnt/extsd/
./sample_rtsp -path ./sample_rtsp.conf
```

 打印中会提示播放链接。

4. 在PC上使用VLC播放rtsp链接

```
rtsp://192.168.10.1:8554/ch0
rtsp://192.168.10.1:8554/ch1
```

【Melis】

暂不支持。

退出测试：

按“ctrl + c”退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 通过VLC 可以正常播放rtsp 链接。

8.1.16 sample_CodecParallel

测试目的：

演示同编同解。

组件依赖：

```
mpp_vi
mpp_venc
mpp_demuxer
mpp_vdec
mpp_muxer
mpp_hw_display
mpp_vo
```

测试通路：

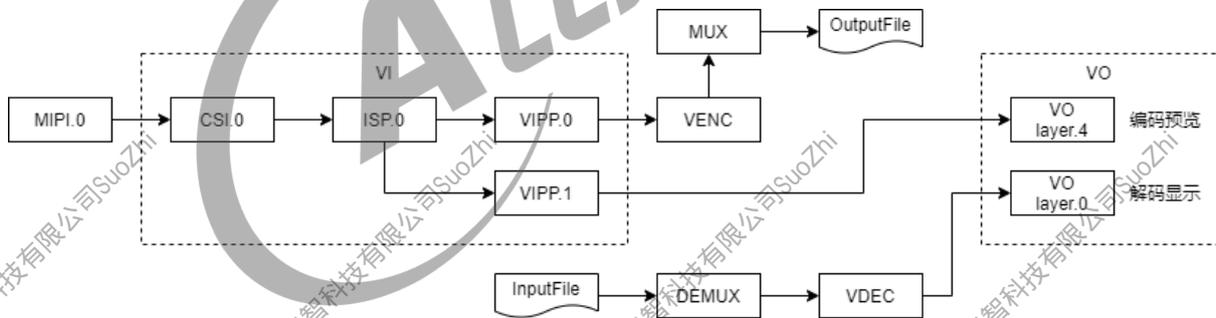


图 8-16: MPP_sample 测试通路-sample_CodecParallel

源文件：

/mnt/extsd/test.mp4

目标文件：

/mnt/extsd/1080p.mp4

参数配置：

```
# 通用配置
test_duration: 指定测试时间
```

```
# 录制配置
record_vipp_id: 指定VIPP设备号
record_vipp_format: 指定camera采集的图像格式
record_vipp_capture_width: 指定camera采集的图像宽度
record_vipp_capture_height: 指定camera采集的图像高度
record_vipp_frame_rate: 指定camera采集的帧率
record_dest_file: 指定录制文件的路径
record_video_file_format: 指定视频封装格式
record_video_frame_rate: 指定视频编码的帧率
record_video_bitrate: 指定视频编码的码率
record_video_width: 指定视频编码的图像宽度
record_video_height: 指定视频编码的图像高度
record_video_encoder_type: 指定视频编码格式
record_video_rc_mode: 指定视频编码的码率控制模式
record_video_duration: 指定视频编码的图像宽度

# 预览配置
preview_vipp_id: 指定VIPP设备号
preview_vipp_format: 指定camera采集的图像格式
preview_vipp_capture_width: 指定camera采集的图像宽度
preview_vipp_capture_height: 指定camera采集的图像高度
preview_vipp_frame_rate: 指定camera采集的帧率
preview_vo_dev_id: 指定VO设备号
preview_vo_display_x: 指定输出图像的X 轴位置
preview_vo_display_y: 指定输出图像的Y 轴位置
preview_vo_display_width: 指定输出图像宽度
preview_vo_display_height: 指定输出图像高度
preview_vo_layer_num: 指定图像显示的图层
preview_vo_disp_type: 指定显示设备 (hdmi, lcd, cvbs)

# 播放配置
play_src_file: 指定原始视频文件的路径
play_vo_dev_id: 指定VO设备号
play_vo_display_x: 指定输出图像的X 轴位置
play_vo_display_y: 指定输出图像的Y 轴位置
play_vo_display_width: 指定输出图像宽度
play_vo_display_height: 指定输出图像高度
play_vo_layer_num: 指定图像显示的图层
play_vo_disp_type: 指定显示设备 (hdmi, lcd, cvbs)
```

测试指令：

【Tina】

```
# cd /mnt/extsd/
# ./sample_CodecParallel -path ./sample_CodecParallel.conf
```

【Melis】

暂不支持。

结束测试：

1. 达到设定的测试时间后，自动退出测试。
2. 播放完test.mp4 文件后，自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. LCD屏幕一半显示预览画面，一半显示播放画面，同时会生成编码文件1080p.mp4。

8.1.17 sample_vdec

测试目的：

连续解码jpg格式图片或者h264裸码流，然后保存为yuv格式文件。

组件依赖：

mpp_vdec

测试通路：

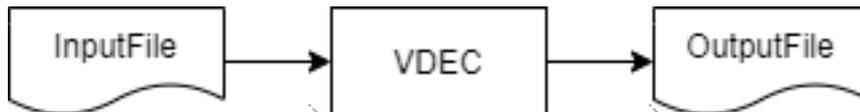


图 8-17: MPP_sample 测试通路-sample_vdec

源文件：

```
/mnt/extsd/vbs1.h264  
/mnt/extsd/vbs1.len  
/mnt/extsd/vbs2.h265  
/mnt/extsd/vbs2.len
```

目标文件：

```
/mnt/extsd/h264.yuv  
/mnt/extsd/h265.yuv
```

参数配置：

```
vdecoder_format: 指定解码格式，jpeg或者h264。  
yuv_file_path: 指定解码后生成的yuv文件的路径。  
jpeg_file_path: 指定原jpg图片路径，该文件必须为jpeg格式的文件。  
h264_vbs_path: 指定原264裸码流路径。  
h264_len_path: 指定原264裸码流帧大小描述文件路径。  
h265_vbs_path: 指定原265裸码流路径。  
h265_len_path: 指定原265裸码流帧大小描述文件路径。
```

测试指令：

【Tina】

```
# cd /mnt/extsd/  
# ./sample_vdec -path ./sample_vdec.conf
```

【Melis】

暂不支持。

退出测试：

文件解码完后会自动结束测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 使用YUVview 软件查看生成的YUV文件正常。

8.1.18 sample_demux2vdec

测试目的：

该sample演示从原始视频文件（如：xxx.mp4）中分离出视频数据帧并解码生成yuv文件。

组件依赖：

mpp_demuxer
mpp_vdec

测试通路：

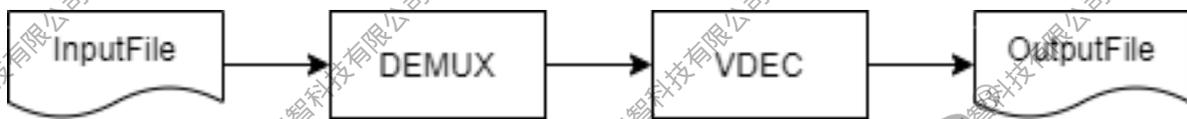


图 8-18: MPP_sample 测试通路-sample_demux2vdec

源文件：

/mnt/extsd/test.mp4

目标文件：

/mnt/extsd/y.yuv
/mnt/extsd/u.yuv
/mnt/extsd/v.yuv
/mnt/extsd/yuv.yuv

参数配置：

src_file: 指定原始视频文件的路径。
src_size: 指定原始视频文件的视频大小，如1080p。
seek_position: 指定原始视频文件的开始解析位置，单位：ms。
y_dst_file: 视频数据帧解码出来y数据分量。
u_dst_file: 视频数据帧解码出来u数据分量。
v_dst_file: 视频数据帧解码出来v数据分量。
yuv_dst_file: 视频数据帧解码出来对应的yuv数据文件。
test_duration: 测试时间，单位：s。

测试指令：

【Tina】

```
# cd /mnt/extsd/
# ./sample_demux2vdec -path ./sample_demux2vdec.conf
```

【Melis】

暂不支持。

退出测试：

测试完文件test.mp4，可以自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 使用YUVView 软件查看生成的YUV文件正常。

8.1.19 sample_demux2vdec_saveFrame

测试目的：

该sample演示从原始视频文件（如：264.mp4）中分离出视频数据帧并解码生成yuv文件。

组件依赖：

mpp_demuxer
mpp_vdec

测试通路：

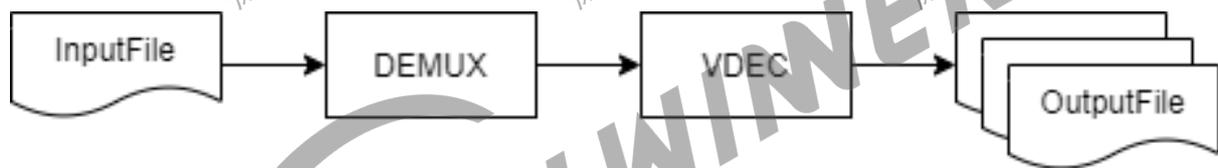


图 8-19: MPP_sample 测试通路-sample_demux2vdec_saveFrame

源文件：

/mnt/extsd/test.mp4

目标文件：

```
/mnt/extsd/sample_demux2vdec_saveFrame_Files/frame[1920x1088][00].nv21  
/mnt/extsd/sample_demux2vdec_saveFrame_Files/frame[1920x1088][01].nv21  
/mnt/extsd/sample_demux2vdec_saveFrame_Files/frame[1920x1088][02].nv21  
/mnt/extsd/sample_demux2vdec_saveFrame_Files/frame[1920x1088][03].nv21  
/mnt/extsd/sample_demux2vdec_saveFrame_Files/frame[1920x1088][04].nv21
```

参数配置：

src_file: 指定原始视频文件的路径。
seek_position: 指定原始视频文件的开始解析位置，单位：ms。
save_num: 保存视频帧的数量。
dst_dir: 保存视频帧的目录。

测试指令：

【Tina】

```
# cd /mnt/extsd/
# ./sample_demux2vdec_saveFrame -path ./sample_demux2vdec_saveFrame.conf
```

【Melis】

暂不支持。

退出测试：

测试完文件test.mp4，可以自动退出测试。测试时间会比较长，可以按“ctrl + c”退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 使用YUVView 软件查看生成的YUV文件正常。

8.1.20 sample_demux2vdec2vo

测试目的：

该sample演示解码视频文件（如：xxx.mp4）并从LCD上显示出来。

组件依赖：

```
mpp_demuxer
mpp_vdec
mpp_hw_display
mpp_vo
```

测试通路：

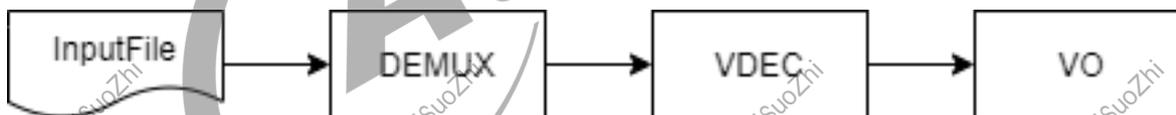


图 8-20: MPP_sample 测试通路-sample_demux2vdec2vo

源文件：

/mnt/extsd/test.mp4

目标文件：

无

参数配置：

```
src_file: 指定原始视频文件的路径。
seek_position: 指定原始视频文件的开始解析位置，单位：ms。
test_duration: 测试时间，单位：s。
display_x: 显示区域的x坐标。
display_y: 显示区域的y坐标。
display_width: 显示区域的宽度。
display_height: 显示区域的高度。
```

ve_freq: 指定VE的频率，默认0，表示300MHz，单位：MHz。

测试指令：

【Tina】

```
# cd /mnt/extsd/  
# ./sample_demux2vdec2vo -path ./sample_demux2vdec2vo.conf
```

【Melis】

```
# sample_demux2vdec2vo -path ./mnt/E/sample_demux2vdec2vo.conf
```

退出测试：

测试完文件test.mp4，可以自动退出测试。可以按“ctrl + c”退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. LCD 显示视频播放正常。

8.1.21 sample_vencQpMap

测试目的：

演示调用VENC MPI 接口，测试视频编码QpMap模式。

从yuv原始数据文件xxx.yuv中读取视频帧，编码，将取得的编码往输出文件里面直接写，生成裸码流视频文件。开启QpMap模式下，在编码完一帧后到编码下一帧前，获取上一帧（刚已编码完成）的编码统计信息，用户需要自己处理这个信息，然后设定下一帧（即将送去编码）的编码方式（QP、skip和en等）。如果是H264或H265编码sample会自动在生成文件的开始加上spspps信息，其他格式则不加。QpMap模式只有H264和H265两种编码格式支持。

组件依赖：

mpp_venc

测试通路：

无

源文件：

/mnt/extsd/test.yuv

目标文件：

/mnt/extsd/test.raw

参数配置：

test_duration: 指定测试时间。

src_file: 指定原始yuv文件的路径。

src_width: 指定原始视频文件的宽度。

src_height: 指定原始视频文件的高度。

src_pixfmt: 指定原始视频图像的像素格式。
 color_space: 指定颜色空间 (jpeg, rec709, rec709_part_range)。
 dst_file: 指定生成的裸码流视频文件路径。
 dest_width: 指定视频编码输出图像的宽度。
 dest_height: 指定视频编码输出图像的高度。
 dest_pixfmt: 指定视频编码输出图像的像素格式。

encoder: 指定视频编码的类型。
 profile: 指定视频编码质量等级。
 framerate: 指定视频编码的帧率。
 bitrate: 指定视频编码的码率。

测试指令:

【Tina】

```
# cd /mnt/extsd/
# ./sample_vencQpMap -path ./sample_vencQpMap.conf
```

【Melis】

暂不支持。

结束测试:

达到设定的测试时间后, 自动退出测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 使用VLC 软件播放生成的raw文件正常。

8.1.22 sample_OnlineVenc

测试目的:

该sample演示IPC在线编码使用场景: 主码流在线编码、子码流离线编码。
 针对主码流和子码流分别创建mpi_vi和mpi_venc, 将它们绑定, 再分别启动。mpi_vi采集图像, 直接传输给mpi_venc进行编码。

组件依赖:

mpp_venc

测试通路:

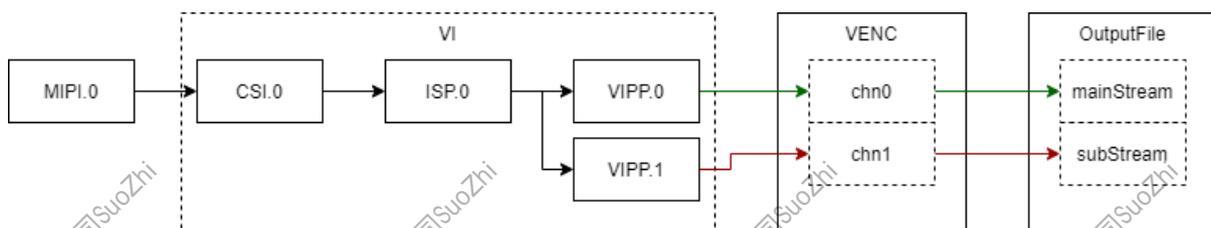


图 8-21: MPP_sample 测试通路-sample_OnlineVenc

源文件：

无

目标文件：

```
/mnt/extsd/mainStream.raw  
/mnt/extsd/subStream.raw  
/mnt/extsd/lapseStream.raw
```

参数配置：

主码流

main_vipp: 指定vipp设备号, 默认0, 表示vipp0。
main_src_width: 指定原始视频的宽度。
main_src_height: 指定原始视频的高度。
main_pixel_format: 指定像素格式 (YUV类型: nv21(yvu420sp), yu12(yuv420p), yv12, nv12。LBC压缩类型: aw_lbc_2_0x, aw_lbc_2_5x, aw_lbc_1_5x, aw_lbc_1_0x)。
main_wdr_enable: 指定是否开启WDR, 默认0, 表示不开WDR。
main_vi_buf_num: 指定VI的buffer个数, 仅对离线编码有效。
main_src_frame_rate: 指定vipp采集的帧率, 单位: fps。
main_viChn: 指定vi虚通道号, 默认0, 表示vi ch0。配置为-1表示不测试主码流。
main_venc_chn: 指定ve通道号, 默认0, 表示ve ch0。配置为-1表示不测试主码流。
main_encode_type: 指定编码格式 (H.264、H.265)。
main_encode_width: 指定生成的裸码流视频的宽度。
main_encode_height: 指定生成的裸码流视频的高度。
main_encode_frame_rate: 指定编码视频的帧率, 单位: fps。
main_encode_bitrate: 指定编码的码率, 单位: bps。
main_file_path: 指定编码裸流文件的存放路径。若不指定, 则不会保存到文件。
main_online_en: 指定是否开启在线编码, 默认0, 表示不开启在线编码, 即当前为离线编码。
main_online_share_buf_num: 指定在线编码的共享buffer个数 (1或2), 默认2, 表示2个buffer。注意该配置仅当配置在线模式时才生效。
main_encpp_disable: 指定是否禁用encpp新通路, 默认0, 表示不禁用。

子码流

sub_vipp: 指定vipp设备号, 默认1, 表示vipp1 (对于V853, 表示vipp4)。
sub_vipp_crop_en: 指定是否开启vipp crop功能。默认1, 表示开启。
sub_vipp_crop_rect_x: 指定vipp crop的x坐标。
sub_vipp_crop_rect_y: 指定vipp crop的y坐标。
sub_vipp_crop_rect_w: 指定vipp crop的宽度。
sub_vipp_crop_rect_h: 指定vipp crop的高度。
sub_src_width: 指定原始视频的宽度。
sub_src_height: 指定原始视频的高度。
sub_pixel_format: 指定像素格式 (YUV类型: nv21(yvu420sp), yu12(yuv420p), yv12, nv12。LBC压缩类型: aw_lbc_2_0x, aw_lbc_2_5x, aw_lbc_1_5x, aw_lbc_1_0x)。
sub_wdr_enable: 指定是否开启WDR, 默认0, 表示不开WDR。
sub_vi_buf_num: 指定VI的buffer个数, 仅对离线编码有效。
sub_src_frame_rate: 指定vipp采集的帧率, 单位: fps。

sub_viChn: 指定vi虚通道号, 默认0, 表示vi ch0。配置为-1表示不测试子码流。
sub_venc_chn: 指定ve通道号, 默认1, 表示ve ch1。配置为-1表示不测试子码流。
sub_encode_type: 指定编码格式 (H.264、H.265)。
sub_encode_width: 指定生成的裸码流视频的宽度。
sub_encode_height: 指定生成的裸码流视频的高度。
sub_encode_frame_rate: 指定编码视频的帧率, 单位: fps。
sub_encode_bitrate: 指定编码的码率, 单位: bps。
sub_file_path: 指定编码裸流文件的存放路径。若不指定, 则不会保存到文件。
sub_encpp_disable: 指定是否禁用encpp新通路, 默认0, 表示不禁用。

缩时录影

lapse_viChn: 指定vi虚通道号, 默认1, 表示vi ch1。配置为-1表示不测试子码流。
 lapse_venc_chn: 指定ve通道号, 默认2, 表示ve ch2。配置为-1表示不测试子码流。
 lapse_encode_type: 指定编码格式 (H.264、H.265)。
 lapse_encode_width: 指定生成的裸码流视频的宽度。
 lapse_encode_height: 指定生成的裸码流视频的高度。
 lapse_encode_frame_rate: 指定编码视频的帧率, 单位: fps。
 lapse_encode_bitrate: 指定编码的码率, 单位: bps。
 lapse_file_path: 指定编码裸流文件的存放路径。若不指定, 则不会保存到文件。
 lapse_time: 指定缩时录影的帧间隔时间, 单位: ms。
 lapse_encpp_disable: 指定是否禁用encpp新通路, 默认0, 表示不禁用。

ISP2VE联动机制参数配置

isp_ve_linkage_enable: 指定是否开启ISP2VE联动机制。默认0, 表示不开启。
 isp_ve_linkage_test_stream: 指定ISP2VE联动机制应用到哪一路, 默认0, 表示主码流。(0:main stream, 1:sub stream, 2:lapse stream)
 ve2isp_param_update_interval: 指定ISP2VE联动参数更新的时间间隔, 单位: ms。

wb_yuv_enable: 指定是否开启存储回写yuv功能。默认0, 表示不开启。
 wb_yuv_buf_num: 指定存储回写yuv的buffer个数, 单位: 一帧图像的大小。
 wb_yuv_start_index: 指定存储回写yuv从第几帧开始保存。
 wb_yuv_total_cnt: 指定存储回写yuv一共保存多少帧。
 wb_yuv_stream_channel: 指定存储回写yuv的编码通道号, 默认0, 表示主码流。(0:main stream, 1:sub stream, 2:sub lapse stream)
 wb_yuv_file_path: 指定存储回写yuv的文件路径。

test_duration: 测试时间, 单位: s。0表示无限时长。

测试指令:

[Tina]

```
# cd /mnt/extsd/
# ./sample_OnlineVenc -path ./sample_OnlineVenc.conf
```

[Melis]

暂不支持。

结束测试:

达到设定的测试时间后, 自动退出测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 使用PC软件VLC播放测试生成的视频文件正常。
3. 使用MediaInfo软件检查编码参数符合预期。
4. 观察实际帧率打印, 主码流和子码流均符合预期。

8.1.23 sample_vencGdcZoom

测试目的:

演示编码GDC数字变焦功能。

组件依赖:

mpp_venc

测试通路：

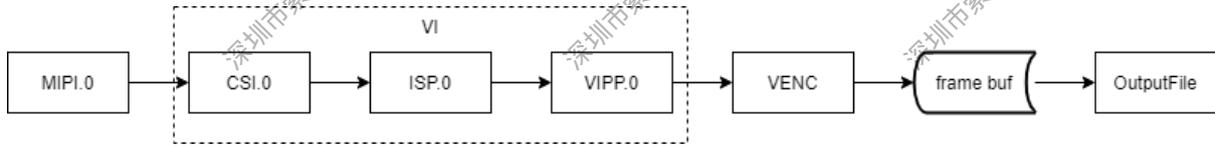


图 8-22: MPP_sample 测试通路-sample_vencGdcZoom

源文件：

无

目标文件：

/mnt/extsd/test.raw

参数配置：

online_en: 指定是否开启在线编码，默认0，表示不开启在线编码，即当前为离线编码。
online_share_buf_num: 指定在线编码的共享buffer个数（1或2），默认2，表示2个buffer。注意该配置仅当配置在线模式时才生效。

vipp_id: 指定vipp设备号，默认0，表示vipp0。
wdr_en: 指定是否开启WDR，默认0，表示不开WDR。

drop_frm_num: 指定丢帧个数，对于离线编码是vipp丢帧，对于在线编码是ve编码前丢帧。

src_width: 指定原始视频的宽度。
src_height: 指定原始视频的高度。

vi_buffer_num: 指定CSI的buffer个数，仅对离线编码有效。
saturation_change: 指定饱和度改变的值，默认0，表示不调整。取值范围[-256, 512]。

src_pixfmt: 指定像素格式（YUV类型：nv21(yuv420sp), yu12(yuv420p), yv12, nv12。LBC压缩类型：
 aw_lbc_2_0x, aw_lbc_2_5x, aw_lbc_1_5x, aw_lbc_1_0x)。
color_space: 指定颜色空间（jpeg, rec709, rec709_part_range）。

venc_ch_id: 指定编码通道，默认0，表示venc ch0。

video_dst_file: 指定编码文件的存放路径。
video_framerate: 指定帧率，单位：fps。
video_bitrate: 指定编码的码率，单位：bps。

video_width: 指定生成的裸码流视频的宽度。
video_height: 指定生成的裸码流视频的高度。

video_encoder: 指定编码格式（H.264、H.265、MJPEG）。
profile: 指定编码质量，对于H264，建议配置High（2），对于H265，建议配置Main（0）。

ve_freq: 指定VE的频率，默认0，表示300MHz，单位：MHz。

product_mode: 指定产品类型（0:Normal, 1:IPC）。
sensor_type: 指定sensor类型（0:DisWdr, 1:EnWdr）。
key_frame_interval: 指定I帧间隔。

enable_gdc: 指定是否开启GDC功能, 默认1, 表示开启。

enable_gdc_zoom: 指定是否开启GDC zoom功能, 默认1, 表示开启。

rc_mode: 指定码率控制模式 (0:CBR, 1:VBR, 2:FIXQP, 3:QPMAP)。

VBR、CBR模式, 编码参数

init_qp: 指定初始QP值, 取值范围(0, 51)。

min_i_qp: 指定I帧最小QP值, 取值范围(0, 51)。FIXQP模式时, 取其值为i_qp。

max_i_qp: 指定I帧最大QP值, 取值范围(0, 51)。

min_p_qp: 指定P帧最小QP值, 取值范围(0, 51)。FIXQP模式时, 取其值为p_qp。

max_p_qp: 指定P帧最大QP值, 取值范围(0, 51)。

mb_qp_limit_en: 指定是否开启mb qp限制, 默认0, 表示不开启。

VBR模式, 编码参数

moving_th: 指定mb16x16中MAD的阈值, 取值范围[0, 31]。

quality: 指定静态P帧的位系数, 取值范围[1, 20]。

p_bits_coef: 指定移动P帧的位系数, 取值范围[1, 50]。

i_bits_coef: 指定I帧的位系数, 取值范围[1, 20]。

gop_mode: 指定GOP模式 (0:NormalP, 1:DualP, 2:SmartP)。

gop_size: 指定GOP大小, 当前只针对H265有效, 取值范围[1, 63]。

高级跳帧参数

AdvancedRef_Base: 设置该值大于0表示开启高级跳帧功能, 等于0表示关闭高级跳帧功能。

AdvancedRef_Enhance: 设置Enhance为5。

AdvancedRef_RefBaseEn: 设置该值为1表示开启参考帧, 等于0表示关闭参考帧。

enable_fast_enc: 指定是否开启快速编码, 默认0, 表示不开启。

encode_rotate: 指定编码旋转角度 (0, 90, 180, 270), 顺时针方向。

mirror: 指定编码镜像是否开启, 默认0, 表示不开启。取值范围[0, 1]。

video_duration: 指定生成一个视频文件的最大持续时间 (如: 每个视频文件长度一分钟), 单位: s。

test_duration: 指定测试时间, 单位: s。

color2grey: 指定是否开启彩转灰功能, 默认"no", 表示不开启。取值范围["no", "yes"]。

2D降噪参数配置

2dnr_en: 编码器2DNR使能, 默认1, 表示开启。取值范围[0, 1]。

2dnr_strength_y: 亮度降噪强度系数。该值越大, 滤波强度越高。默认127, 取值范围[0, 255]。

2dnr_strength_c: 色度降噪强度系数。该值越大, 滤波强度越高。默认127, 取值范围[0, 255]。

2dnr_threshold_y: 亮度邻域像素降噪开关阈值。该值越大, 越容易触发2d滤波。默认7, 取值范围[0, 15]。

2dnr_threshold_c: 色度邻域像素降噪开关阈值。该值越大, 越容易触发2d滤波。默认7, 取值范围[0, 15]。

3D降噪参数配置

3dnr_en: 编码器3DNR使能, 默认1, 表示开启。取值范围[0, 1]。

3dnr_pix_level_en: 3d滤波权重系数自适应修正开关。默认0, 表示不开启。取值范围[0, 1]。

3dnr_smooth_en: 像素级3x3平滑滤波使能。默认1, 表示开启。取值范围[0, 1]。

3dnr_pix_diff_th: 仅当3d_adjust_pix_level_enable为1时生效, 自适应修正幅度阈值。该值越大, 滤波强度越高。默认6, 取值范围[0, 31]。

3dnr_max_mv_th: 源图像素块运动矢量最大阈值, 单个像素块仅当其水平和垂直MV皆小于该阈值, 才会进行3d滤波。该值越大, 越容易触发3d滤波。默认2, 取值范围[0, 63]。

3dnr_max_mad_th: 源图与参考图之间像素块的MAD最大阈值, 单个像素块仅当其MAD值小于该阈值, 才会进行3d滤波。该值越大, 越容易触发3d滤波。默认11, 取值范围[0, 63]。

3dnr_min_coef: 3d滤波权重系数的最小阈值。该值越小, 自适应3d滤波的强度上限越高。默认14, 取值范围[0, 3dnr_max_coef]。

3dnr_max_coef: 3d滤波权重系数的最大阈值。该值越大, 自适应3d滤波的强度下限越低。默认16, 取值范围[3dnr_min_coef, 16]。

ROI测试参数配置

roi_num: 指定ROI的个数, 默认0, 表示不开启ROI测试。取值范围[0, 8]。
roi_qp: 指定ROI区域的QP值, 默认50, 表示最差质量。取值范围(0, 51)。
roi_BgFrameRateEnable: 指定是否开启非ROI区域低帧率编码, 默认0, 表示不开启。
roi_BgFrameRateAttenuation: 指定非ROI区域的帧率衰减比例, 默认3, 表示非ROI区域的帧率是正常帧率的1/3。

IntraRefresh_BlockNum: 指定P帧帧内刷新的block个数, 默认0, 表示不开启P帧帧内刷新功能。
orl_num: 指定ORL(Object Rectangle Label)的个数, 取值范围[0, 16]。

配置vbv buffer大小和vbv buffer阈值大小

vbvBufferSize: 指定vbv buffer大小, 默认0, 表示由mpp middleware自行计算并设置给编码器, 大于0, 表示由app设置。

vbvThreshSize: 指定vbv buffer阈值大小, 默认0, 表示由mpp middleware自行计算并设置给编码器, 大于0, 表示由app设置。

裁剪编码参数设置

crop_en: 指定是否开启裁剪编码, 默认0, 表示不开启。

crop_rect_x: 指定裁剪区域的x坐标。

crop_rect_y: 指定裁剪区域的y坐标。

crop_rect_w: 指定裁剪区域的宽度。

crop_rect_h: 指定裁剪区域的高度。

vui_timing_info_present_flag: 指定是否添加VUI信息, 默认0, 表示不添加。

测试指令:

[Tina]

```
# cd /mnt/extsd/  
# ./sample_vencGdcZoom -path ./sample_vencGdcZoom.conf
```

[Melis]

暂不支持。

结束测试:

达到设定的测试时间后, 自动退出测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 使用PC软件VLC播放测试生成的视频文件正常。注意观察视频是否有缩放效果。
3. 使用MediaInfo软件检查编码参数符合预期。

8.1.24 sample_takePicture

测试目的:

演示拍照功能。支持通过配置开启单拍和连拍测试。

组件依赖:

```
mpp_vi  
mpp_venc
```

测试通路:

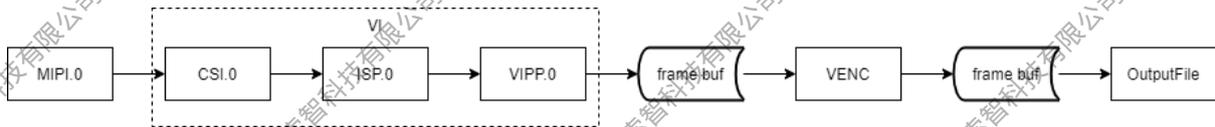


图 8-23: MPP_sample 测试通路-sample_takePicture

源文件:

无

目标文件:

照片文件, 单拍: 图片编号pic[n][0].jpg, 连拍: 图片编号pic[0][n].jpg

参数配置:

dev_num: 指定VI Dev设备节点。
 src_width: 指定camera采集的图像宽度。
 src_height: 指定camera采集的图像高度。
 frame_rate: 指定camera采集图像的帧率。
 src_format: 指定camera采集图像像素格式。
 color_space: 指定颜色空间类型。
 drop_frm_num: 指定VIPP丢帧的帧数。

take_picture_mode: 拍照模式 (0: 不拍, 1: 单拍, 2: 连拍)。
 take_picture_num: 拍照的图片张数。
 take_picture_interval: 拍照时间间隔, 单位: ms。
 jpeg_width: jpeg拍照图片的宽度。
 jpeg_height: jpeg拍照图片的高度。
 store_dir: jpeg拍照文件的存储路径 (单拍: 图片编号pic[n][0], 连拍: 图片编号pic[0][n])。

测试指令:

【Tina】

```
# cd /mnt/extsd/  
# ./sample_takePicture -path ./sample_takePicture.conf
```

【Melis】

暂不支持。

结束测试:

达到设定的测试时间后, 自动退出测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 使用PC图片查看工具检查拍照图片显示正常。
单拍模式下, 拍照后, 图片的编号是: pic[0][0]、pic[1][0]、pic[1][0] ... pic[n][0]。
3. 使用MediaInfo软件检查编码参数符合预期。
检查拍照的图片格式和分辨率:
格式: JPEG
宽度: 1 920 像素
高度: 1 080 像素

8.1.2.5 sample_recorder

测试目的：

支持最大四路录制编码封装或者预览显示。
每路创建一个vip和虚通道，根据参数的不同与编码器绑定测试录制编码，与VO绑定测试预览显示。

组件依赖：

mpp_vi
mpp_venc
mpp_vo

测试通路：

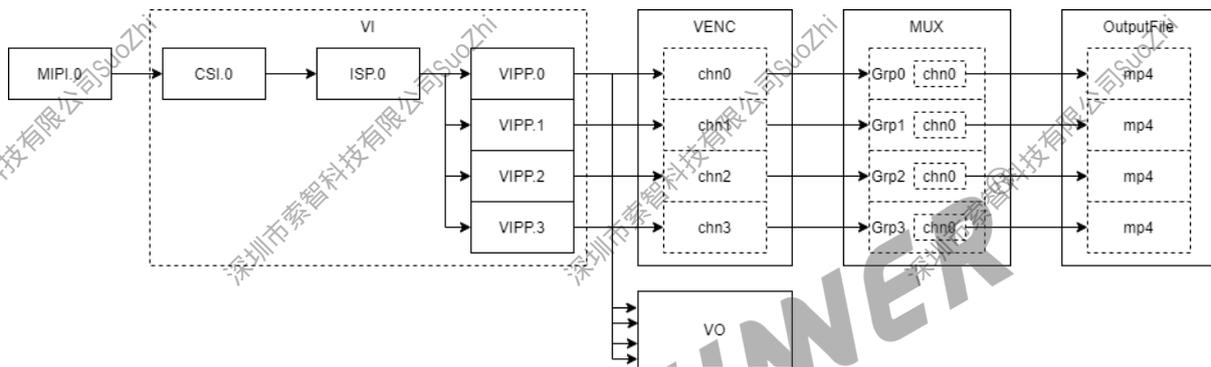


图 8-24: MPP_sample 测试通路-sample_recorder

源文件：

无

目标文件：

recorder1_1080p@20.mp4
recorder2_1080p@20.mp4
recorder3_1080p@20.mp4
recorder4_1080p@20.mp4

参数配置：

最多支持4路录像和预览，以下是第1路的参数配置，其他路与其类似。

- recorder1_vi_dev: VIPP设备号。
- recorder1_isp_dev: ISP设备号。
- recorder1_cap_width: 录制分辨率宽度。
- recorder1_cap_height: 录制分辨率高度。
- recorder1_cap_frmrate: 录制帧率。
- recorder1_cap_format: 录制视频格式。
- recorder1_vi_bufnum: 配置VIPP设备buffer数量。
- recorder1_enable_WDR: 是否启用WDR模式(0: 不启用 1: 启用)。
- recorder1_enc_online: 是否启用在线编码模式(0: 不启用 1: 启用)。
- recorder1_enc_online_share_bufnum: 在线编码共享buffer个数。
- recorder1_enc_type: 编码器类型。
- recorder1_enc_width: 编码视频分辨率宽度。
- recorder1_enc_height: 编码视频分辨率高度。

recorder1_enc_frmrate: 编码视频帧率。
recorder1_enc_bitrate: 编码视频码率。
recorder1_enc_rcmode: 编码码率控制模式。
recorder1_disp_x: 显示区域X坐标。
recorder1_disp_y: 显示区域y坐标。
recorder1_disp_width: 显示区域宽度。
recorder1_disp_height: 显示区域高度。
recorder1_disp_dev: 显示设备类型。
recorder1_rec_duration: 视频文件录制时长。
recorder1_rec_file_cnt: 视频文件最大分片数。
recorder1_rec_file: 视频文件保存路径。

参数配置说明:

1. 支持录制格式: nv21、yv12、nv12、yu12、aw_fbc、aw_lbc_2_0x、aw_lbc_2_5x、aw_lbc_1_5x、aw_lbc_1_0x
 2. 支持编码格式: H.264、H.265、MJPEG
 3. 支持封装格式: MP4、TS
 4. 特殊参数配置
 - 将recorder1_vi_dev设置为 -1 , 表示不启用这组配置测试。
 - 将recorder1_enc_width 或者 recorder1_enc_height设置为 0, 表示不测试这组配置的视频编码与封装。
 - 将recorder1_disp_width 或 recorder1_disp_height设置为0, 表示不测试这组配置的预览显示。
- 视频编码与预览显示不可在同一组配置中同时使用。

测试指令:

【Tina】

```
# cd /mnt/extsd/  
# ./sample_recorder -path ./sample_recorder.conf
```

【Melis】

暂不支持。

结束测试:

达到设定的测试时间后, 自动退出测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 使用PC软件VLC播放测试生成的视频文件正常。
3. 使用MediaInfo软件检查编码参数符合预期。

8.1.26 sample_vencRecreate

测试目的:

演示动态配置编码格式、帧率、码率等功能。
该sample测试mpi_vi和mpi_venc组件的绑定组合。创建mpi_vi和mpi_venc, 将它们绑定, 再分别启动。
mpi_vi采集图像, 直接传输给mpi_venc进行编码。

组件依赖:

无

测试通路:

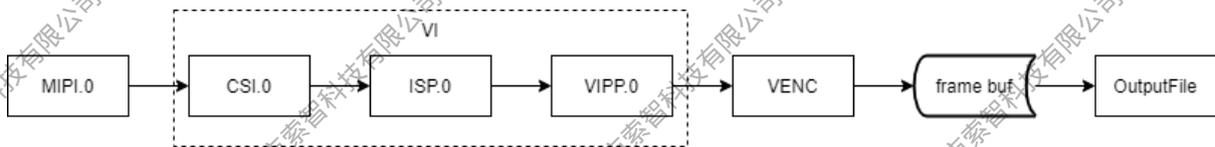


图 8-25: MPP_sample 测试通路-sample_vencRecreate

源文件:

无

目标文件:

/mnt/extsd/test.raw

参数配置:

online_en: 指定是否开启在线编码，默认0，表示不开启在线编码，即当前为离线编码。
online_share_buf_num: 指定在线编码的共享buffer个数（1或2），默认2，表示2个buffer。注意该配置仅当配置在线模式时才生效。

vipp_id: 指定vipp设备号，默认0，表示vipp0。
wdr_en: 指定是否开启WDR，默认0，表示不开WDR。

drop_frm_num: 指定丢帧个数，对于离线编码是vipp丢帧，对于在线编码是ve编码前丢帧。

src_width: 指定原始视频的宽度。
src_height: 指定原始视频的高度。

vi_buffer_num: 指定CSI的buffer个数，仅对离线编码有效。
saturation_change: 指定饱和度改变的值，默认0，表示不调整。取值范围[-256, 512]。

src_pixfmt: 指定像素格式（YUV类型：nv21(yuv420sp), yu12(yuv420p), yv12, nv12。LBC压缩类型：aw_lbc_2_0x, aw_lbc_2_5x, aw_lbc_1_5x, aw_lbc_1_0x）。

color_space: 指定颜色空间（jpeg, rec709, rec709_part_range）。

venc_ch_id: 指定编码通道，默认0，表示venc ch0。

video_dst_file: 指定编码文件的存放路径。
video_framerate: 指定帧率，单位：fps。
video_bitrate: 指定编码的码率，单位：bps。

video_width: 指定生成的裸码流视频的宽度。
video_height: 指定生成的裸码流视频的高度。

video_encoder: 指定编码格式（H.264、H.265、MJPEG）。
profile: 指定编码质量，对于H264，建议配置High（2），对于H265，建议配置Main（0）。

ve_freq: 指定VE的频率，默认0，表示300MHz，单位：MHz。

product_mode: 指定产品类型（0:Normal, 1:IPC）。
sensor_type: 指定sensor类型（0:DisWdr, 1:EnWdr）。
key_frame_interval: 指定I帧间隔。

enable_gdc: 指定是否开启GDC功能，默认0，表示不开GDC。
rc_mode: 指定码率控制模式（0:CBR, 1:VBR, 2:FIXQP, 3:QPMAP）。

VBR、CBR模式，编码参数

init_qp: 指定初始QP值, 取值范围(0, 51)。
min_i_qp: 指定I帧最小QP值, 取值范围(0, 51)。FIXQP模式时, 取其值为i_qp。
max_i_qp: 指定I帧最大QP值, 取值范围(0, 51)。
min_p_qp: 指定P帧最小QP值; 取值范围(0, 51)。FIXQP模式时, 取其值为p_qp。
max_p_qp: 指定P帧最大QP值, 取值范围(0, 51)。
mb_qp_limit_en: 指定是否开启mb_qp限制, 默认0, 表示不开启。

VBR模式, 编码参数

moving_th: 指定mb16x16中MAD的阈值, 取值范围[0, 31]。
quality: 指定静态P帧的位系数, 取值范围[1, 20]。
p_bits_coef: 指定移动P帧的位系数, 取值范围[1, 50]。
i_bits_coef: 指定I帧的位系数, 取值范围[1, 20]。

gop_mode: 指定GOP模式 (0:NormalP, 1:DualP, 2:SmartP)。
gop_size: 指定GOP大小, 当前只针对H265有效, 取值范围[1, 63]。

高级跳帧参数

AdvancedRef_Base: 设置该值大于0表示开启高级跳帧功能, 等于0表示关闭高级跳帧功能。
AdvancedRef_Enhance: 设置Enhance为5。
AdvancedRef_RefBaseEn: 设置该值为1表示开启参考帧, 等于0表示关闭参考帧。
enable_fast_enc: 指定是否开启快速编码, 默认0, 表示不开启。

encode_rotate: 指定编码旋转角度 (0, 90, 180, 270), 顺时针方向。
mirror: 指定编码镜像是否开启, 默认0, 表示不开启。取值范围[0, 1]。
color2grey: 指定是否开启彩转灰功能, 默认"no", 表示不开启。取值范围["no", "yes"]。

2D降噪参数配置

2dnr_en: 编码器2DNR使能, 默认1, 表示开启。取值范围[0, 1]。
2dnr_strength_y: 亮度降噪强度系数。该值越大, 滤波强度越高。默认127, 取值范围[0, 255]。
2dnr_strength_c: 色度降噪强度系数。该值越大, 滤波强度越高。默认127, 取值范围[0, 255]。
2dnr_threshold_y: 亮度邻域像素降噪开关阈值。该值越大, 越容易触发2d滤波。默认7, 取值范围[0, 15]。
2dnr_threshold_c: 色度邻域像素降噪开关阈值。该值越大, 越容易触发2d滤波。默认7, 取值范围[0, 15]。

3D降噪参数配置

3dnr_en: 编码器3DNR使能, 默认1, 表示开启。取值范围[0, 1]。
3dnr_pix_level_en: 3d滤波权重系数自适应修正开关。默认0, 表示不开启。取值范围[0, 1]。
3dnr_smooth_en: 像素级3x3平滑滤波使能。默认1, 表示开启。取值范围[0, 1]。
3dnr_pix_diff_th: 仅当3d_adjust_pix_level_enable为1时生效, 自适应修正幅度阈值。该值越大, 滤波强度越高。默认6, 取值范围[0, 31]。
3dnr_max_mv_th: 源图像素块运动矢量最大阈值, 单个像素块仅当其水平和垂直MV皆小于该阈值, 才会进行3d滤波。该值越大, 越容易触发3d滤波。默认2, 取值范围[0, 63]。
3dnr_max_mad_th: 源图与参考图之间像素块的MAD最大阈值, 单个像素块仅当其MAD值小于该阈值, 才会进行3d滤波。该值越大, 越容易触发3d滤波。默认11, 取值范围[0, 63]。
3dnr_min_coef: 3d滤波权重系数的最小阈值。该值越小, 自适应3d滤波的强度上限越高。默认14, 取值范围[0, 3dnr_max_coef]。
3dnr_max_coef: 3d滤波权重系数的最大阈值。该值越大, 自适应3d滤波的强度下限越低。默认16, 取值范围[3dnr_min_coef, 16]。

ROI测试参数配置

roi_num: 指定ROI的个数, 默认0, 表示不开启ROI测试。取值范围[0, 8]。
roi_qp: 指定ROI区域的QP值, 默认50, 表示最差质量。取值范围(0, 51)。
roi_BgFrameRateEnable: 指定是否开启非ROI区域低帧率编码, 默认0, 表示不开启。
roi_BgFrameRateAttenuation: 指定非ROI区域的帧率衰减比例, 默认3, 表示非ROI区域的帧率是正常帧率的1/3。

IntraRefresh_BlockNum: 指定P帧帧内刷新的block个数, 默认0, 表示不开启P帧帧内刷新功能。
orl_num: 指定ORL(Object Rectangle Label)的个数, 取值范围[0, 16]。

配置vbv buffer大小和vbv buffer阈值大小

vbvBufferSize: 指定vbv buffer大小, 默认0, 表示由mpp middleware自行计算并设置给编码器, 大于0, 表示由

app设置。

vbvThreshSize: 指定vbv buffer阈值大小, 默认0, 表示由mpp middleware自行计算并设置给编码器, 大于0, 表示由app设置。

裁剪编码参数设置

crop_en: 指定是否开启裁剪编码, 默认0, 表示不开启。

crop_rect_x: 指定裁剪区域的x坐标。

crop_rect_y: 指定裁剪区域的y坐标。

crop_rect_w: 指定裁剪区域的宽度。

crop_rect_h: 指定裁剪区域的高度。

vui_timing_info_present_flag: 指定是否添加VUI信息, 默认0, 表示不添加。

normal_test_duration: 指定正常测试时间, 单位: s。

编码recreate参数配置

venc_recreate_enable: 指定是否开启编码recreate, 默认1, 表示开启。

dynamic_video_encoder: 指定编码格式 (H.264、H.265)。

dynamic_video_framerate: 指定帧率, 单位: fps。

dynamic_video_bitrate: 指定编码的码率, 单位: bps。

dynamic_video_width: 指定生成的裸码流视频的宽度。

dynamic_video_height: 指定生成的裸码流视频的高度。

dynamic_key_frame_interval: 指定I帧间隔。

dynamic_rc_mode: 指定码率控制模式 (0:CBR, 1:VBR)。

venc_recreate_test_duration: 指定recreate测试时间, 单位: s。

测试指令:

[Tina]

```
# cd /mnt/extsd/  
# ./sample_vencRecreate -path ./sample_vencRecreate.conf
```

[Melis]

不支持

退出测试:

测试达到设定的时间后自动退出测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 使用VLC 软件播放生成的raw文件正常。

8.2 音频

8.2.1 sample_ai

测试目的:

根据配置参数采集对应的pcm数据, 并写入到文件中保存。

组件依赖：

mpp_aino

测试通路：

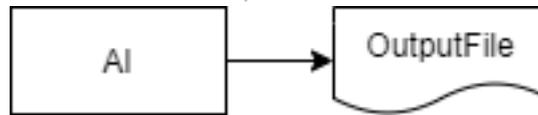


图 8-26: MPP_sample 测试通路-sample_ai

源文件：

无

目标文件：

./mnt/extsd/test.wav

参数配置：

pcm_file_path: 指定目标pcm文件的路径, 该文件是包含wave头的wav格式文件。
pcm_sample_rate: 指定采样率, 通常设置为8000。
pcm_channel_cnt: 指定通道数目, 通常为1或2。
pcm_bit_width: 指定位宽, 必须设置为16。
pcm_frame_size: 指定frame_size, 此值可不指定。
pcm_cap_duration: 指定测试时间。

测试指令：

【Tina】

```
# cd /mnt/extsd/  
# ./sample_ai -path ./sample_ai.conf
```

【Melis】

```
# sample_ai -path ./mnt/E/sample_ai.conf
```

退出测试：

测试时间达到指定时间后, 自动退出测试。

预期结果：

1. 测试程序运行正常, 测试过程没有异常打印。
2. 生成pcm 文件test.wav播放正常。

8.2.2 sample_ao

测试目的：

根据配置参数读取pcm数据，然后播放声音，从耳机口或喇叭输出声音。

组件依赖：

mpp_ao

测试通路：

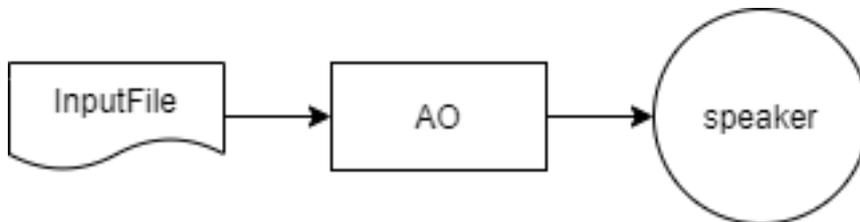


图 8-27: MPP_sample 测试通路-sample_ao

源文件：

/mnt/extsd/test.wav

目标文件：

/mnt/extsd/SampleAo_AoSaveFile.pcm

参数配置：

pcm_file_path: 指定音频pcm文件的路径，该文件是包含wave头(大小为44Bytes)的wav格式文件，如果找不到这种格式文件，可以用sample_ai生成一个。
pcm_sample_rate: 指定采样率，设置为文件中的采样率的值。
pcm_channel_cnt: 指定通道数目，设置为文件中的通道数。
pcm_bit_width: 指定位宽，设置为文件中的位宽。
pcm_frame_size: 固定指定为1024。
ao_volume: 输出的音量，范围：0~100。

测试指令：

【Tina】

```
# cd /mnt/extsd/  
# ./sample_ao -path ./sample_ao.conf
```

【Melis】

```
# sample_ao -path ./mnt/E/sample_ao.conf
```

退出测试：

播放完wav 文件后会自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 喇叭播放声音正常。同时，抓取pcm 文件SampleAo_AoSaveFile.pcm。

8.2.3 sample_aoSync

测试目的：

演示接口AW_MPI_AO_SendFrameSync 的使用方法，采用同步的方式send pcm frame。而sample_ao 是采用异步的方式。
根据配置参数读取pcm数据，然后播放声音，从耳机口输出声音。

组件依赖：

mpp_aoio

测试通路：

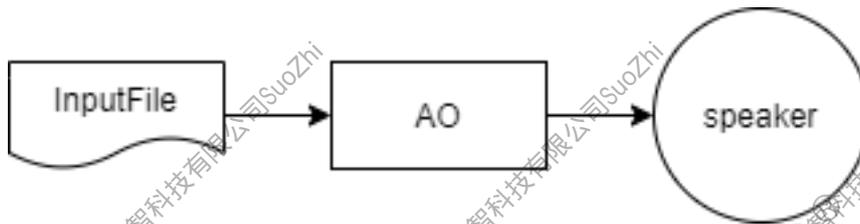


图 8-28: MPP_sample 测试通路-sample_aoSync

源文件：

/mnt/extsd/test.wav

目标文件：

无

参数配置：

pcm_file_path: 指定音频pcm文件的路径，该文件是包含wave头（大小为44Bytes）的wav格式文件，如果找不到这种格式文件，可以用sample_ai生成一个。

pcm_frame_size: 指定每次取pcm的帧数，用来与其他参数配合决定分配pcm frame buf 的大小。

ao_volume: 指定音量大小。

test_duration: 指定测试时间，单位：秒。

parse_wav_header_enable: 指定是否使用sample 内部wav header 的解析。

注意：如果使能，wav 文件在sample 内部会自己解析这些参数，并覆盖下面指定的参数值（pcm_sample_rate、pcm_channel_cnt、pcm_bit_width）。

pcm_sample_rate: 指定采样率，设置为文件中的采样率的值。

pcm_channel_cnt: 指定通道数目，设置为文件中的通道数。

pcm_bit_width: 指定位宽，设置为文件中的位宽。

测试指令：

【Tina】

```
# cd /mnt/extsd/
# ./sample_aoSync -path ./sample_aoSync.conf
```

【Melis】

```
# sample_aoSync -path ./mnt/E/sample_aoSync.conf
```

退出测试：

播放完wav 文件后会自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 喇叭播放声音正常。

8.2.4 sample_ao_resample_mixer

测试目的：

根据配置参数读取pcm数据，然后播放声音，从耳机口或喇叭输出声音。
可以测试resample及混音功能。

组件依赖：

mpp_ao

测试通路：

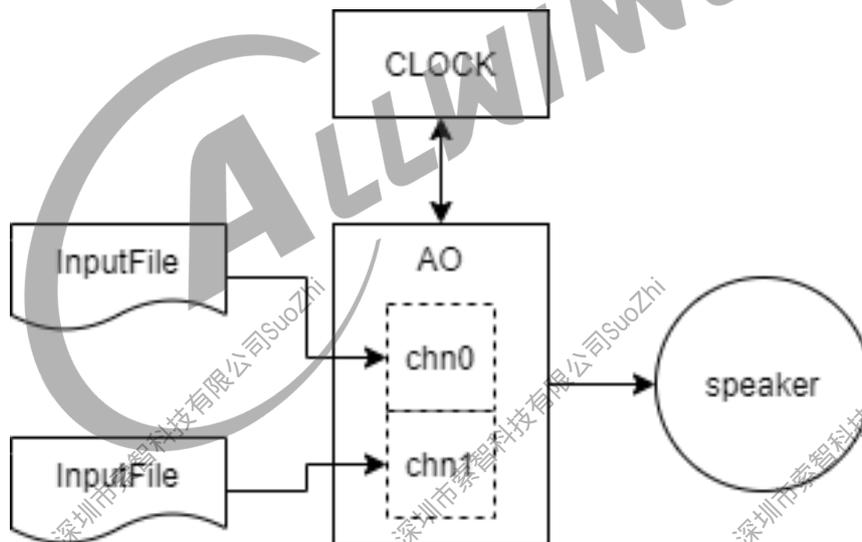


图 8-29: MPP_sample 测试通路-sample_ao_resample_mixer

源文件：

```
/mnt/extsd/speech.wav  
/mnt/extsd/48000_ch2_bit16.wav
```

目标文件：

无

参数配置：

pcm_file_path: 指定音频pcm文件的路径, 该文件是包含wave头(大小为44Bytes)的wav格式文件, 如果找不到这种格式文件, 可以用sample_ai生成一个。

pcm_sample_rate: 指定采样率, 设置为文件中的采样率的值。

pcm_channel_cnt: 指定通道数目, 设置为文件中的通道数。

pcm_bit_width: 指定位宽, 设置为文件中的位宽。

pcm_frame_size: 固定指定为1024。

pcm_second_chl_en: 是否启动第二路ao, 以测试resample及混音功能。

pcm_file_path_slave: 指定音频pcm文件的路径, 该文件是包含wave头(大小为44Bytes)的wav格式文件, 如果找不到这种格式文件, 可以用sample_ai生成一个。

pcm_sample_rate_slave: 指定采样率, 设置为文件中的采样率的值。

pcm_channel_cnt_slave: 指定通道数目, 设置为文件中的通道数。

pcm_bit_width_slave: 指定位宽, 设置为文件中的位宽。

pcm_frame_size_slave: 固定指定为1024。

测试指令:

【Tina】

```
# cd /mnt/extsd/  
# ./sample_ao_resample_mixer -path ./sample_ao_resample_mixer.conf
```

【Melis】

暂不支持。

退出测试:

播放完wav 文件后会自动退出测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 喇叭播放声音正常。

8.2.5 sample_ao2ai_aec

测试目的:

本sample主要用来演示aec功能的使用。

sample创建两路ai, 一路通过tunnel mode 绑定于audio enc, 直接aac编码, 并保存编码后的数据; 一路采用非tunnel模式, 由app获取ai数据, 做其他处理。

sample运行过程如下:

根据配置参数读取pcm数据, 然后播放声音。同时ai采集音频数据, 并做aec回声消除处理, 后送aac编码或直接获取aec后的数据用做他用。

组件依赖:

```
mpp_aenc  
mpp_aio
```

测试通路:

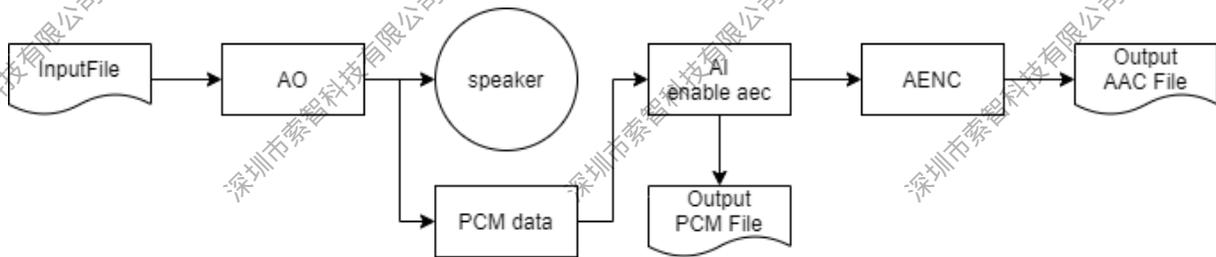


图 8-30: MPP_sample 测试通路-sample_ao2ai_aec

源文件:

```
/mnt/extsd/sample_ai/sample_ai_16000_ch1_bit16_aec_20.wav
```

目标文件:

```
/mnt/extsd/sample_ao2ai/tmp/ai_cap.pcm  
/mnt/extsd/sample_ao2ai/tmp/ai_cap.aac
```

参数配置:

pcm_src_path: 指定音频pcm文件的路径, 该文件是包含wave头(大小为44Bytes)的wav格式文件, 如果找不到这种格式文件, 可以用sample_ai生成一个。
 pcm_dst_path: 指定目标文件的路径, 该文件是ai组件采集音频生成的文件, 不带wav头, 如果想听该音频, 需手动加上wave头。
 aac_dst_path: 指定编码后aac目标文件路径, 该文件为aac raw data。
 pcm_sample_rate: 指定采样率, 设置为文件中的采样率的值, 启用aec后, sample rate须为8000。
 pcm_channel_cnt: 指定通道数目, 设置为文件中的通道数, 启用aec后, sample rate须为1。
 pcm_bit_width: 指定位宽, 设置为文件中的位宽, 启用aec后, bit_width须为16。
 pcm_frame_size: 帧大小, 固定指定为1024。
 aec_en: 是否启动aec回声消除功能。1: 启动; 0: 不启用。
 aec_delay_ms: 启用aec回声消除功能时, 设置的延迟时间, 主要给aec回声消除算法使用, 建议先设置为0。

测试指令:

[Tina]

```
# cd /mnt/extsd/  
# ./sample_ao2ai_aec -path ./sample_ao2ai_aec.conf
```

[Melis]

```
# sample_ao2ai_aec -path ./mnt/E/sample_ao2ai_aec.conf
```

退出测试:

按“ctrl + c”退出测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 喇叭播放没有回声。

8.2.6 sample_ao2ai_aec_rate_mixer

测试目的：

演示回声消除功能。

sample创建两路ai，一路通过tunnel mode 绑定于audio enc，直接aac编码，并保存编码后的数据；一路采用非tunnel模式，由app获取ai数据，做其他处理。

sample运行过程如下：

根据配置参数读取pcm数据，然后播放声音。同时ai采集音频数据，并做aec回声消除处理，后送aac编码或直接获取aec后的数据用做他用。

组件依赖：

mpp_aenc
mpp_aio

测试通路：

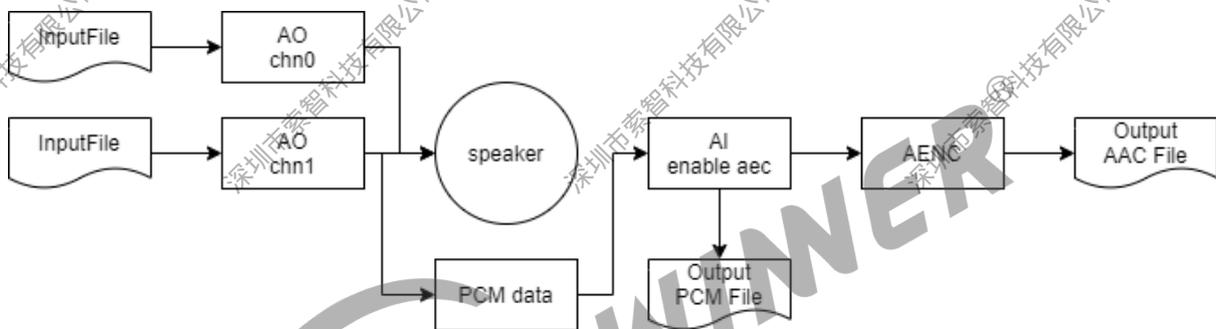


图 8-31: MPP_sample 测试通路-sample_ao2ai_aec_rate_mixer

源文件：

第一路：/mnt/extsd/speech.wav
第二路：/mnt/extsd/48000_ch2_bit16.wav

目标文件：

/mnt/extsd/ai_cap.pcm
/mnt/extsd/ai_cap.aac

参数配置：

第一路音频数据文件及对应参数：

pcm_src_path: 指定音频pcm文件的路径，该文件是包含wave头(大小为44Bytes)的wav格式文件，如果找不到这种格式文件，可以用sample_ai生成一个。

pcm_dst_path: 指定目标文件的路径，该文件是ai组件采集音频生成的文件，不带wav头，如果想听该音频，需手动加上wave头。

aac_dst_path: 指定编码后aac目标文件路径，该文件为aac raw data。

pcm_sample_rate: 指定采样率，设置为文件中的采样率的值，启用aec后，sample rate须为8000。

pcm_channel_cnt: 指定通道数目，设置为文件中的通道数，启用aec后，sample rate须为1。

pcm_bit_width: 指定位宽，设置为文件中的位宽，启用aec后，bit_width须为16。

pcm_frame_size: 固定指定为1024。

aec_en: 1:启动aec回声消除功能；0:不启用aec回声消除功能；

aec_delay_ms: 启用aec回声消除功能时，设置的延迟时间，主要给aec回声消除算法使用，建议先设置为0。

第二路音频数据文件及对应参数：

pcm_src_path_slave: 指定音频pcm文件的路径。
pcm_sample_rate_slave: 指定采样率。
pcm_channel_cnt_slave: 指定通道数目。
pcm_bit_width_slave: 指定位宽。
pcm_frame_size_slave: 指定帧大小, 固定指定为1024。

测试指令：

【Tina】

```
# cd /mnt/extsd/  
# ./sample_ao2ai_aec_rate_mixer -path ./sample_ao2ai_aec_rate_mixer.conf
```

【Melis】

暂不支持。

退出测试：

按“ctrl + c”退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 喇叭播放没有回声。

8.2.7 sample_aec

测试目的：

本sample主要用来演示aec（回声消除）功能的使用。
sample创建ao，播放音频文件作为回声。
sample创建ai，接收外界声音（其中必然包括ao播放出的音频）。mpi_ai启动回声消除功能。
sample从mpi_ai组件获取采集的数据保存为wav文件，在mpi_ai打开回声消除的情况下，wav文件应已过滤了ao播放的音乐。如果没有打开回声消除，wav文件会混合外界说话声和ao播放的音乐。
sample也包含了DRC（软件增强音量）的测试。

启用aec功能的方式如下：

启用aec功能并不复杂，相比于不启用aec功能的操作，只是在设置ai dev属性时多了两个属性参数：

- 1) ai_aec_en: 是否使能aec。1: enable; 0: disable。
- 2) aec_delay_ms: aec算法用delay参数。单位：ms，建议先设为0。

然后调用api: AW_MPI_AI_SetPubAttr()。

音频回声消除（参考MPP媒体开发指南文档）

内核驱动采集播放出的音频数据，提供接口供获取。AI 通道获取播放的音频数据作为音频参考帧，利用回声消除算法，消除采集的音频数据中的相同音频帧。

组件依赖：

mpp_ao

测试通路：

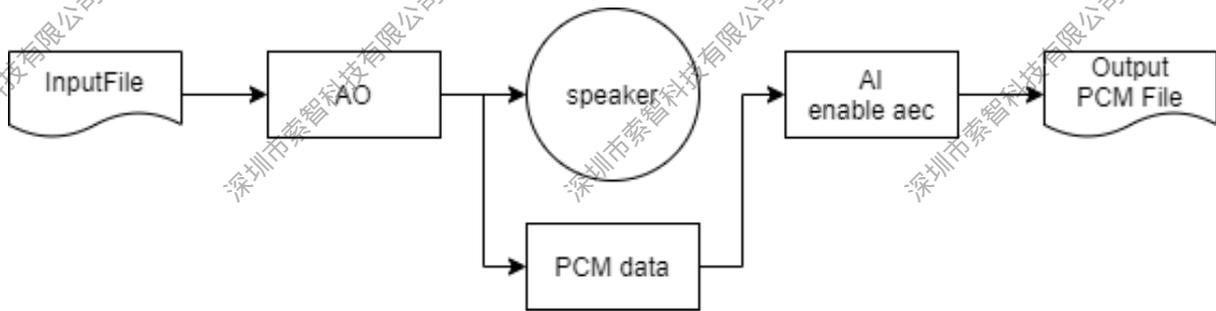


图 8-32: MPP_sample 测试通路-sample_aec

源文件：

```
/mnt/extsd/sample_aoref_8000_ch1_bit16_aec_30s.wav
```

目标文件：

```
/mnt/extsd/ai_cap.wav
```

参数配置：

pcm_src_path: 指定音频pcm文件的路径, 该文件是包含wave头(大小为44Bytes)的wav格式文件, 如果找不到这种格式文件, 可以用sample_ai生成一个。

pcm_dst_path: 指定目标文件的路径, 该文件是ai组件采集音频生成的文件, 可配置是否带wav头, 如果想在PC上播放音频文件, 需带wav头。

pcm_sample_rate: 指定采样率, 设置为文件中的采样率的值, 启用aec后, sample rate须为8000。

pcm_channel_cnt: 指定通道数目, 设置为文件中的通道数, 启用aec后, sample rate须为1。

pcm_bit_width: 指定位宽, 设置为文件中的位宽, 启用aec后, bit_width须为16。

pcm_frame_size: 固定指定为1024。

aec_en: 1:启动aec回声消除功能; 0: 不启用aec回声消除功能;

aec_delay_ms: 启用aec回声消除功能时, 设置的延迟时间, 主要给aec回声消除算法使用, 建议先设置为0;

add_wav_header: 保存pcm文件是否需要加wav头。

测试指令：

【Tina】

```
# cd /mnt/extsd/
# ./sample_aec -path ./sample_aec.conf
```

【Melis】

```
# sample_aec -path ./mnt/E/sample_aec.conf
```

退出测试：

按“ctrl + c”退出测试。

预期结果：

1. 测试程序运行正常, 测试过程没有异常打印。
2. 喇叭播放没有回声。

8.2.8 sample_aenc

测试目的：

从pcm文件（如：test.wav）中读取每一帧的数据，进行编码，然后保存为aac/mp3/adpcm/pcm/g711a/g711u/g726格式的压缩文件。

组件依赖：

mpp_aenc
mpp_aio

测试通路：

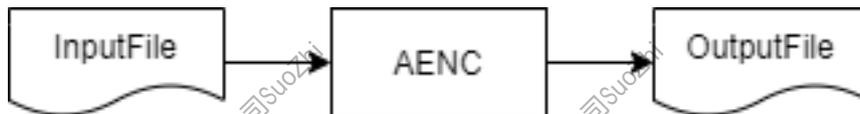


图 8-33: MPP_sample 测试通路-sample_aenc

源文件：

/mnt/extsd/test.wav

目标文件：

/mnt/extsd/test.aac

参数配置：

sample_aenc_src_file: 指定原始pcm文件的路径，该文件是包含wave头的wav格式文件。
sample_aenc_dst_file: 指定编码后生成的aac或mp3或其它格式文件的路径。注意指定的后缀名必须小写，用于指定编码方式，否则按aac格式来编码。

测试指令：

【Tina】

```
# cd /mnt/extsd/  
# ./sample_aenc -path ./sample_aenc.conf
```

【Melis】

```
# sample_aenc -path ./mnt/E/sample_aenc.conf
```

退出测试：

处理完wav 文件后，会自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 音频编码生成文件test.aac 播放正常。

8.2.9 sample_ai2aenc

测试目的：

mic录音送入编码器，取得每一帧数据的编码，写到文件中。

组件依赖：

mpp_aenc
mpp_aio

测试通路：

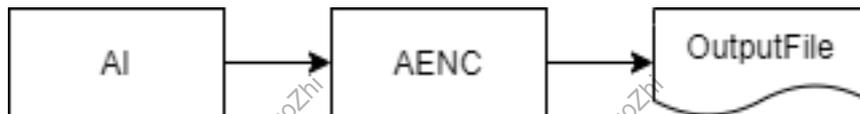


图 8-34: MPP_sample 测试通路-sample_ai2aenc

源文件：

无

目标文件：

/mnt/extsd/test.aac

参数配置：

dst_file: 生成文件路径
encoder_type: 编码类型, 如“aac”
sample_rate: mic录音采样值
channel_cnt: 录音通道数 (1 or 2)
bit_width: 录音采样位宽
frame_size: 帧大小 (如: 1024 / 2048)
test_duration: sample一次测试时间(单位: s)

测试指令：

【Tina】

```
# cd /mnt/extsd/  
# ./sample_ai2aenc -path ./sample_ai2aenc.conf
```

【Melis】

暂不支持。

退出测试：

测试时间达到指定时间后，自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 生成编码文件test.aac播放正常。

8.2.10 sample_ai2aenc2muxer

测试目的：

根据配置参数采集对应的pcm数据，然后根据配置信息将pcm数据编码，最后写入到文件中进行保存。

组件依赖：

mpp_aenc
mpp_aio

测试通路：

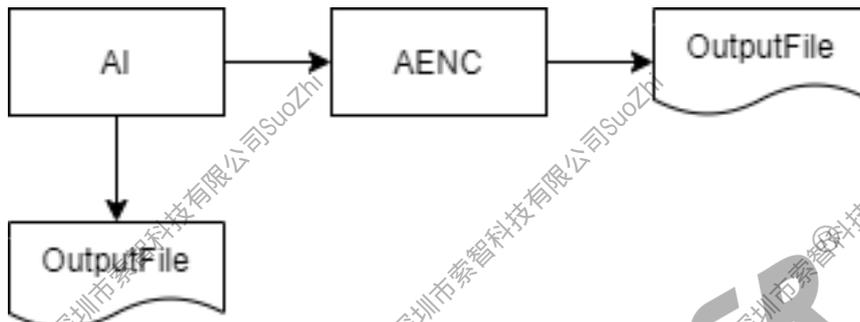


图 8-35: MPP_sample 测试通路-sample_ai2aenc2muxer

源文件：

无

目标文件：

/mnt/extsd/SampleAi2Aenc2Muxer_AiSaveFile.pcm
/mnt/extsd/sample_ai2aenc2muxer_mono_16000.aac

参数配置：

dst_file: 指定目标pcm文件的路径，该文件是包含wave头的wav格式文件。
cap_dura: 指定采集时间长度，单位：s。
chn_cnt: 指定通道数目，通常为1或2。
bit_width: 指定位宽，必须设置为16。
smp_rate: 指定采样率，通常设置为8000。
bitRate: 编码的码率，单位：bps。

测试指令：

【Tina】

```
# cd /mnt/extsd/  
# ./sample_ai2aenc2muxer -path ./sample_ai2aenc2muxer.conf
```

【Melis】

```
# sample_ai2aenc2muxer -path ./mnt/E/sample_ai2aenc2muxer.conf
```

退出测试：

测试时间达到指定时间后，自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 生成文件sample_ai2aenc2muxer_mono_16000.aac 播放正常，同时抓取pcm 文件 SampleAi2Aenc2Muxer_AiSaveFile.pcm。

8.2.11 sample_select

测试目的：

演示AW_MPI_SYS_HANDLE_Select()的用法。

从pcm文件（如：sample_aenc.wav）中读取每一帧的数据，进行不同编码类型同时编码，然后保存为aac/mp3/adpcm/pcm/g711a/g711u/g726格式的压缩文件。

组件依赖：

mpp_aenc

测试通路：

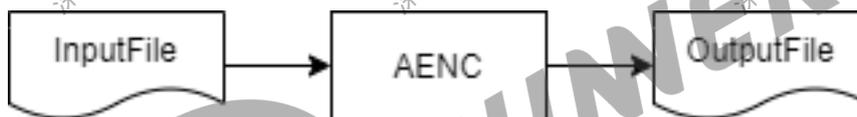


图 8-36: MPP_sample 测试通路-sample_select

源文件：

/mnt/extsd/sample_select.wav

目标文件：

/mnt/extsd/sample_select.aac

参数配置：

sample_select_asrc_file: 指定原始pcm文件的路径，该文件是包含wave头的wav格式文件。

sample_select_adst_file: 指定编码后生成的aac/mp3/adpcm/pcm/g711a/g711u/g726格式文件的路径。

测试指令：

【Tina】

```
# cd /mnt/extsd/
# ./sample_select -path ./sample_select.conf
```

【Melis】

暂不支持。

退出测试：

测试完wav文件会自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 生成aac 文件sample_select.aac 播放正常。

8.2.12 sample_adec

测试目的：

从已编码的ADTS格式的AAC文件（如：test.aac）中读取每一帧的数据，进行解码，然后保存为WAV格式的pcm文件。

组件依赖：

mpp_adec
mpp_aio

测试通路：



图 8-37: MPP_sample 测试通路-sample_adec

源文件：

/mnt/extsd/test.aac

目标文件：

/mnt/extsd/test.wav

参数配置：

aac_file_path: 指定原始已压缩的音频文件的路径，该文件必须为aac格式的文件。
pcm_file_path: 指定解码aac后生成的pcm文件的路径。

测试指令：

[Tina]

```
# cd /mnt/extsd/  
# ./sample_adec -path ./sample_adec.conf
```

[Melis]

```
# sample_adec -path ./mnt/E/sample_adec.conf
```

退出测试：

aac 文件处理完，会自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 音频解码生成目标文件test.wav 播放正常。

8.2.13 sample_adec2ao

测试目的：

从已编码的ADTS格式的AAC文件（如：test.aac）中读取每一帧的数据，进行解码，然后通过ao输出。

组件依赖：

mpp_adec
mpp_aio

测试通路：

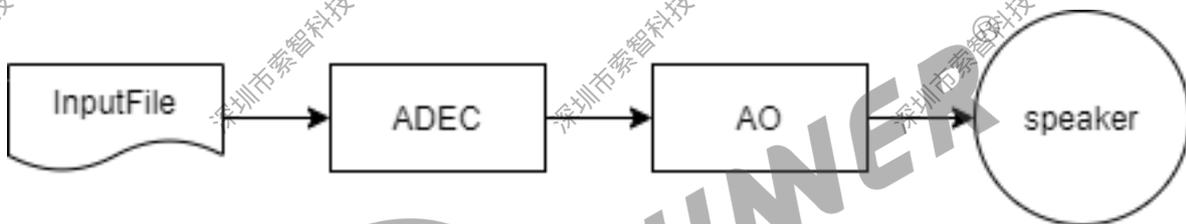


图 8-38: MPP_sample 测试通路-sample_adec2ao

源文件：

/mnt/extsd/test.aac

目标文件：

无

参数配置：

audio_file_path: 指定音频文件路径。
 pcm_sample_rate: 指定pcm的采样率，单位：Hz。默认16000，表示16000Hz。对于g711和g726需要配置该值。
 pcm_channel_cnt: 指定pcm通道数，默认1，表示单通道。对于g711和g726需要配置该值。
 audio_data_type: 指定音频数据类型（g711a:19;g711u:20;aac:37;g726a:21,g726u:2000）。

测试指令：

【Tina】

```
# cd /mnt/extsd/
# ./sample_adec2ao -path ./sample_adec2ao.conf
```

【Melis】

不支持

退出测试：

音频文件处理完，会自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 音频解码后，输出声音正常。

8.2.14 sample_demux2adec

测试目的：

根据配置参数读取视频文件，解封装，获取到音频数据后，送解码器，保存为wav文件。

组件依赖：

mpp_demuxer
mpp_adec

测试通路：

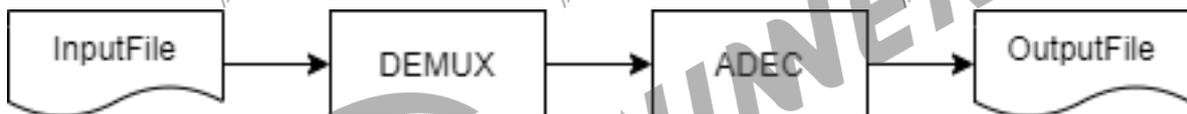


图 8-39: MPP_sample 测试通路-sample_demux2adec

源文件：

/mnt/extsd/test.mp4

目标文件：

/mnt/extsd/output.wav

参数配置：

src_file: 指定mp4视频文件的路径。
dst_file: 指定wav音频文件的路径。

测试指令：

【Tina】

```
# cd /mnt/extsd/  
# ./sample_demux2adec -path ./sample_demux2adec.conf
```

【Melis】

暂不支持。

退出测试：

测试完文件test.mp4，可以自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 生成音频文件output.wav 播放正常。

8.2.15 sample_demux2adec2ao

测试目的：

根据配置参数读取视频文件，解封装，获取到音频数据后，送解码器，然后播放出声音，从耳机口或喇叭输出声音。

组件依赖：

```
mpp_demuxer
mpp_adec
mpp_aio
```

测试通路：

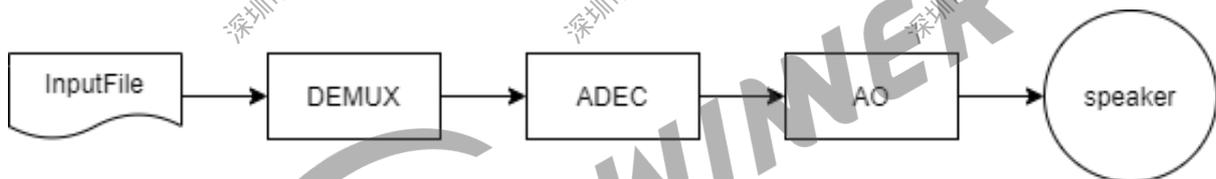


图 8-40: MPP_sample 测试通路-sample_demux2adec2ao

源文件：

```
/mnt/extsd/test.mp4
```

目标文件：

```
无
```

参数配置：

```
src_file: 指定mp4视频文件的路径。
```

测试指令：

[Tina]

```
# cd /mnt/extsd/
# ./sample_demux2adec2ao -path ./sample_demux2adec2ao.conf
```

[Melis]

```
# sample_demux2adec2ao -path ./mnt/E/sample_demux2adec2ao.conf
```

退出测试：

测试完文件test.mp4，可以自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 喇叭播放声音，声音正常。

8.3 ISE 和 EIS

8.3.1 sample_fish

测试目的：

该sample测试mpi_ise组件单目鱼眼功能。

创建ise组件，将图像传输给mpi_ise，ISE组件对图像进行校正，通过调用mpi获取ISE组件处理后的数据，到达指定次数后，停止运行并销毁ISE组件。也可以手动按ctrl-c，终止测试。

如果需要获取ISE组件处理后的YUV数据，需要将sample_fish.c中的Save_Picture宏打开；如果需要在Normal模式下对PTZ参数进行调节，需要将sample_fish.c中的Dynamic_PTZ宏打开。

ISE GDC补充说明：

对于新增的ISE GDC算法，由于Warp_Ptz4In1这个模式要求输出分辨率是输入分辨率的两倍，而现有的2048原图对于内存以及硬件方面性能有所增加，所以针对Warp_Ptz4In1增加了一张1024原图以及sample_fish_ptz4in1.conf，若需要运行Warp_Ptz4In1这个模式可以使用fisheye_1024x1024.yuv420这张图像以及sample_fish_ptz4in1.conf。

组件依赖：

```
mpp_ise
mpp_vdec
mpp_hw_display
mpp_vo
mpp_adec
mpp_demuxer
```

测试通路：

无

源文件：

```
/mnt/extsd/fisheye_2048x2048.yuv420
```

目标文件：

无

参数配置：

在C源文件里面的ISE_GROUP_INS_CNT宏可以控制测试开启的Ise实例数量，目前默认为2

1. auto_test_count：指定自动化测试次数
2. process_count：指定ISE组件处理次数
3. pic_width：指定源图像宽度
4. pic_height：指定源图像高度
5. pic_stride：指定源图像的stride值，该值必须是32的倍数
6. pic_frame_rate：指定发送源图像的帧率

7. pic_file_path: 指定源图像的路径
 8. ise_dewarp_mode: 指定单目鱼眼校正的模式, 包括180模式(WARP_PAN0180)/360度左右展开模式(WARP_PAN0360)/PTZ模式(WARP_NORMAL)/畸变校正模式(WARP_UNDISTORT)/360度上下展开模式(WARP_180WITH2)
 9. ise_mount_mode: 指定PTZ模式下镜头的安装方式, 包括顶装(MOUNT_TOP)/壁装(MOUNT_WALL)/地装(MOUNT_BOTTOM)指定360度上下展开模式下镜头的安装方式, 包括顶装(MOUNT_TOP)/壁装(MOUNT_WALL)指定360度左右展开模式下镜头的安装方式, 包括顶装(MOUNT_TOP)/壁装(MOUNT_WALL)
 10. ise_normal_pan: 左右移动的角度
 11. ise_normal_tilt: 上下移动的角度
 12. ise_normal_zoom: 镜头变焦倍数
 13. ise_port_num: 指定ISE组件端口个数
 14. ise_output_file_path: 指定ISE组件输出图像的路径
 15. ise_portx_width: 指定校正处理后图像的宽度
 16. ise_portx_height: 指定校正处理后图像的高度
 17. ise_portx_stride: 指定校正处理后图像的stride值, 该值必须是32的倍数
 18. ise_portx_flip_enable: 指定是否使能图像翻转
 19. ise_portx_mirror_enable: 指定是否使能图像镜像
- 注: 每个ISE端口图像翻转和镜像只能使能其中一个, 测试时需要拷贝yuv文件到测试路径下面, 程序默认打开程序所在文件夹的yuv文件。

测试指令:

[Tina]

```
# cd /mnt/extsd/  
# ./sample_fish_path ./sample_fish.conf
```

[Melis]

暂不支持。

退出测试:

自动退出测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。

8.3.2 sample_virvi2fish2venc

测试目的:

该sample测试mpi_vi、mpi_ise、mpi_venc组件的绑定组合。创建mpi_vi、mpi_ise和mpi_venc, 将它们绑定, 再分别启动。mpi_vi采集图像, 传输给mpi_ise对鱼眼图像进行校正, mpi_ise将校正后的图像传给mpi_venc进行编码, 同时保存裸码流视频文件。

组件依赖:

mpp_ise

测试通路:

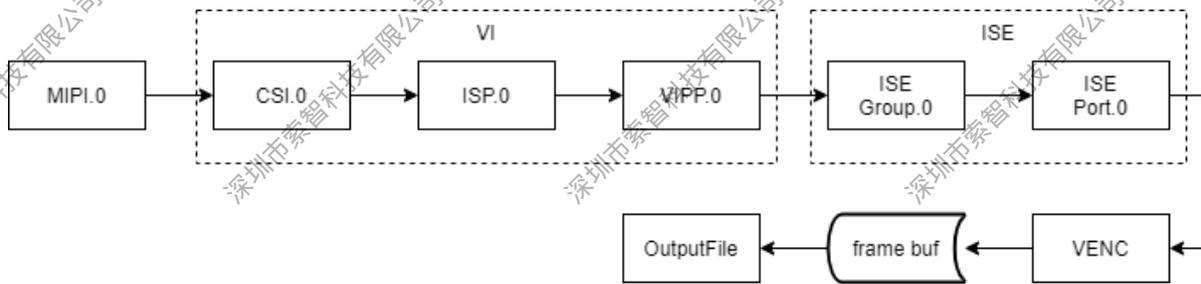


图 8-41: MPP_sample 测试通路-sample_virvi2fish2venc

源文件:

无

目标文件:

/mnt/extsd/AW_FishEncoderVideoChn0.H264

参数配置:

1. auto_test_count: 指定自动测试次数
 2. dev_id: 指定VI Dev设备节点
 3. src_width: 指定camera采集的图像宽度,由于VI模块硬件设计的原因,src_width必须是32的倍数
 4. src_height: 指定camera采集的图像高度
 5. src_frame_rate: 指定camera采集图像的帧率
 6. ISEPortNum: 指定ISE组件端口个数
 7. ise_dewarp_mode: 指定单目鱼眼校正的模式,包括180模式(WARP_PAN0180)/360度左右展开模式(WARP_PAN0360)/PTZ模式(WARP_NORMAL)/畸变校正模式(WARP_UNDISTORT)/360度上下展开模式(WARP_180WITH2)
 8. ise_mount_mode: 指定PTZ模式下镜头的安装方式,包括顶装(MOUNT_TOP)/壁挂(MOUNT_WALL)/地装(MOUNT_BOTTOM)
指定360度上下展开模式下镜头的安装方式,包括顶装(MOUNT_TOP)/壁挂(MOUNT_WALL)
指定360度左右展开模式下镜头的安装方式,包括顶装(MOUNT_TOP)/壁挂(MOUNT_WALL)
 9. ise_normal_pan: 左右移动的角度
 10. ise_normal_tilt: 上下移动的角度
 11. ise_normal_zoom: 镜头变焦倍数
 12. ise_portx_width: 指定校正后图像的宽度
 13. ise_portx_height: 指定校正后图像的高度
 14. ise_portx_stride: 指定校正后图像的stride值,该值必须是32的倍数
 15. ise_portx_flip_enable: 指定是否使能图像翻转
 16. ise_portx_mirror_enable: 指定是否使能图像镜像
 17. VencChnNum: 指定VENC组件通道个数
 18. encoder_type: 指定编码格式
 19. encoder_count: 指定每次测试编码帧数, -1为循环编码,不退出sample_virvi2fish2venc
 20. picture_format: 指定编码后图像的格式
 21. venc_chnx_dest_width: 指定编码后图像的宽度
 22. venc_chnx_dest_height: 指定编码后图像的高度
 23. venc_chnx_dest_frame_rate: 指定编码帧率
 24. venc_chnx_dest_bit_rate: 指定编码码率
 25. venc_chnx_output_file_path: 指定编码后视频文件的路径
- 注: 每个ISE端口图像翻转和镜像只能使能一个其中一个

测试指令:

【Tina】

```
# cd /mnt/extsd/
# ./sample_virvi2fish2venc -path ./sample_virvi2fish2venc.conf
```

【Melis】

暂不支持。

退出测试：

测试达到设定的时间后自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 使用VLC播放测试生成的目标文件/mnt/extsd/AW_FishEncoderVideoChn0.H264 正常。

测试说明：

由于硬件设计的原因，一个ISE组件对应一个Group，一个Group最多可以创建4个Port，其中Port 0为ISE硬件模块处理后输出的图像，Port 1~Port 3是对Port 0输出的图像进行缩放而得到的。Group创建完后必须创建Port 0，Port 1~Port 3可以根据实际需要依次创建，Port 1~Port 3支持无极缩放，图像缩放的范围为Port 0宽高的1/8 ~ 1，各个Port之间相互独立，互不影响。ISE组件端口ID号和VENC组件通道ID号一一对应，ISE组件端口个数必须与VENC组件通道个数相等。

8.3.3 sample_virvi2fish2vo

测试目的：

该sample测试mpi_vi、mpi_ise、mpi_vo组件的绑定组合。创建mpi_vi、mpi_ise和mpi_vo，将它们绑定，再分别启动。mpi_vi采集图像，传输给mpi_ise对鱼眼图像进行校正，mpi_ise将校正后的图像传给mpi_vo进行显示预览。

组件依赖：

```
mpp_hw_display
mpp_vo
mpp_ise
```

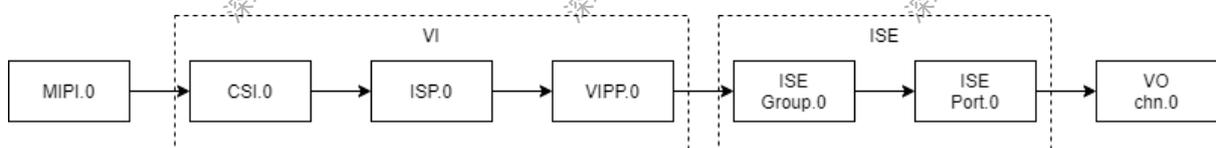
测试通路：

图 8-42: MPP_sample 测试通路-sample_virvi2fish2vo

源文件：

无

目标文件：

```
/mnt/extsd/isp0_1920_1080_60_ctx_saved.bin
```

参数配置：

1. auto_test_count: 指定自动化测试次数
2. dev_id: 指定VI_Dev设备节点
3. src_width: 指定camera采集的图像宽度,由于VI模块硬件设计的原因,src_width必须是32的倍数
4. src_height: 指定camera采集的图像高度
5. src_frame_rate: 指定camera采集图像的帧率
6. ise_dewarp_mode: 指定单目鱼眼校正的模式,包括180模式(WARP_PAN0180)/360度左右展开模式(WARP_PAN0360)/Normal模式(WARP_NORMAL)/畸变校正模式(WARP_UNDISTORT)/360度上下展开模式(WARP_180WITH2)
7. ise_mount_mode: 指定Normal模式下镜头的安装方式,包括顶装(MOUNT_TOP)/壁挂(MOUNT_WALL)/地装(MOUNT_BOTTOM)指定360度上下展开模式下镜头的安装方式,包括顶装(MOUNT_TOP)/壁挂(MOUNT_WALL)指定360度左右展开模式下镜头的安装方式,包括顶装(MOUNT_TOP)/壁挂(MOUNT_WALL)
8. ise_normal_pan: 左右移动的角度
9. ise_normal_tilt: 上下移动的角度
10. ise_normal_zoom: 镜头变焦倍数
11. ise_width: 指定校正后图像的宽度
12. ise_height: 指定校正后图像的高度
13. ise_stride: 指定校正后图像的stride值,该值必须是32的倍数
14. ise_flip_enable: 指定是否使能图像翻转
15. ise_mirror_enable: 指定是否使能图像镜像
16. display_width: 指定显示宽度
17. display_height: 指定显示高度
18. vo_test_duration: 指定测试时间,0为循环显示,不退出sample_virvi2fish2vo

测试指令：

【Tina】

```
# cd /mnt/extsd/
# ./sample_virvi2fish2vo -path ./sample_virvi2fish2vo.conf
```

【Melis】

```
# sample_virvi2fish2vo -path ./mnt/E/sample_virvi2fish2vo.conf
```

退出测试：

不自动退出测试,需要“ctrl + c”退出测试

预期结果：

1. 测试程序运行正常,测试过程没有异常打印。
2. LCD屏幕实时显示Camera采集的视频图像正常。

8.3.4 sample_virvi2eis2venc

测试目的：

该例程用于测试防抖相关功能。支持离线仿真、在线调试模式,支持绑定、非绑定的组件形式。

1. 使用 config.mk 文件里面的 EIS_OFFLINE_SIMU = Y/N 来控制是否使用离线仿真模式。
 2. 使用 config.mk 文件里面的 EIS_MULTI_INSTANCE = Y/N 来控制是否使用防抖多实例模式。
 3. Bypass 模式需要修改根目录下的下面注释(用于旁路防抖输出的视频数据,通常用于防抖调试或者仿真的数据采集):
- ```
#if 0
ViEisVencCfg.EisCfg.stEisAttr.eEisAlgoMode = EIS_ALGO_MODE_BP;
```

```

ViEisVencCfg.EisCfg.stEisAttr.pBPDataSavePath = "/tmp";
ViEisVencCfg.EisCfg.stEisAttr.bSaveYUV = 0;
#endif
并且该模式必须全部关闭 V0 打开编码，也就是 use_vo = 0; use_venc = 1
4. 使用 #define EIS_BYPASS_VENC 宏定义来控制是否开启离线模式下的编码，如果定义了该宏，那么会直接输出 YUV 文件。

```

#### 组件依赖：

```

mpp_eis
mpp_hw_display
mpp_vo

```

#### 测试通路：

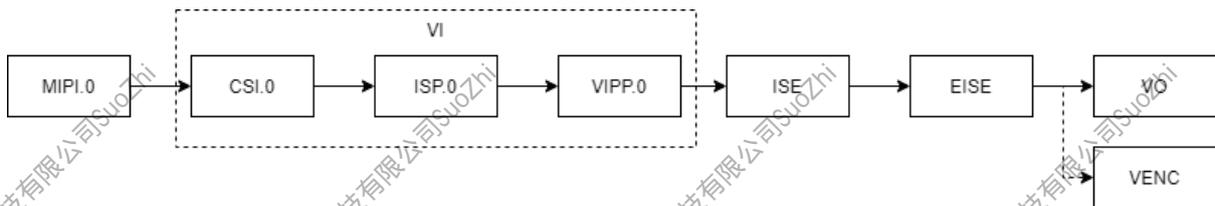


图 8-43: MPP\_sample 测试通路-sample\_virvi2eis2venc

#### 源文件：

无

#### 目标文件：

无

#### 参数配置：

```

reboot_cnt = 1000 #仅用于在线模式，控制重启测试的次数
frame_cnt = 150 #控制采集的视频帧数

input_bufnum = 10 #输入 buffer 的数量，目前调试阶段建议值至少为 11
output_bufnum = 5 #输出 buffer 的数量，目前调试阶段建议值至少为 10
time_delay = 0 #控制陀螺仪与视频帧的时间戳对齐偏移，目前推荐 -33
sync_tolerance = 1 #时间戳对齐的容忍度，建议值 5 以内，不能为 0
src parameter
dev number:0(CSI0)/1(CSI1)
cap_width * cap_height:720p/1080p;
fps: 30
video_dev = 1 #仅用于在线模式，控制采集视频的节点
cap_width = 1920 #仅用于在线模式，控制采集视频的分辨率
cap_height = 1080
fps = 30

Vo 如果与编码同时打开，则 Vo 是优先的，也就是编码的使能选项会被 V0 的使能选项覆盖掉。
V0, type: [lcd]\[hdmi]
use_vo = 1 # 仅用于在线模式，控制是否开启视频实时预览（在多实例模式下该选项无效，强制打开）
disp_w = 1920 # 仅用于在线模式，控制视频显示的分辨率
disp_h = 1080
disp_type = "hdmi" # 仅用于在线模式，视频显示的终端类型

VENC: default use EIS output size

```

```
use_venc = 0 # 在多实例模式下该选项无效，强制打开，离线仿真模式下该选项使用 EIS_BYPASS_VENC 来控制。

VENC save file and offline simulation mode save file path.
save_file = "/mnt/extsd/SampleEis2Venc.H264" # 控制编码输出文件的保存位置（离线模式如果不开编码则默认保存在 /tmp/目录下，路径为：/tmp/SampleEis2File.yuv)

gyro_file = "/mnt/extsd/SampleEis2VencGyro.txt" #仅用于离线模式，指定输入的陀螺仪数据文件
video_file = "/mnt/extsd/SampleEis2VencVideo.yuv" #仅用于离线模式，指定输入的视频数据文件
timestamp_file = "/mnt/extsd/SampleEis2VencTs.txt" #仅用于离线模式，指定输入的视频的时间戳数据文件
```

测试指令：

【Tina】

```
cd /mnt/extsd/
./sample_virvi2eis2venc -path ./sample_virvi2eis2venc.conf
```

【Melis】

暂不支持。

退出测试：

按“ctrl + c”退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 测试防抖功能正常。

### 8.3.5 sample\_ise\_dzoom

测试目的：

该sample测试使用ise实现dzoom功能。  
需要注意的是，此功能需要使用gdc算法库，因此必须在启用mpp\_ise组件时，勾选gdc算法库。在此sample中，VI组件与ISE通过绑定的方式，向ISE组件输入YUV数据。ISE组件的输出使用非绑定模式，手动获取ISE数据送入V0组件。通过设置配置文件中的display\_rotation可设置g2d旋转。通过设置set\_crop，sample会起一个线程取测试动态设置crop到ISE组件。

组件依赖：

```
mpp_vi
mpp_hw_display
mpp_vo
mpp_ise
```

测试通路：

无

源文件：

无

目标文件：

无

参数配置：

dev\_id: 指定VI Dev设备节点  
src\_width: 指定camera采集的图像宽度,由于VI模块硬件设计的原因,src\_width必须是32的倍数  
src\_height: 指定camera采集的图像高度  
src\_frame\_rate: 指定camera采集图像的帧率  
  
display\_width: 指定显示宽度  
display\_height: 指定显示高度  
display\_rotation: 图像旋转角度  
  
ise\_width: 指定校正后图像的宽度  
ise\_height: 指定校正后图像的高度  
ise\_stride: 指定校正后图像的stride值,该值必须是32的倍数  
ise\_ldc\_zoom\_h: 水平视角角的缩放尺度  
ise\_ldc\_zoom\_v: 垂直视角角的缩放尺度  
ise\_ldc\_center\_offset\_x: 图像中心点相对物理中心点的水平偏移  
ise\_ldc\_center\_offset\_y: 图像中心点相对物理中心点的竖直偏移  
  
test\_crop: 测试动态设置crop  
test\_duration: 指定测试时间,0为循环显示,不退出sample\_ise\_dzoom

测试指令：

【Tina】

```
cd /mnt/extsd/
./sample_ise_dzoom -path ./sample_ise_dzoom.conf
```

【Melis】

暂不支持。

结束测试：

达到设定的测试时间后，自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 通过V0设备查看缩放效果正常。

### 8.3.6 sample\_gdc\_dzoom

测试目的：

该sample测试使用gdc算法库实现dzoom功能。  
需要内核打开ISE硬件驱动，并在menuconfig中勾选gdc算法库。

组件依赖：

mpp\_gdc

测试通路：

无

源文件：

无

目标文件：

无

参数配置：

```
src_pic_width: 源图片宽度
src_pic_height: 源图片高度
src_pic_format: 源图片格式
src_pic_stride: 源图片Y分量存储宽度
src_file_path: 原图片文件路径

dst_pic_width: 输出图片宽度(需32位对齐)
dst_pic_height: 输出图片高度
dst_pic_format: 输出图片格式
dst_pic_stride: 输出图片Y分量存储宽度
dst_file_path: 输出图片文件路径

crop_x: crop区域横坐标
crop_y: crop区域纵坐标
crop_w: crop区域宽度
crop_h: crop区域高度

warp_mode: gdc工作模式
```

测试指令：

【Tina】

```
cd /mnt/extsd/
./sample_gdc_dzoom -path ./sample_gdc_dzoom.conf
```

【Melis】

暂不支持。

结束测试：

达到设定的测试时间后，自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 通过V0设备查看缩放效果正常。

## 8.4 视频显示

### 8.4.1 sample\_vo

测试目的：

从yuv原始数据文件xxx.yuv中读取视频帧，标记时间戳，送给mpi\_vo组件显示。  
sample\_vo也负责视频帧的帧管理，接收mpi\_vo组件归还的视频帧，重装新帧，再送入mpi\_vo组件显示。

组件依赖：

mpp\_hw\_display  
mpp\_vo

测试通路：

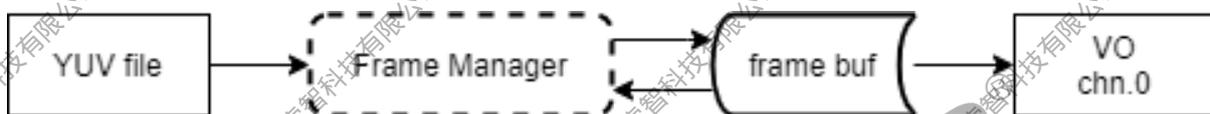


图 8-44: MPP\_sample 测试通路-sample\_vo

源文件：

/mnt/extsd/resources/Tennis\_1920x1080\_24.yuv  
测试需要的yuv文件，可以用ffmpeg转换得到，转换指令：  
./ffmpeg -i two\_video.mp4 Tennis\_1920x1080\_24.yuv

目标文件：

无

参数配置：

yuv\_file\_path: 指定yuv原始数据文件的路径。  
pic\_width: 指明yuv原始数据文件的视频帧的宽度  
pic\_height: 指明yuv原始数据文件的视频帧的高度  
display\_width: 指定输出图像的宽度  
display\_height: 指定输出图像的高度  
pic\_format: 指明yuv原始数据文件的视频帧的像素格式  
disp\_type: 指定显示设备类型 (hdm, lcd, cvbs)  
frame\_rate: 指定播放yuv原始数据文件的帧率

测试指令：

【Tina】

```
cd /mnt/extsd/
./sample_vo -path ./sample_vo.conf
```

【Melis】

暂不支持。

退出测试：

播放完yuv 文件后自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. LCD屏幕显示播放yuv 文件的视频图像正常。

## 8.4.2 sample\_UILayer

测试目的：

测试UILayer的格式。  
sample自己创建指定格式的RGB图，设置给UILayer。

组件依赖：

mpp\_hw\_display  
mpp\_vo

测试通路：

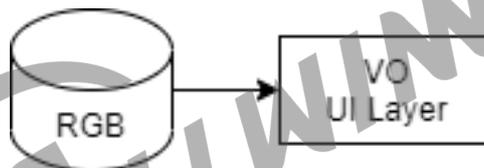


图 8-45: MPP\_sample 测试通路-sample\_UILayer

源文件：

无

目标文件：

无

参数配置：

pic\_width: 源图的宽度  
pic\_height: 源图的高度  
display\_width: 指定输出图像的宽度  
display\_height: 指定输出图像的高度  
bitmap\_format: 源图格式

测试指令：

**[Tina]**

```
cd /mnt/extsd/
./sample_UILayer -path ./sample_UILayer.conf
```

## 【Melis】

暂不支持。

退出测试：

一直显示，需要“ctrl+c”手动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. LCD屏幕显示绿色和蓝色矩形填充。

## 8.5 G2D

### 8.5.1 sample\_g2d

测试目的：

该sample 通过直接处理image图片来演示g2d模块常用功能的使用方式。g2d模块支持最大2K输入输出。

组件依赖：

无

测试通路：

无

源文件：

/mnt/extsd/sample\_g2d/board\_1920x1080.nv12

目标文件：

/mnt/extsd/sample\_g2d/dst.yuv

参数配置：

g2d 模块功能的选择主要通过参数的组合来实现，如：

- 1) rotate  
配置conf文件中的dst\_rotate 配置项，设置要旋转的角度：0: none,1:90,2:180,3:270。  
Note: 此时 dst\_width 及 dst\_height配置项要分别与src\_width及src\_height一致，否则会同时启动scale功能，但此时会产生错误结果(g2d模块不能同时做rotation及scaling)。
- 2) scale  
配置conf文件中的dst\_rotate为0 (disable rotation) ,  
配置dst\_width dst\_height目标大小。
- 3) cut  
cut功能用来截取部分src内容，具体区域由  
src\_rect\_x =  
src\_rect\_y =  
src\_rect\_w =

```
src_rect_h =
 设定，注意：若无效启动rotation, dst_rotate = 0
4) pixel format格式转换
 该功能通过参数：
 pic_format = nv21
 dst_format = nv12
 来实现。
```

测试指令：

【Tina】

```
cd /mnt/extsd/
./sample_g2d -path ./sample_g2d.conf
```

【Melis】

暂不支持。

退出测试：

测试完后可以自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 使用YUVview 软件分析生成的YUV文件正常。

## 8.5.2 sample\_vi\_g2d

测试目的：

该sample从mpi\_vi组件获取视频帧，调用g2d驱动做旋转、剪切、缩放等处理，处理后的图像送mpi\_vo显示，也可保存为原始图片供分析。g2d模块支持最大2K输入输出。

组件依赖：

```
mpp_hw_display
mpp_vo
```

测试通路：

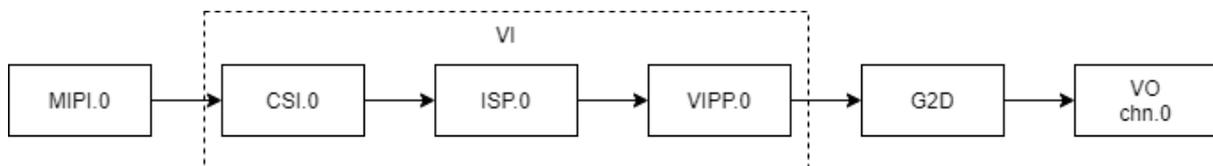


图 8-46: MPP\_sample 测试通路-sample\_vi\_g2d

源文件：

无

目标文件：

```
/mnt/extsd/pic[0].nv21
/mnt/extsd/pic[1].nv21
```

参数配置：

```
dev_num: 指定VI Dev设备节点
frame_rate: 指定camera采集图像的帧率
pic_format: 指定camera采集图像像素格式
drop_frame_num: 指定vi组件丢弃的帧数
src_width,src_height: 指定camera采集的源图宽度和高度
src_rect_x,src_rect_y,src_rect_w,src_rect_h: 指定g2d处理的源图区域
dst_rotate: 指定g2d旋转角度
dst_width,dst_height: 指定g2d处理后的目标buffer的宽度和高度
dst_rect_x,dst_rect_y,dst_rect_w,dst_rect_h: 指定g2d处理的目标图区域
dst_store_count: 指定保存图片的数量
dst_store_interval: 指定保存图片的帧间隔
store_dir: 指定保存目录，目录必须已经存在
display_flag: 是否V0显示
display_x,display_y,display_w,display_h: 指定V0的显示区域
test_duration: 指定测试时间，单位：秒。0表示无限时长。
```

测试指令：

**[Tina]**

```
cd /mnt/extsd/
./sample_vi_g2d -path ./sample_vi_g2d.conf
```

**[Melis]**

```
sample_vi_g2d -path ./mnt/E/sample_vi_g2d.conf
```

退出测试：

不自动退出测试，需要“ctrl + c”退出测试

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 使用YUVview 软件分析生成的YUV文件正常。
3. LCD屏幕实时显示Camera 采集的视频图像正常。

## 8.6 CE

### 8.6.1 sample\_twinchn\_virvi2venc2ce

测试目的：

```
该sample_同时测试两路编码的加解密。
两路mpi_vi 和mpi_venc 组件分别绑定组合。mpi_vi采集图像，直接传输给mpi_venc 进行编码，在进行ce加密解密。
```

组件依赖：

PACKAGE\_libopenssl

测试通路：

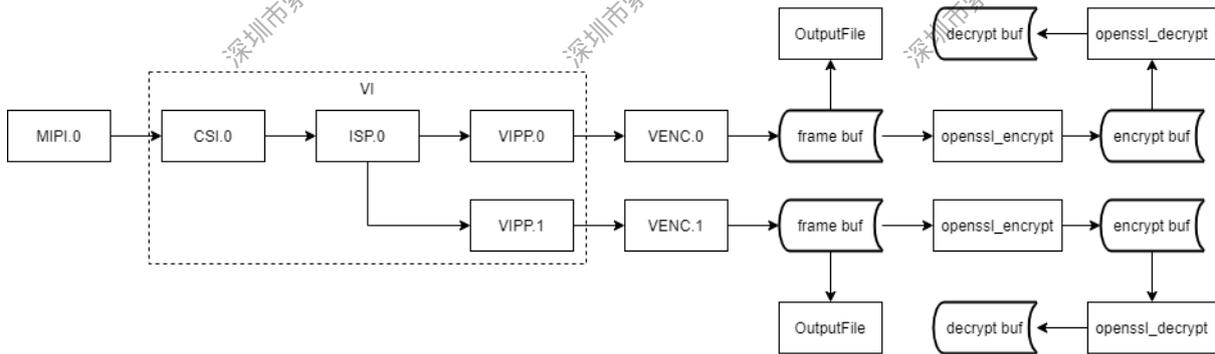


图 8-47: MPP\_sample 测试通路-sample\_twinchn\_virvi2venc2ce

源文件：

无

目标文件：

/mnt/extsd/isp0\_1920\_1080\_60\_ctx\_saved.bin

参数配置：

dev\_node\_main: 指定主路vipp 设备号。  
dev\_node\_sub: 指定子路vipp 设备号。  
src\_size\_main: 指定主路源大小。3840: 3840x2160, 1080: 1920x1080, 720: 1280x720, 640: 640x360。  
src\_size\_sub: 指定子路源大小。3840: 3840x2160, 1080: 1920x1080, 720: 1280x720, 640: 640x360。  
video\_dst\_file\_main: 指定主路目标文件路径。  
video\_dst\_file\_sub: 指定子路目标文件路径。  
video\_framerate\_main: 指定主路视频帧率, 单位: fps。  
video\_framerate\_sub: 指定子路视频帧率, 单位: fps。  
video\_bitrate\_main: 指定主路视频码率, 单位: bps。  
video\_bitrate\_sub: 指定子路视频码率, 单位: bps。  
video\_size\_main: 指定主路视频目标大小。3840: 3840x2160, 1080: 1920x1080, 720: 1280x720, 640: 640x360。  
video\_size\_sub: 指定子路视频目标大小。3840: 3840x2160, 1080: 1920x1080, 720: 1280x720, 640: 640x360。  
video\_encoder\_main: 指定主路编码格式。"H.264", "H.265"。  
video\_encoder\_sub: 指定子路编码格式。"H.264", "H.265"。  
rc\_mode\_main: 指定主路rc mode。0: CBR, 1: VBR, 2: FIXQP, 3: QPMAP。  
rc\_mode\_sub: 指定子路rc mode。0: CBR, 1: VBR, 2: FIXQP, 3: QPMAP。  
enable\_fast\_enc\_main: 指定主路fast enc 是否开启。1: enable, 0: disable。  
enable\_fast\_enc\_sub: 指定子路fast enc 是否开启。1: enable, 0: disable。  
enable\_roi\_main: 指定主路roi 是否开启。1: enable, 0: disable。  
enable\_roi\_sub: 指定子路roi 是否开启。1: enable, 0: disable。  
video\_duration\_main: 指定主路视频时长, 单位: s。  
video\_duration\_sub: 指定子路视频时长, 单位: s。  
test\_duration: 指定测试时长, 单位: s。  
color2grey: 彩转灰。yes: 开启, no: 关闭。  
3dnr: 3D去噪声。yes: 开启, no: 关闭。

测试指令：

**【Tina】**

```
cd /mnt/extsd/
./sample_twinchn_virvi2venc2ce -path ./sample_twinchn_virvi2venc2ce.conf
```

**【Melis】**

暂不支持。

**退出测试：**

测试达到设定的时间后自动退出测试。

**预期结果：**

1. 测试程序运行正常，测试过程没有异常打印。
2. 使用VLC 软件分析生成的raw文件正常。

## 8.6.2 sample\_virvi2venc2ce

**测试目的：**

该sample 测试一路编码的加解密。  
mpi\_vi 和mpi\_venc 组件绑定组合。mpi\_vi采集图像，直接传输给mpi\_venc 进行编码，在进行ce加密解密。

**组件依赖：**

PACKAGE\_libopenssl

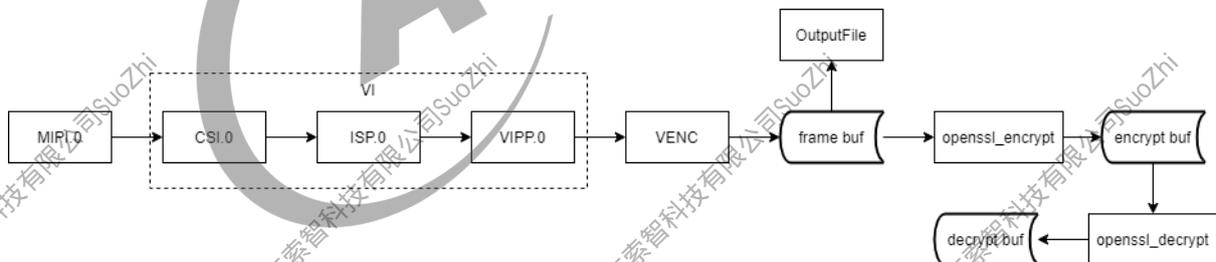
**测试通路：**

图 8-48: MPP\_sample 测试通路-sample\_virvi2venc2ce

**源文件：**

无

**目标文件：**

/mnt/extsd/AW\_VirviEncoder.H264

**参数配置：**

auto\_test\_count:指定自动测试次数  
encoder\_count:指定每次测试编码帧数  
dev\_num:指定VI Dev设备节点  
src\_width:指定camera采集的图像宽度  
src\_height:指定camera采集的图像高度  
src\_frame\_rate:指定camera采集图像的帧率  
dest\_encoder\_type:指定编码格式  
dest\_width:指定编码后图像的宽度  
dest\_height:指定编码后图像的高度  
dest\_frame\_rate:指定编码帧率  
dest\_bit\_rate:指定编码码率  
dest\_pic\_format:指定编码后图像的格式  
output\_file\_path:指定编码后视频文件的路径

测试指令:

【Tina】

```
cd /mnt/extsd/
./sample_virvi2venc2ce -path ./sample_virvi2venc2ce.conf
```

【Melis】

暂不支持。

退出测试:

达到指定测试次数后自动退出测试。

预期结果:

1. 测试程序运行正常，测试过程没有异常打印。
2. 使用VLC 软件分析生成的raw文件正常。

## 8.7 UVC 和 UAC

### 8.7.1 sample\_uvc2vdec\_vo

测试目的:

测试mpi\_uvc->mpi\_vdec的绑定方式传输数据，mpi\_vdec->mpi\_vo采用非绑定方式。  
从mpi\_uvc组件获取jpeg编码格式图片，交给mpi\_vdec组件解码，配置vdec解码缩小，并且大小图两路输出，  
mpi\_vdec的输出采用非绑定方式，app主动获取大小图，再交给mpi\_vo组件的两个图层分别显示。

组件依赖:

```
mpp_uvc
mpp_hw_display
mpp_vo
mpp_vdec
```

测试通路:

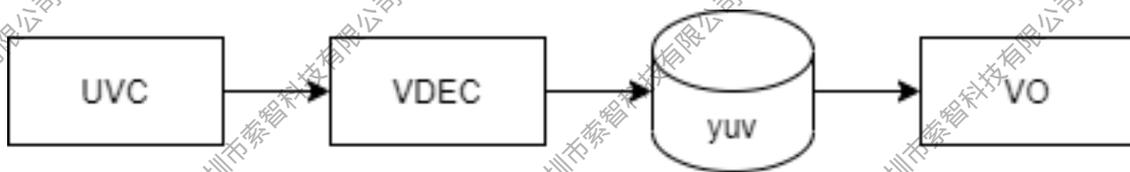


图 8-49: MPP\_sample 测试通路-sample\_uvc2vdec\_vo

源文件：

无

目标文件：

无

参数配置：

```

dev_name: uvc设备字符串
pic_format: uvc输出图像格式
capture_width: uvc输出图像宽度
capture_height: uvc输出图像高度
capture_framerate: uvc采集帧率
decode_sub_out_width: vdec解码输出的小图宽度
decode_sub_out_height: vdec解码输出的小图高度
display_main_x: 主图显示区域的左上角起点坐标x
display_main_y: 主图显示区域的左上角起点坐标y
display_main_width: 主图显示区域宽度
display_main_height: 主图显示区域高度
display_sub_x: 子图显示区域的左上角起点坐标x
display_sub_y: 子图显示区域的左上角起点坐标y
display_sub_width: 子图显示区域宽度
display_sub_height: 子图显示区域高度
test_frame_count: 测试帧数, 0代表无限。

```

测试指令：

【Tina】

```

cd /mnt/extsd/
./sample_uvc2vdec_vo -path ./sample_uvc2vdec_vo.conf

```

【Melis】

暂不支持。

退出测试：

达到测试帧数后自动退出测试，或者按“ctrl + c”退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. LCD屏幕正常显示。

## 8.7.2 sample\_uvc2vdec2vo

### 测试目的：

测试mpi\_uvc->mpi\_vdec的绑定方式传输数据，mpi\_vdec->mpi\_vo也采用绑定方式。  
从mpi\_uvc组件获取mjpeg编码格式图片，交给mpi\_vdec组件解码，再传给mpi\_vo组件显示。

### 组件依赖：

```
mpp_uvc
mpp_hw_display
mpp_vo
mpp_vdec
```

### 测试通路：



图 8-50: MPP\_sample 测试通路-sample\_uvc2vdec2vo

### 源文件：

无

### 目标文件：

无

### 参数配置：

```
dev_name: uvc设备字符串
pic_format: uvc输出图像格式
capture_width: uvc输出图像宽度
capture_height: uvc输出图像高度
capture_framerate: uvc采集帧率
decode_out_width: vdec解码输出图像的宽度
decode_out_height: vdec解码输出图像的高度
display_width: 显示宽度
display_height: 显示高度
test_duration: 测试时间, 0代表无限。
```

### 测试指令：

#### 【Tina】

```
cd /mnt/extsd/
./sample_uvc2vdec2vo -path ./sample_uvc2vdec2vo.conf
```

#### 【Melis】

暂不支持。

### 退出测试：

达到测试帧数后自动退出测试，或者按“ctrl + c”退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. LCD屏幕正常显示。

### 8.7.3 sample\_uvc2vo

测试目的：

测试mpi\_uvc组件的绑定方式输出。  
mpi\_uvc组件输出yuyv格式（或其他raw格式）到mpi\_vo组件显示。mpi\_uvc和mpi\_vo组件采用绑定方式连接。

组件依赖：

mpi\_uvc  
mpi\_hw\_display  
mpi\_vo

测试通路：

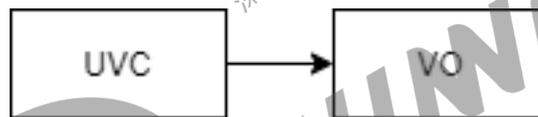


图 8-51: MPP\_sample 测试通路-sample\_uvc2vo

源文件：

无

目标文件：

无

参数配置：

dev\_name: uvc设备字符串  
pic\_format: uvc输出图像格式  
capture\_width: uvc输出图像宽度  
capture\_height: uvc输出图像高度  
display\_width: 显示宽度  
display\_height: 显示高度  
test\_duration: 测试时间，单位秒  
brightness: uvc亮度设置  
contrast: uvc对比度设置  
hue: uvc色度设置  
saturation: uvc饱和度设置  
sharpness: uvc锐度设置

测试指令：

【Tina】

```
cd /mnt/extsd/
./sample_uvc2vo -path ./sample_uvc2vo.conf
```

### 【Melis】

暂不支持。

### 退出测试：

达到测试帧数后自动退出测试，或者按“ctrl + c”退出测试。

### 预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. LCD屏幕正常显示。

## 8.7.4 sample\_uvc\_vo

### 测试目的：

测试mpi\_uvc组件的非绑定方式输出。  
从mpi\_uvc组件获取yuyv格式（或其他raw格式），交给mpi\_vo组件显示。mpi\_uvc和mpi\_vo组件采用非绑定方式运行。

### 组件依赖：

```
mpp_uvc
mpp_hw_display
mpp_vo
```

### 测试通路：



图 8-52: MPP\_sample 测试通路-sample\_uvc\_vo

### 源文件：

无

### 目标文件：

无

### 参数配置：

```
dev_name: uvc设备字符串
pic_format: uvc输出图像格式
capture_width: uvc输出图像宽度
capture_height: uvc输出图像高度
display_width: 显示宽度
display_height: 显示高度
```

test\_frame\_count:测试帧数，0代表无限帧数。  
 brightness:uvc亮度设置  
 contrast:uvc对比度设置  
 hue:uvc色度设置  
 saturation:uvc饱和度设置  
 sharpness:uvc锐度设置

测试指令：

【Tina】

```
cd /mnt/extsd/
./sample_uvc_vo -path ./sample_uvc_vo.conf
```

【Melis】

暂不支持。

退出测试：

达到测试帧数后自动退出测试，或者按“ctrl + c”退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. LCD屏幕正常显示。

## 8.7.5 sample\_uvcout

测试目的：

读取jpg文件，反复送由UVC设备输出到电脑（注意：不是获取UVC摄像头的的数据，而是把整个板子当作UVC摄像头，将图像送由电脑显示）电脑端使用VLC工具查看。

组件依赖：

mpp\_uvc

测试通路：

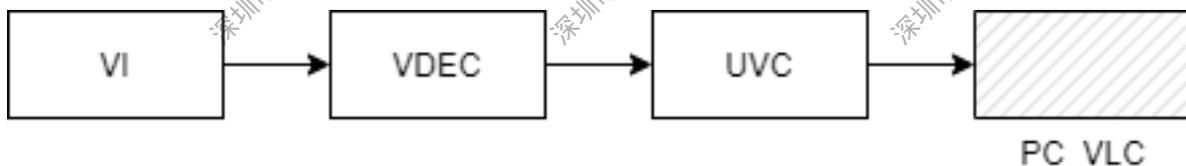


图 8-53: MPP\_sample 测试通路-sample\_uvcout

源文件：

无

目标文件：

无

### 参数配置：

```
uvc_dev: uvc设备号
vin_dev: 摄像头数据采集设备号
cap_format: 数据采集格式, nv21, nv12, yu12, yv12
cap_width: 图像采集宽度
cap_height: 图像采集高度
cap_frame_rate: 图像采集帧率
encoder_type: 编码类型
enc_frame_quality: 编码质量
enc_width: 编码宽度
enc_height: 编码高度
enc_frame_rate: 编码帧率
enc_bit_rate: 编码bit率 4M(4194304) 8M(8388608)
use_eve: 是否使用人脸识别, 1-使用, 0-不使用
```

### 测试指令：

#### 【Tina】

1. 执行命令生成video设备节点  
执行run\_otg脚本生成设备节点  
./run\_otg
2. 启动数据传输（uvc设备号要改为上述步骤生成的video节点）  
cd /mnt/extsd/  
./sample\_uvcout -path ./sample\_uvcout.conf
3. 使用VLC进行查看

#### 【Melis】

暂不支持。

### 退出测试：

按“ctrl + c”退出测试。

### 预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. PC端使用VLC工具播放正常。

### 详细配置步骤：

```
==>内核配置：
make kernel_menuconfig
[*] Device Driver --->
[*] USB support --->
[*] USB Gadget Support --->
[*] USB functions configurable through configs
[*] USB Webcam function

==>删除掉adb 的初始化，防止adb 影响webcam 的功能
rm target/allwinner/v459-perf1/busbox-init-base-files/etc/init.d/S50usb

==>编译sample_uvcout
```

```
vim /softwinner/eyesee-mpp/middleware/v459/tina.mk
#去掉下面行的注释:
make -c sample/sample_uvcout -f tina.mk all
编译mkmp, sample_uvcout目录生成sample_uvcout。

==>拷贝sample_uvcout/run_otg 脚本到小机
==>确定小机生成新的video节点

ls /dev/video*
/dev/video0 /dev/video1 /dev/video2

#初始化usb webcam 驱动,生成新的video节点,用于传输数据
./run_otg

==>#查看新增video节点
ls /dev/video*
/dev/video0 /dev/video1 /dev/video2/dev/video3

==>#确定新节点,是video4
==>修改sample_uvcout.conf 配置文件
vi sample/sample_uvcout/sample_uvcout.conf

[parameter]
uvc_dev = 2 #将新增节点的序号加入这里。
在小机上运行sample

==>./sample_uvcout -path ./sample_uvcout.conf

==>电脑查看,可使用VLC
[媒体] -->
[打开捕获设备] -->
捕获模式[DirectShow] 适配器设备名称 [UVC camera]
```

## 8.7.6 sample\_uac

测试目的:

测试UAC功能。

组件依赖:

无

测试通路:

无

源文件:

sample\_ai\_pcm.wav

目标文件:

无

## 参数配置：

```
pcm_file_path = "/mnt/extsd/sample_ai_pcm.wav"
pcm_sample_rate = 48000
pcm_channel_cnt = 1
pcm_bit_width = 16
pcm_frame_size =
pcm_cap_duration = 60
pcm_ai_gain = 23
uac_audio_card_name = "hw:UAC1Gadget"
```

## 环境准备和测试步骤：

## 1. 需要开启内核配置

```
CONFIG_USB_CONFIGFS_F_UAC1
```

```
CONFIG_SND_PROC_FS
```

## 2. 执行以下命令打开USB的UAC功能

```
./setusbconfig-configfs uac1
```

## 3. 此时，PC上将会创建一个新的audio（AC接口），同时板端将创建一个新的声卡。

检查声卡的方法：

```

cat /proc/asound/cards
0 [sun8iw19codec]: sun8iw19-codec - sun8iw19-codec
 sun8iw19-codec
1 [snddaudio0]: snddaudio0 - snddaudio0
 snddaudio0
2 [UAC1Gadget]: UAC1_Gadget - UAC1_Gadget
 UAC1_Gadget 0

```

其中，“2 [UAC1Gadget ]”就是UAC声卡。

此时，需要修改配置文件sample\_uac.conf中uac\_audio\_card\_name为"hw:UAC1Gadget"。

## 4. 执行sample，测试通路是：mic --&gt; uac --&gt; PC

```
./sample_uac -path ./sample_uac.conf
```

## 5. 在PC上使用音频工具抓取UAC的PCM数据验证测试结果。

## 测试指令：

## 【Tina】

```
cd /mnt/extsd/
./sample_uac -path ./sample_uac.conf
```

## 【Melis】

暂不支持。

## 结束测试：

达到设定的测试时间后，自动退出测试。

## 预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 按以上方法在PC上使用音频工具抓取UAC的PCM数据验证测试结果符合预期。

## 8.7.7 sample\_usbcamera

测试目的：

测试UVC、UAC复合设备。

组件依赖：

无

测试通路：

无

源文件：

无

目标文件：

无

参数配置：

```
uvc_dev: uvc设备号
vin_dev: 摄像头数据采集设备号
enc_bit_rate: JPEG、H264模式编码码率
enable_uac: 是否启用测试uac功能
uac_dev: uac设备号
uac_sample_rate: 音频采样率
uac_chn_cnt: 音频通道号
uac_bit_width: 音频位宽
uac_ai_gain: 采集音频音量
```

测试使用步骤

uvc内核配置：

```
CONFIG_USB_CONFIGFS_F_UVC=y
```

uac内核配置：

```
CONFIG_USB_CONFIGFS_F_UAC1=y
CONFIG_SND_PROC_FS=y
```

初始化uvc、uac复合设备：

```
setusbconfig uvc,uac1
```

确认生成节点：

```
==>#查看新增video节点
ls /dev/video*
/dev/video0 /dev/video1 /dev/video2 /dev/video3
```

```
==>#确定新节点,是video4
==>修改sample_uvcout.conf 配置文件
```

```
vi sample/sample_uvcout/sample_uvcout.conf
```

```
[parameter]
```

```
uvc_dev = 2 #将新增节点的序号加入这里。
```

```
cat /proc/asound/cards
0 [sun8iw19codec]: sun8iw19-codec - sun8iw19-codec
sun8iw19-codec
1 [snddaudio0]: snddaudio0 - snddaudio0
snddaudio0
2 [UAC1Gadget]: UAC1_Gadget - UAC1_Gadget
UAC1_Gadget 0

enable_uac = 1
uac_audio_card_name = "hw:UAC1Gadget"

运行sample
./sample_usbcamera -path /mnt/extsd/sample_usbcamera.conf
```

注：

uvc、uac复合设备只能在uvc bulk模式下使用，sample已改为usb bulk模式。

测试指令：

【Tina】

```
cd /mnt/extsd/
./sample_usbcamera -path ./sample_usbcamera.conf
```

【Melis】

暂不支持。

结束测试：

达到设定的测试时间后，自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 测试UVC、UAC复合设备功能正常。

## 8.8 多媒体文件

### 8.8.1 sample\_demux

测试目的：

从视频文件（如：test.mp4）中分离出所有的视频、音频、subtitle数据，分别组成对应文件。

组件依赖：

mpp\_demuxer

测试通路：

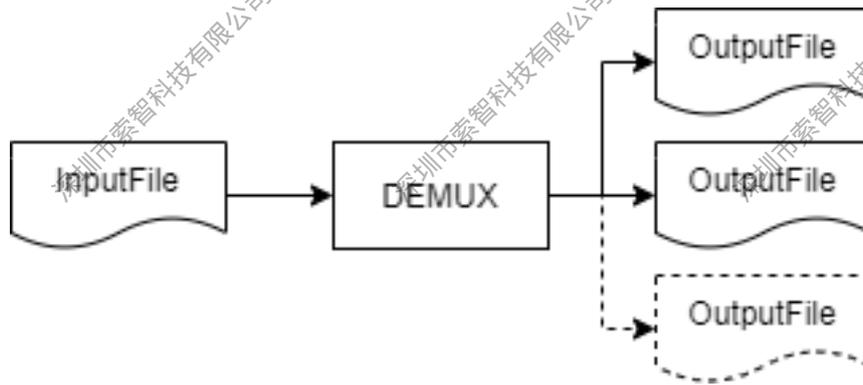


图 8-54: MPP\_sample 测试通路-sample\_demux

源文件:

/mnt/extsd/two\_video.mp4

目标文件:

/mnt/extsd/video.bin  
/mnt/extsd/audio.bin

参数配置:

src\_file: 指定原始视频文件的路径  
 src\_size: 指定原始视频文件的视频大小, 如1080p  
 seek\_position: 指定原始视频文件的开始解析位置(ms)  
 video\_dst\_file: 解析出来的视频数据生成的文件路径  
 audio\_dst\_file: 解析出来的音频数据生成的文件路径  
 subtitle\_dst\_file: 解析出来的文字数据生成的文件路径  
 test\_duration: sample一次测试时间(单位: s)

测试指令:

【Tina】

```
cd /mnt/extsd/
./sample_demux -path ./sample_demux.conf
```

【Melis】

暂不支持。

退出测试:

测试完文件two\_video.mp4, 可以自动退出测试。

预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 生成解析文件 video.bin 和audio.bin。

## 8.8.2 sample\_file repair

测试目的：

该sample 用来修复mp4 文件。

组件依赖：

mpp\_muxer

测试通路：

无

源文件：

/mnt/extsd/1080p.mp4

目标文件：

覆盖原文件 /mnt/extsd/1080p.mp4

参数配置：

无

测试指令：

**[Tina]**

```
cd /mnt/extsd/
./sample_file_repair 1080p.mp4
```

**[Melis]**

暂不支持。

退出测试：

修复完源文件，会自动退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 使用VLC 软件播放修复后的mp4文件正常。

## 8.9 AI demo

### 8.9.1 sample\_odet\_demo

测试目的：

演示AI demo方案。

该方案共有四路主应用，分别为：

- 第一路：支持H265高清编码存盘。
- 第二路：支持640\*480预览。
- 第三路：支持NPU检测，返回坐标给其它两路绘框。
- 第四路：支持抓拍。

组件依赖：

```
mpp_vi
mpp_venc
mpp_vo
```

测试通路：

无

源文件：

无

目标文件：

无

参数配置：

环境准备

#### 1. 准备TF卡存储模型

TF卡的作用是存储模型文件和保存H265编码数据。

AI Demo用例对模型文件的存储位置有要求，需要将模型文件存储在TF的指定位置，在本用例中，这个位置是：

```
/mnt/extsd/object_det/model/
```

#### 2. 挂载存储卡到系统

执行如下命令将其挂载到 /mnt/extsd目录下：

```
挂载命令：mount -t vfat /dev/mmcblk1p1 /mnt/extsd
```

挂载成功后，执行 `ls -l /mnt/extsd/object_det/model/` 可以看到模型文件

#### 3. AI应用放入TF卡

有两种办法将测试AI应用放入SD卡，分别介绍：

第一种，即使用读卡器手动将应用拷贝到TF卡根目录。

第二种，通过ADB将测试AI应用推到平台端系统，这种方式不需要将TF卡从平台端取下来，在线即可完成。

测试指令：

**[Tina]**

#### 1. 启动OSD显示应用

OSD显示应用的主要作用是对当前的目标检测应用的统计信息进行显示，主要包括检测类别，目标检测个数，NPU检测帧率等等，受限于LCD尺寸的限制，部分信息，比如检测概率通过串口输出。

OSD应用已经随Tina系统打包进固件里面，不需要像AI用例一样手工拷贝或者adb推入，所以可以直接在控制台下执行 `df_andi&` 即可。

#### 2. 启动AI Demo应用

```
cd /mnt/extsd/
./sample_odet_demo
```

**[Melis]**

暂不支持。

结束测试：

按"ctrl + c"退出测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 录制写卡功能正常。
3. 拍照功能正常。
4. LCD屏幕预览正常，同时显示人脸识别正常。

## 8.9.2 sample\_RegionDetect

测试目的：

演示区域检测功能。

组件依赖：

mpp\_vi  
mpp\_venc

测试通路：

无

源文件：

无

目标文件：

无

参数配置：

环境准备

1. 板端接好camera和屏幕。
2. 将motion目录复制到tf卡中。

motion目录的文件：

aw-pdet.nb: 模型文件 (在mpp\_sample\_RegionDetect目录下)  
libpdet.so: 网络级运行库 (在mpp\_sample\_RegionDetect目录下)  
libVIPlite.so: NPU用户态运行库 (单独申请提供)  
libVIPuser.so: NPU用户态BSP运行库 (单独申请提供)  
run.sh: demo执行脚本 (见下面的备注)  
sample\_RegionDetect: 编译生成的demo程序

备注：

run.sh脚本的内容：

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:$PWD
./sample_RegionDetect $@
```

测试指令：

**【Tina】**

## demo演示

1. shell进入/mnt/extsd/motion目录。
2. 执行./run.sh N开始区域检测demo演示，其中NE{0, 1, 2, 3}。分别为：移动侦测，区域检测，越界检测，人形检测。
3. 检测demo会显示预览界面，其中红色为配置区域或界线，绿色正方形小框显示移动侦测区域，绿色矩形框为人形区域，中心为绿色小点。越界算法以人形中心点为准计算，蓝色点为越界坐标。
4. 录像自动保存为motion.h265。 5. 查看log打印，区域检测打印为：<on\_cross\_callback> on\_cross\_callback: RECT[0] type: 1; 其中RECT[0]表示第0个区域，type: 1表示进入区域，2表示离开区域。越界检测打印为：<on\_cross\_callback> on\_cross\_callback: LINE[0] type: 1。其中LINE[0]表示第一条线，type:1表示从左到右，2表示从右到左。

**【Melis】**

暂不支持。

## 结束测试：

按"ctrl + c" 退出测试。

## 预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 查看预览界面，其中红色为配置区域或界线，绿色正方形小框显示移动侦测区域，绿色矩形框为人形区域，中心为绿色小点。

## 8.10 其他

### 8.10.1 sample\_glog

## 测试目的：

该sample 演示glog (Google Logging Library) 的使用方法。

## 组件依赖：

无

## 测试通路：

无

## 源文件：

无

## 目标文件：

无

## 参数配置：

无

测试指令：

【Tina】

```
cd /mnt/extsd/
./sample_glog
```

【Melis】

暂不支持。

退出测试：

测试完后可以自动退出测试。

预期结果：

alogd、alogw、aloge 可以正常打印。

## 8.10.2 sample\_hello

测试目的：

该sample 用来验证mpp middleware sample 运行基本环境是否正常。

组件依赖：

无

测试通路：

无

源文件：

无

目标文件：

无

参数配置：

无

测试指令：

【Tina】

```
cd /mnt/extsd/
./sample_hello
```

【Melis】

暂不支持。

退出测试：

测试完后可以自动退出测试。

预期结果：

运行正常，不死机。

### 8.10.3 sample\_pthread\_cancel

测试目的：

该sample测试pthread\_cancel()对目标线程发送cancel信号后，目标线程接收cancel信号的处理过程。

组件依赖：

无

测试通路：

无

源文件：

无

目标文件：

无

参数配置：

无

测试指令：

【Tina】

```
cd /mnt/extsd/
./sample_pthread_cancel
```

【Melis】

暂不支持。

退出测试：

按“Enter”可以正常退出测试。

预期结果：

线程可以正常退出测试。

## 8.10.4 sample\_motor

测试目的：

【说明：该项测试暂不支持】

该sample 演示电机motor 测试。

组件依赖：

无

测试通路：

无

源文件：

无

目标文件：

无

参数配置：

无

测试指令：

【Tina】

```
cd /mnt/extsd/
./sample_motor
```

【Melis】

暂不支持。

退出测试：

自动结束测试。

预期结果：

无

## 8.10.5 sample\_sound\_controler

测试目的：

【说明：该项测试暂不支持】

演示语音识别。

1. 声控：支持的命令词：{ "小志开始录像", "小志停止录像", "小志拍照", "小志连拍", "小志关机" }
2. 每次送给音频320个字节音频数据
3. 包含的库和头文件在external/sound\_controler 目录下
4. 音频要求：  
pcm\_sample\_rate = 16000  
pcm\_bit\_width = 16

组件依赖：

```
mpp_external_sound_controler
mpp_aio
```

测试通路：

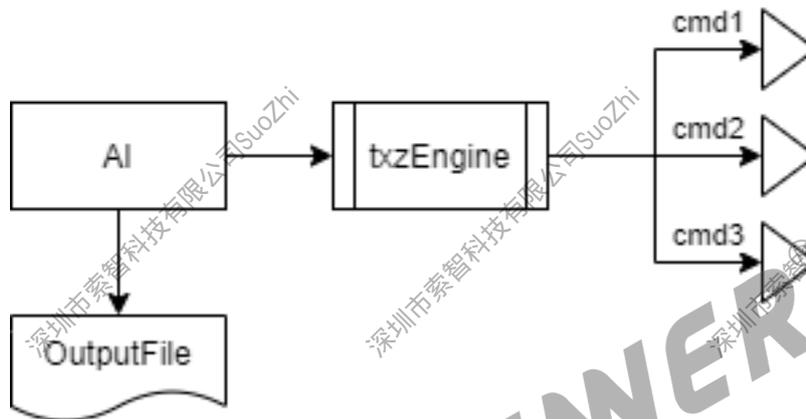


图 8-55: MPP\_sample 测试通路-sample\_sound\_controler

源文件：

无

目标文件：

```
/mnt/extsd/sample_sound_controler.pcm
```

参数配置：

```
pcm_file_path: 指定pcm 文件路径。
pcm_sample_rate: 指定采样率。
pcm_channel_cnt: 指定通道数。
pcm_bit_width: 指定位宽，必须设置为16。
pcm_frame_size: 指定pcm 帧的大小，单位：bytes。
pcm_cap_duration: 指定测试时间，单位：s。
```

测试指令：

【Tina】

```
cd /mnt/extsd/
./sample_sound_controler -path ./sample_sound_controler.conf
```

【Melis】

暂不支持。

退出测试：

达到设定测试时长后可以自动结束测试。

预期结果：

可以获得正确的控制指令，同时生成pcm 文件sample\_sound\_controler.pcm。

## 8.10.6 sample\_ai\_Demo\_0.9.2

智能算子demo

## 8.10.7 sample\_ai\_Demo\_0.9.6

智能算子demo

## 8.10.8 sample\_thumb

测试目的：

演示抓取缩略图。

组件依赖：

mpp\_vi  
mpp\_venc

测试通路：

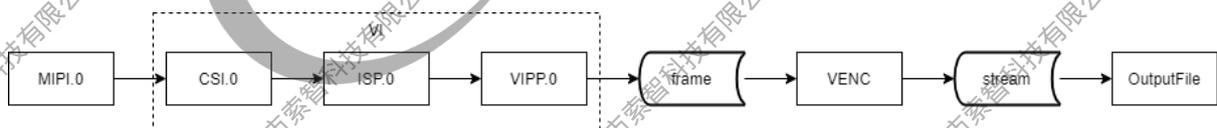


图 8-56: MPP\_sample 测试通路-sample\_thumb

源文件：

无

目标文件：

无

参数配置：

无

测试指令：

**[Tina]**

暂不支持。

**[Melis]**

```
sample_thumb
```

结束测试：

自动结束测试。

预期结果：

1. 测试程序运行正常，测试过程没有异常打印。
2. 打印缩略图的大小等信息符合预期。

## 8.10.9 sample\_nna

测试目的：

演示人形检测、人脸检测。

1. 将编译出来的sample\_nna放到nna\_g2d文件夹中
2. run ./sample\_nna 人形检测
3. run ./sample\_nna xx 人脸检测
4. 识别结果 nna\_g2d/out/nna.h264

组件依赖：

```
mpp_vi
mpp_venc
```

测试通路：

无

源文件：

图片源文件：nna\_g2d/c1080.jpg

目标文件：

生成的编码文件：nna\_g2d/out/nna.h264

参数配置：

无

测试指令：

**[Tina]**

暂不支持。

## 【Melis】

暂不支持。

## 结束测试：

检测完成后会自动结束测试。

## 预期结果：

生成编码文件nna\_g2d/out/nna.h264，标注人脸位置。

## 8.10.10 sample\_MotionDetect

## 测试目的：

该sample演示视频编码库的移动侦测功能的使用。

创建mpi\_vi和mpi\_venc，将它们绑定，再分别启动。mpi\_vi采集图像，直接传输给mpi\_venc进行编码。

编码过程中获取移动侦测信息，可保存裸码流视频文件。支持手动按ctrl-c，终止测试。

注意：视频编码驱动只支持在VBR，IPC-MODE模式下做移动侦测。

## 组件依赖：

mpp\_vi  
mpp\_venc

## 测试通路：

无

## 源文件：

无

## 目标文件：

/mnt/extsd/md.h265

## 参数配置：

online\_en：指定是否开启在线编码，默认0，表示不开启在线编码，即当前为离线编码。

online\_share\_buf\_num：指定在线编码的共享buffer个数（1或2），默认2，表示2个buffer。注意该配置仅当配置在线模式时才生效。

encoder\_count：指定每次测试编码帧数。

dev\_num：指定VI Dev设备节点。

src\_pic\_format：指定采集图像的格式。

src\_width：指定camera采集的图像宽度。

src\_height：指定camera采集的图像高度。

src\_frame\_rate：指定camera采集图像的帧率。

dst\_encoder\_type：指定编码格式。

dst\_width：指定编码后图像的宽度。

dst\_height：指定编码后图像的高度。

dst\_frame\_rate：指定编码帧率。

dst\_bit\_rate：指定编码码率。

output\_file\_path：指定编码后视频文件的路径。

**测试指令：****【Tina】**

```
cd /mnt/extsd/
./sample_MotionDetect -path ./sample_MotionDetect.conf
```

**【Melis】**

暂不支持。

**结束测试：**

达到设定的测试时间后，自动退出测试。

**预期结果：**

1. 测试程序运行正常，测试过程没有异常打印。
2. 使用VLC 软件分析生成的raw文件正常。

## 8.10.11 sample\_PersonDetect

**测试目的：**

该sample演示使用人形检测库，将检测的人形区域在主码流和子码流上分别画框、编码。  
针对主码流和子码流分别创建mpi\_vi和mpi\_venc，将它们绑定，再分别启动。mpi\_vi采集图像，直接传输给mpi\_venc进行编码。  
子码流的mpi\_vi再开一路虚通道用于人形检测，再开一路虚通道进行预览。

**组件依赖：**

```
mpp_vi
mpp_venc
```

**测试通路：**

无

**源文件：**

无

**目标文件：**

```
/mnt/extsd/mainStreamPd.raw
/mnt/extsd/subStreamPd.raw
```

**参数配置：**

```
main_vipp: 主码流VIPP号。
main_src_width: 主VIPP输出图像宽度。
main_src_height: 主VIPP输出图像高度。
main_pixel_format: 主VIPP输出图像格式。nv21, lbc25等。
sub_vipp: 子码流VIPP号。
sub_src_width: 子VIPP输出图像宽度。
```

sub\_src\_height: 子VIPP输出图像高度。  
sub\_pixel\_format: 子VIPP输出图像格式。nv21, lbc25等。  
src\_frame\_rate: 采集帧率。

orl\_color: 指定画人形框的颜色。  
enable\_main\_vipp\_orl: 是否在主VIPP画人形框。  
enable\_sub\_vipp\_orl: 是否在子VIPP画人形框。

preview\_vipp: 用于预览的vipp通道号。  
preview\_width: 预览的显示宽度, 0表示不预览。  
preview\_height: 预览的显示高度, 0表示不预览。

main\_encode\_type: 主码流编码类型, H.265, H.264等。  
main\_encode\_width: 主码流的编码目标宽度。  
main\_encode\_height: 主码流的编码目标高度。  
main\_encode\_frame\_rate: 主码流的编码目标帧率。  
main\_encode\_bitrate: 主码流的编码目标码率, 单位: bps。  
main\_file\_path: 保存主码流的文件路径, 空表示不保存主码流。  
sub\_encode\_type: 子码流编码类型, H.265, H.264等。  
sub\_encode\_width: 子码流的编码目标宽度。  
sub\_encode\_height: 子码流的编码目标高度。  
sub\_encode\_frame\_rate: 子码流的编码目标帧率。  
sub\_encode\_bitrate: 子码流的编码目标码率, 单位: bps。  
sub\_file\_path: 保存子码流的文件路径, 空表示不保存子码流。

test\_duration: 测试时间, 0表示无限。单位: 秒。

#### 测试指令:

##### [Tina]

```
cd /mnt/extsd/
./sample_PersonDetect -path ./sample_PersonDetect.conf
```

##### [Melis]

暂不支持。

#### 结束测试:

达到设定的测试时间后, 自动退出测试。

#### 预期结果:

1. 测试程序运行正常, 测试过程没有异常打印。
2. 使用VLC 软件分析生成的raw文件正常。

## 8.10.12 sample\_directIORead

#### 测试目的:

该sample演示使用directIO方式读文件。理论上不占内存作为缓存。

#### 组件依赖:

无

**测试通路：**

无

**源文件：**`/mnt/extsd/1080p.mp4`**目标文件：**`/mnt/extsd/1080p_rewrite.mp4`**参数配置：**

`src_file`：指定被读取的源文件。  
`dst_file`：读出的内容写入目标文件，可以在测试结束后和源文件比对。判断是否读取正确。如果为空，则读取内容不写入目标文件。  
`read_size`：指定一次读取的长度，单位字节。  
`read_interval`：指定读取的间隔时间，单位：ms。  
`loop_cnt`：指定循环读取文件的次数，0表示无限次。

**测试指令：****【Tina】**

```
cd /mnt/extsd/
./sample_directIORead -path ./sample_directIORead.conf
```

**【Melis】**

暂不支持。

**结束测试：**

达到设定测试次数后，自动退出测试。

**预期结果：**

1. 测试程序运行正常，测试过程没有异常打印。
2. 使用VLC 软件分析生成的mp4文件正常。

## 9 FAQ

本章节主要汇总 MPP sample 使用过程中可能遇到的一些问题及对应的解决方法。

### 9.1 某些方案上开机后执行某些 MPP sample 会运行失败

在某些客户分支方案上运行 MPP sample 之前，需要先 kill 掉客户的应用程序（比如：/usr/bin/lsdvcam），否则，可能出现与 MPP sample 测试的资源冲突而导致其运行失败。

### 9.2 MPP sample 测试时 SD 卡识别异常

情况一：

在测试一些录流的 MPP sample 之前，由于系统存储空间不足，需要准备 SD 卡。但是，有时会碰到识别到的 SD 卡空间太小，格式化 SD 卡也没用。此时，需要在 Linux 环境下用 dd 命令删除前面的分区。

情况二：

某些客户方案上，SD 卡默认没有 mount。mount 的方法：

```
mkdir /mnt/extsd/
mount -t vfat /dev/mmcblk0 /mnt/extsd/
或
mount -t vfat /dev/mmcblk0p1 /mnt/extsd/
```

### 9.3 视频编解码常用功能检查方法

#### 9.3.1 常用检测工具介绍

##### 9.3.1.1 码流播放软件

测试时，检查编码后的文件是否正常，一般需要使用 VLC 等播放软件看下是否有花屏、马赛克、卡顿等问题。



图 9-1: VLC 播放码流

### 9.3.1.2 码流分析软件

测试时，检查编码参数是否符合预期，一般需要使用 MediaInfo 软件。



图 9-2: MediaInfo 分析码流

提取 MediaInfo 分析码流的信息如下:

|                |                                           |
|----------------|-------------------------------------------|
| 概览             |                                           |
| 完整名称           | : Y:\tools\ffmpeg\1-编码格式\1-1080p-H264.mp4 |
| 格式             | : MPEG-4                                  |
| 格式配置 (Profile) | : Base Media                              |

```

编解码器 ID : isom (isom/iso2/mp41)
文件大小 : 15.5 MiB
时长 : 57 秒 648 毫秒
总体码率 : 2 252 kb/s

视频
ID : 1
格式 : AVC
格式/信息 : Advanced Video Codec
格式配置 (Profile) : High@L5.1
格式设置 : CABAC / 1 Ref Frames
格式设置, CABAC : 是
格式设置, 参考帧 : 1 帧
格式设置, GOP : M=1, N=50
编解码器 ID : avc1
编解码器 ID/信息 : Advanced Video Coding
时长 : 57 秒 648 毫秒
源, 时长 : 57 秒 599 毫秒
码率 : 2 106 kb/s
宽度 : 1 920 像素
高度 : 1 080 像素
画面比例 : 16:9
帧率 : 20.018 FPS
色彩空间 : YUV
色度抽样 : 4:2:0
位深 : 8 位
扫描类型 : 逐行扫描 (连续)
数据密度 [码率/(像素*帧率)] : 0.051
流大小 : 14.5 MiB (94%)
源, 流大小 : 14.5 MiB (94%)
语言 : 英语 (English)
色彩范围 : Limited
色彩原色 : BT.709
传输特性 : BT.709
矩阵系数 : BT.709
mdhd_Duration : 57648
编码配置区块 (box) : avcC

```

### 9.3.1.3 YUV 文件分析软件

分析 YUV 文件一般使用 YUView 软件。

为使工具分析能力更强，需要配置 ffmpeg 动态库。

YUView 工具配置 ffmpeg 动态库的方式：

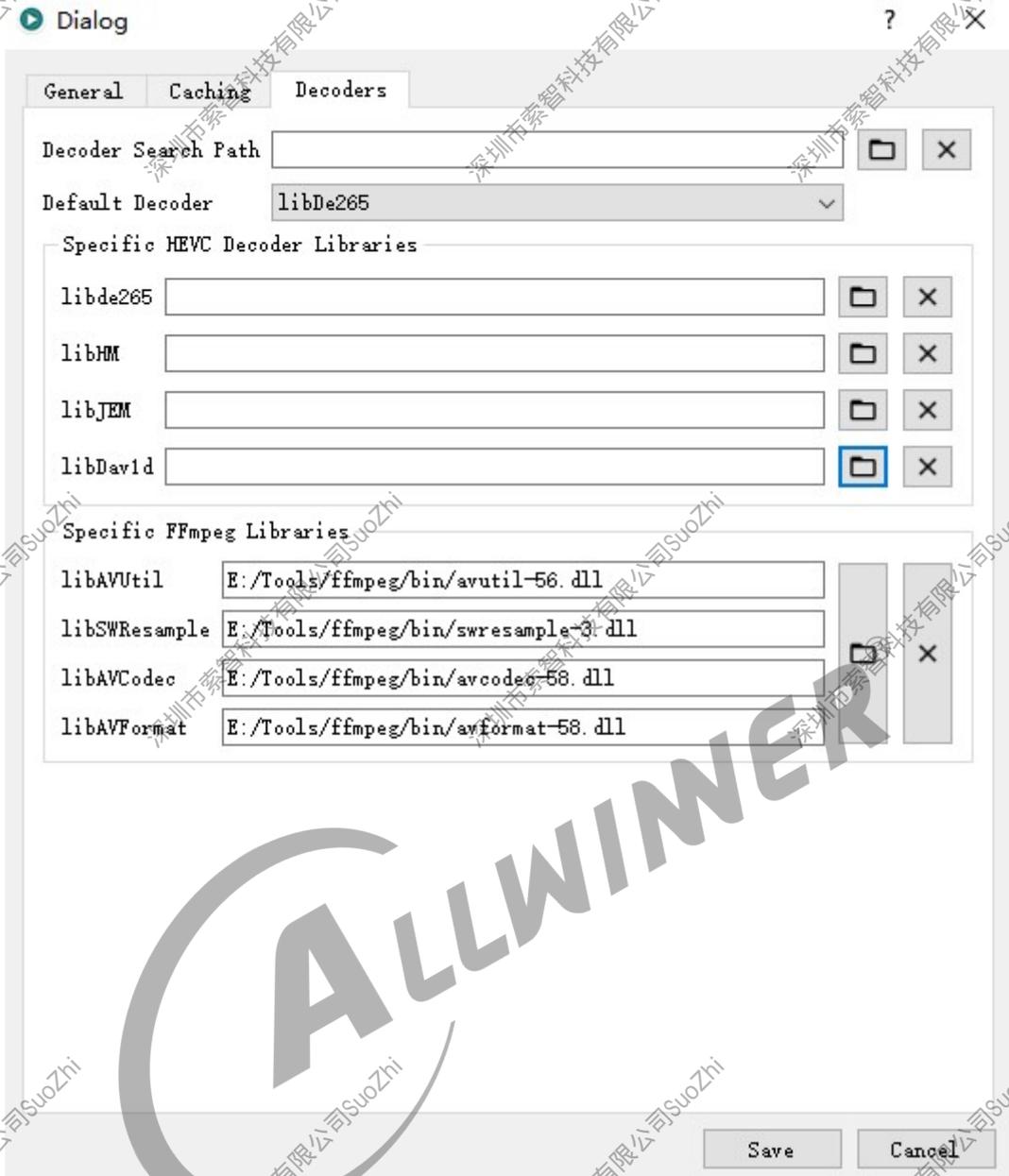


图 9-3: YUView 软件配置 ffmpeg 动态库

### 9.3.1.4 H264 码流分析工具

常用的 H264 码流分析工具：Elescard StreamEye。

该工具可解析帧类型信息、流的信息等，支持逐帧播放和分析编码层内容等。

## 9.3.2 典型的视频编解码功能检测示例

### 9.3.2.1 编码格式

使用 MediaInfo 软件检查编码格式。

- 格式: HEVC

### 9.3.2.2 彩转灰

使用 PC 软件 VLC 播放彩转灰测试生成的视频文件，效果如下：



图 9-4: 彩转灰效果

### 9.3.2.3 旋转编码

使用 PC 软件 VLC 播放测试生成的视频文件，效果如下：

#### 1. 第一组

- 不旋转、不翻转的效果
- 不旋转、翻转的效果

- 旋转 90 度、不翻转的效果
- 旋转 90 度、翻转的效果



图 9-5: 旋转-1

## 2. 第二组

- 旋转 180 度、不翻转的效果
- 旋转 180 度、翻转的效果
- 旋转 270 度、不翻转的效果
- 旋转 270 度、翻转的效果

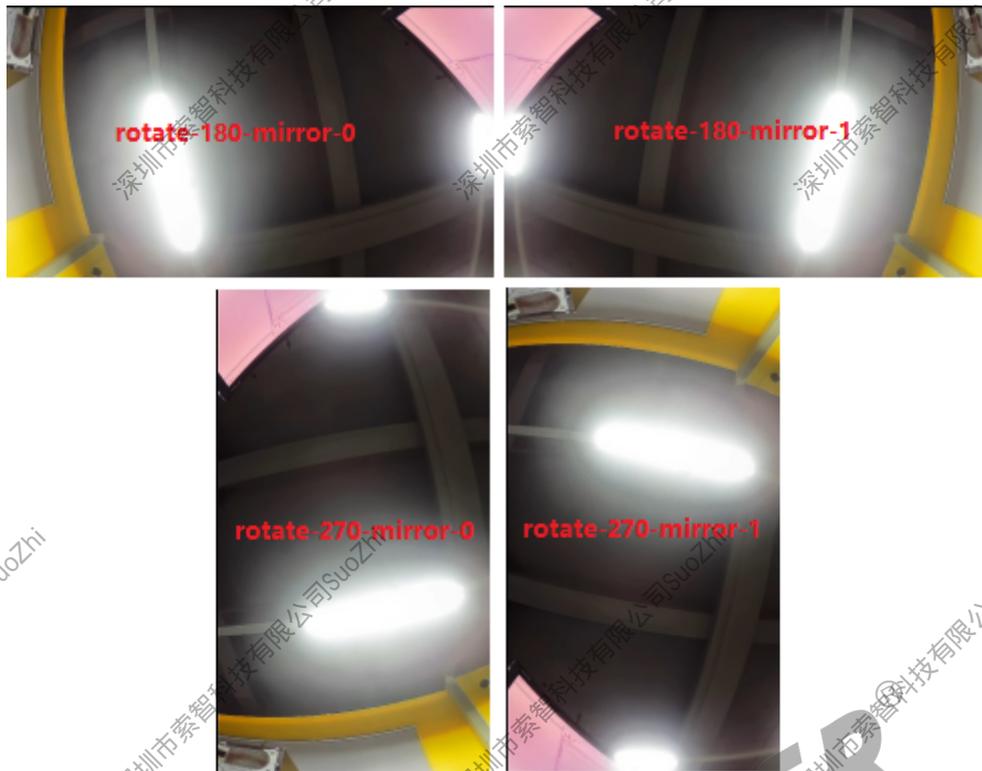


图 9-6: 旋转-2

### 9.3.2.4 P 帧帧内刷新

H264:

使用 Elescared StreamEye 4.6 工具（只支持 H264），按如下截图配置后，开启 P 帧帧内刷新后，图像看到一个橙色的竖状矩形条，同时逐帧往后查看时矩形条会从左往右移动；关闭 P 帧帧内刷新后，则无此矩形条。

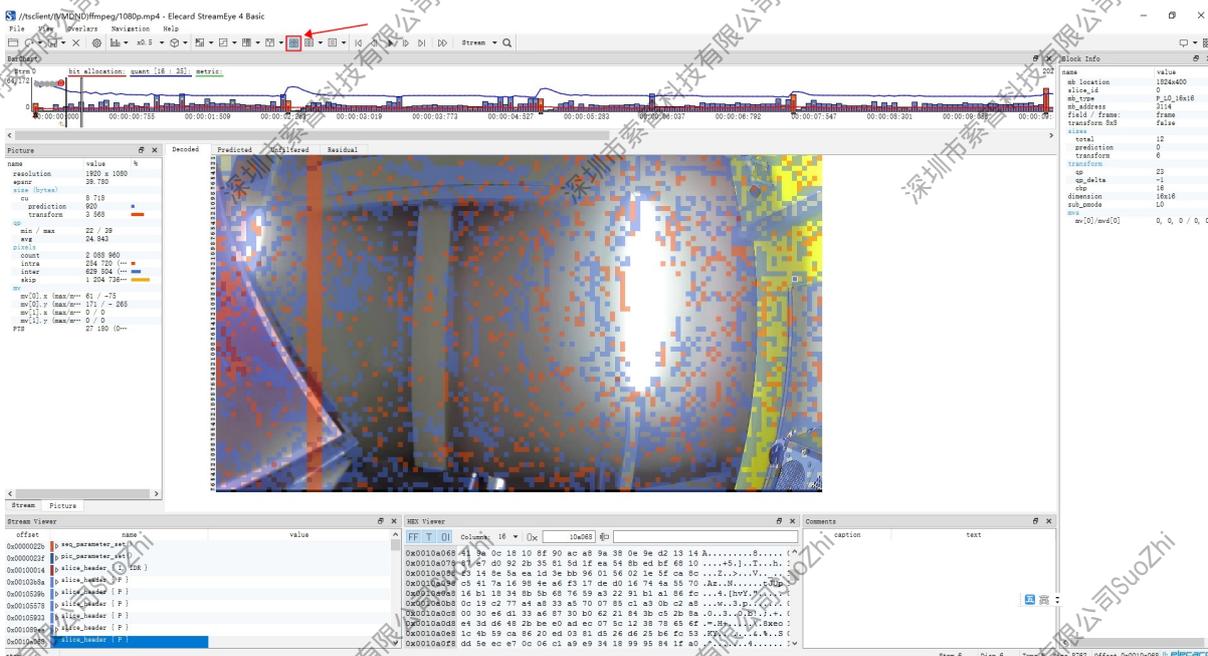


图 9-7: H264-开启 P 帧帧内刷新

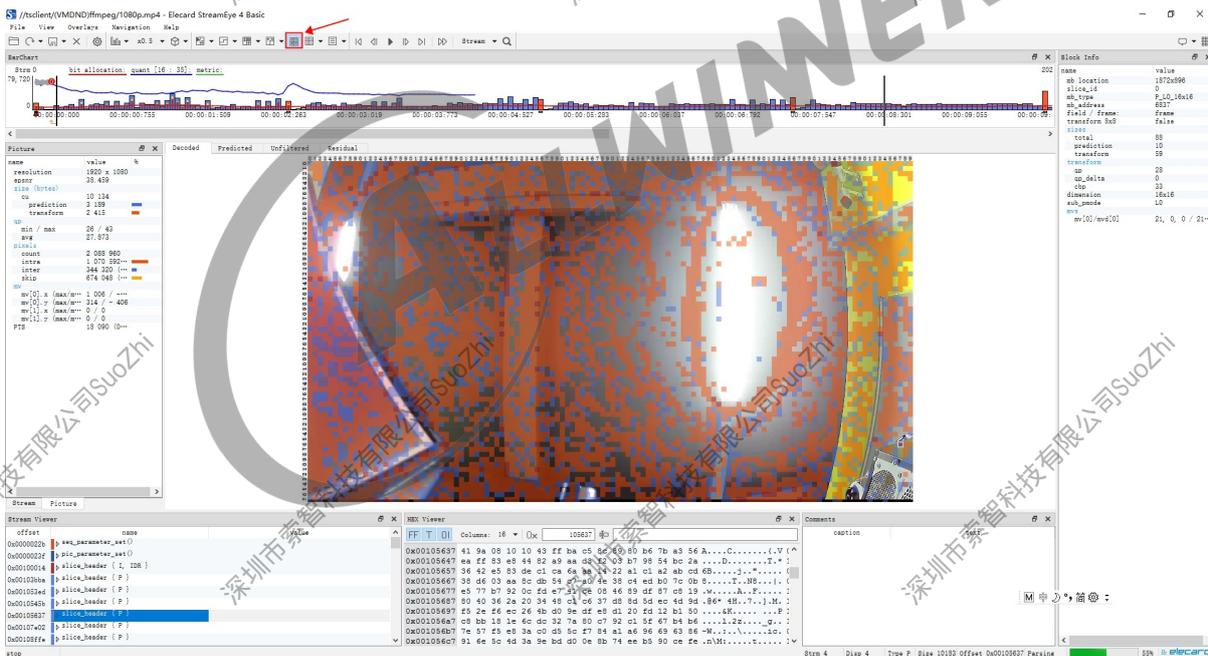


图 9-8: H264-关闭 P 帧帧内刷新

H265:

使用 YUView 工具（需要配置 ffmpeg 动态库）可分析 H265 文件，勾选 Pred Mode 后，可显示一个蓝色的竖状矩形条，同时逐帧往后查看时矩形条会从左往右移动；关闭 P 帧帧内刷新后，则无此矩形条。

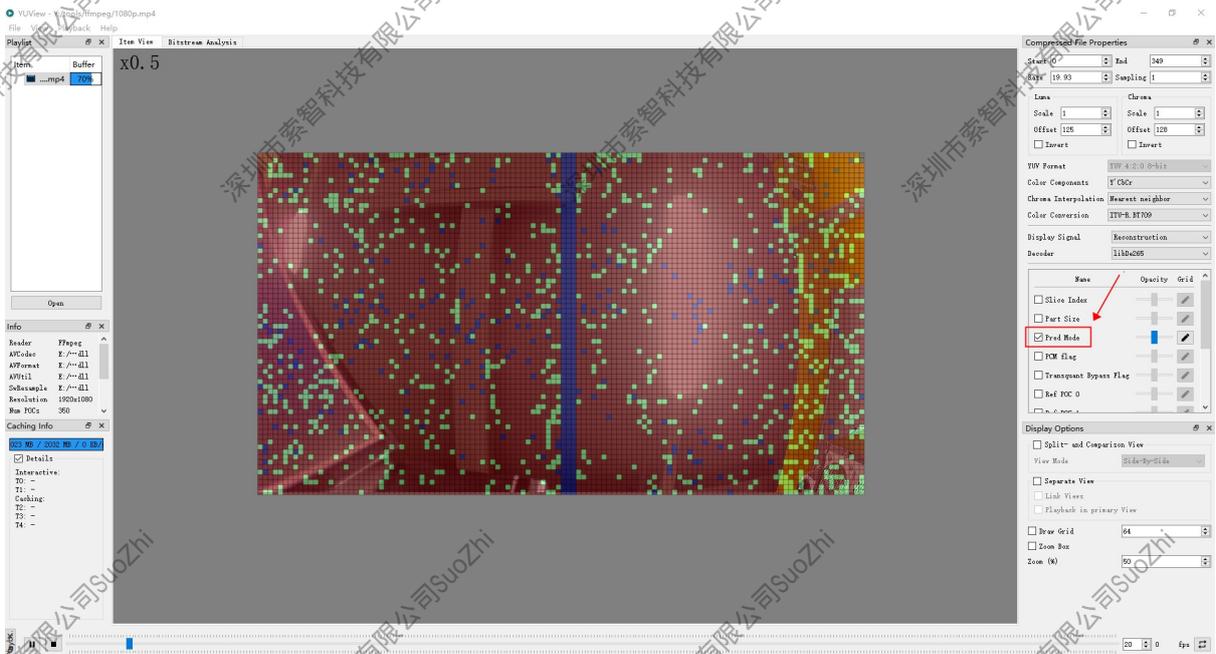


图 9-9: H265-开启 P 帧帧内刷新



图 9-10: H265-关闭 P 帧帧内刷新

### 9.3.2.5 视频编码 OSD

使用 PC 软件 VLC 播放测试生成的视频文件，效果如下：

1. 显示 OSD 的效果：

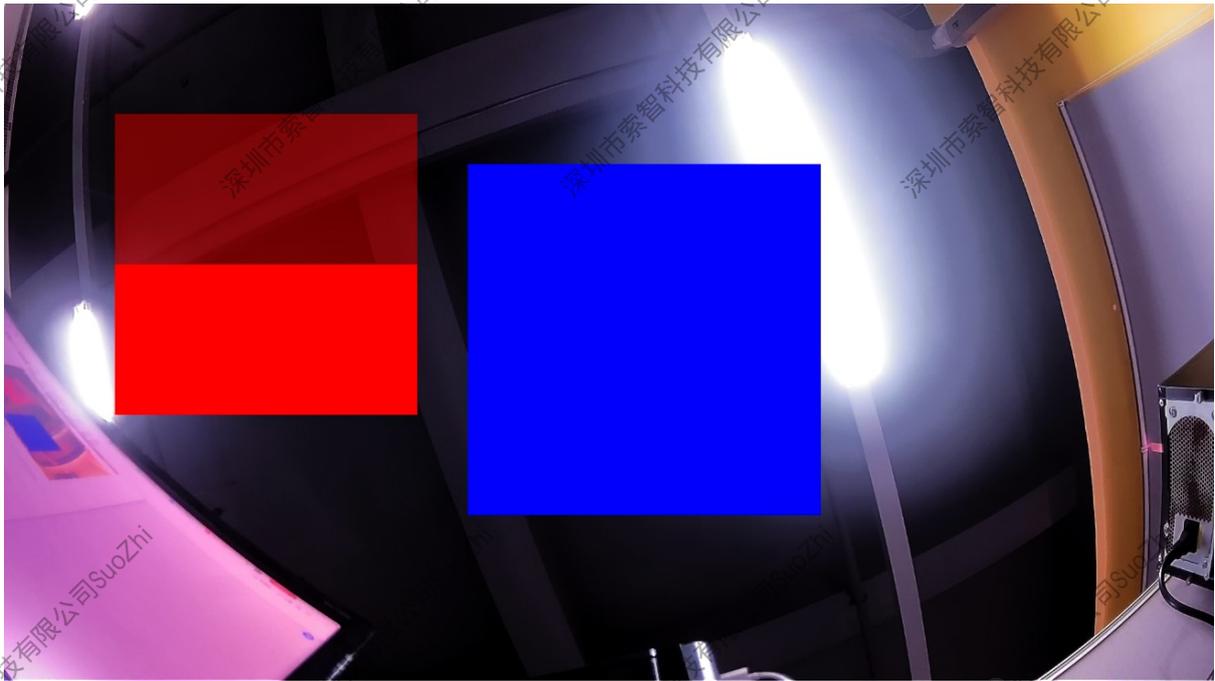


图 9-11: 编码 OSD-1

2. 改变 Region 位置后，显示 OSD 的效果：

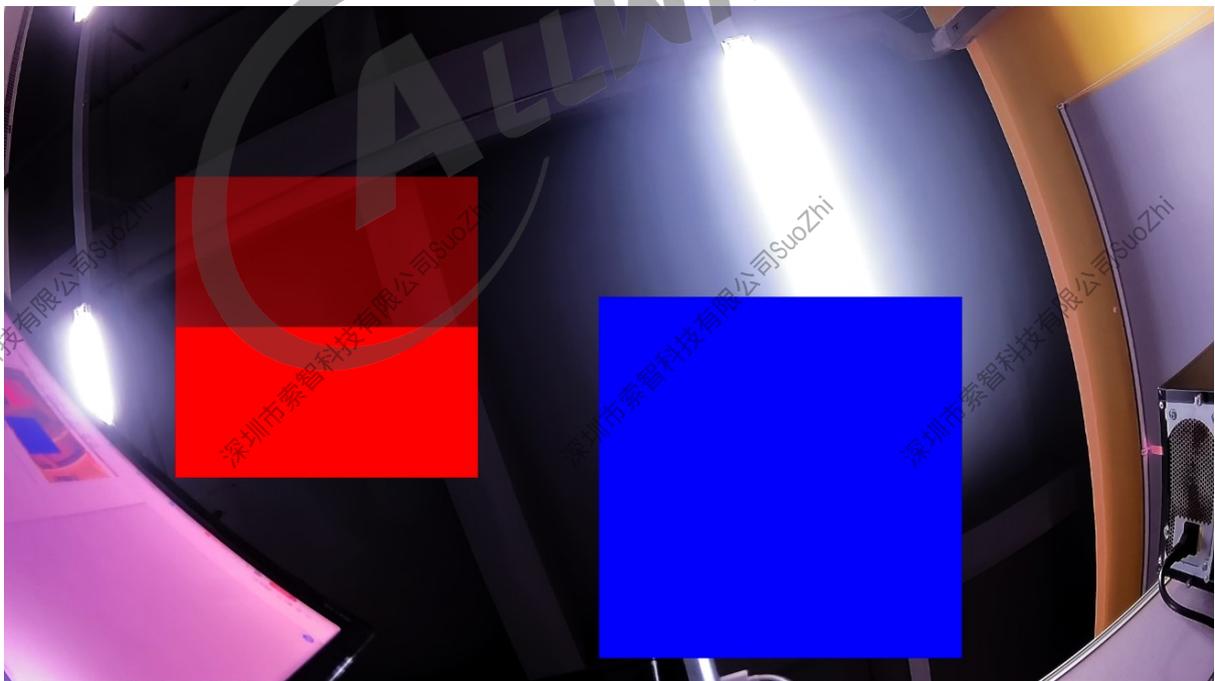


图 9-12: 编码 OSD-2

3. OSD 坐标和大小示意图

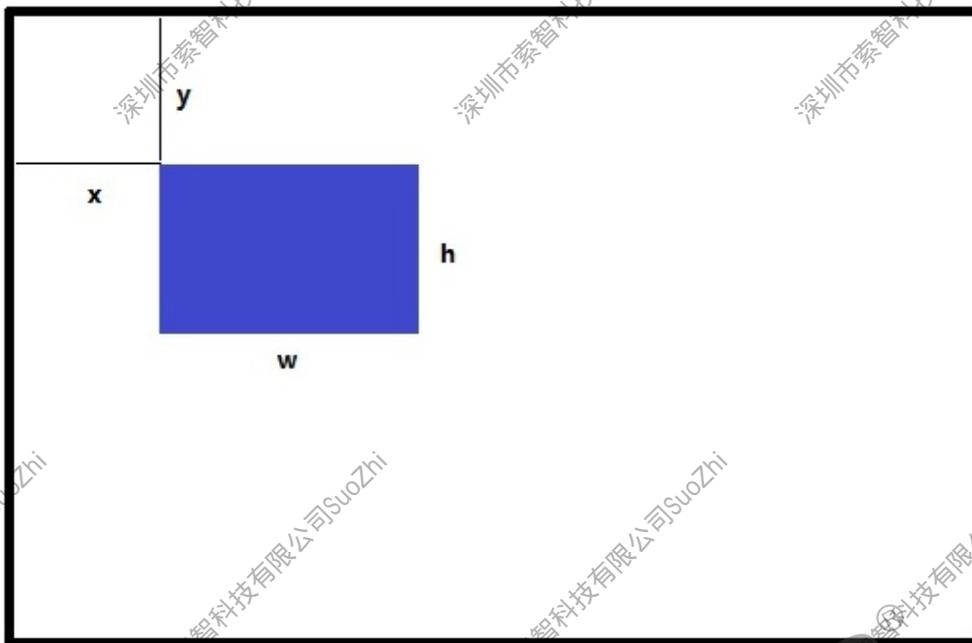


图 9-13: OSD 坐标和大小示意图

## 9.4 音频编解码常用功能检查方法

### 9.4.1 常用检测工具介绍

#### 1. 普通效果测试

将测试生成的 wav 文件拷贝到 PC 端使用 Windows Media Player 播放。

#### 2. 专业波形测试

专业的音频波形分析软件：Audacity。

比如测试回声消除效果时，可借助该工具分析。

使用说明：

检测采集通道的数据和回采通道的数据的延迟关系是否正确。

需要特殊的正弦波文件 tone\_16k\_accurate.wav, tone\_8k\_accurate.wav。

一般采集通道的数据应该稍微延迟一些。

**操作步骤：**

1. 为进行专业波形测试，需修改代码，audio\_hw.c中打开宏 `#define AI_HW_AEC_DEBUG_EN`。
2. 然后编译 `sample_aec`，这样底层模块在运行过程中会保存采集通道的数据和回采通道的数据。
3. 再用软件 Audacity 进行分析，从波形上看延迟时间是否正常。

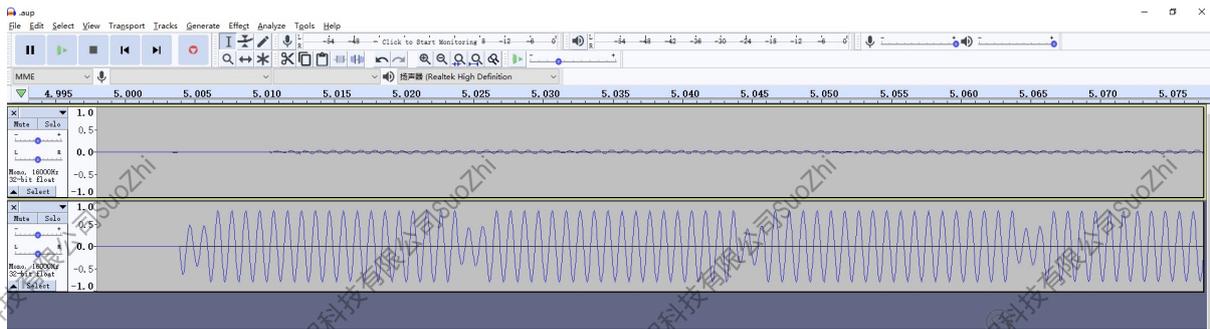


图 9-14: 通道波形对比图

## 著作权声明

版权所有 © 2022 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

## 商标声明

、 **全志科技** （不完全列举）均为珠海全志科技股份有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

## 免责声明

您购买的产品、服务或特性应受您与珠海全志科技股份有限公司（“全志”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，全志概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。全志尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，全志概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予全志的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。全志不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。全志不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。