



# AX VO API 文档

文档版本: V2.2

发布日期: 2024/9/10

AEXRA CONFIDENTIAL FOR SIPEED

# 目 录

前 言 .....	6
修订历史 .....	7
1 概述 .....	8
1.1 概述 .....	8
1.2 功能描述 .....	9
1.2.1 硬件功能框图 .....	9
1.2.2 重要概念 .....	10
1.2.3 通道个数配置 .....	11
1.2.4 画框线条宽度配置 .....	11
1.2.5 相关限制 .....	12
2 API 简介 .....	14
3 API 定义 .....	17
AX_VO_Init .....	17
AX_VO_Deinit .....	18
AX_VO_SetPubAttr .....	19
AX_VO_GetPubAttr .....	20
AX_VO_SetCSC .....	21
AX_VO_GetCSC .....	23
AX_VO_Enable .....	24
AX_VO_Disable .....	25
AX_VO_EnumMode .....	26
AX_VO_CreateVideoLayer .....	27

AX_VO_DestroyVideoLayer .....	28
AX_VO_SetVideoLayerAttr .....	29
AX_VO_GetVideoLayerAttr .....	30
AX_VO_SetVideoLayerCSC .....	31
AX_VO_GetVideoLayerCSC .....	33
AX_VO_BatchBegin .....	34
AX_VO_BatchEnd .....	36
AX_VO_EnableVideoLayer .....	37
AX_VO_DisableVideoLayer .....	38
AX_VO_BindVideoLayer .....	39
AX_VO_UnBindVideoLayer .....	40
AX_VO_BindGraphicLayer .....	41
AX_VO_UnBindGraphicLayer .....	42
AX_VO_GetLayerFd .....	43
AX_VO_GetLayerFrame .....	44
AX_VO_ReleaseLayerFrame .....	45
AX_VO_EnableChn .....	46
AX_VO_DisableChn .....	47
AX_VO_SetChnAttr .....	48
AX_VO_GetChnAttr .....	49
AX_VO_SendFrame .....	50
AX_VO_ShowChn .....	52
AX_VO_HideChn .....	53
AX_VO_QueryChnStatus .....	54
AX_VO_ClearChnBuf .....	55
AX_VO_GetChnFrame .....	57

AX_VO_ReleaseChnFrame .....	59
AX_VO_EnableWBC .....	60
AX_VO_DisableWBC .....	61
AX_VO_SetWBCAttr .....	62
AX_VO_GetWBCAttr .....	63
AX_VO_GetWBCFrame .....	64
AX_VO_ReleaseWBCFrame .....	66
AX_VO_GetWBCFd .....	67
AX_VO_CscMatrix .....	68
<b>4 数据结构 .....</b>	<b>69</b>
AX_VO_PUB_ATTR_T .....	69
AX_VO_CSC_T .....	74
AX_VO_DISPLAY_MODE_T .....	76
AX_VO_RECT_T .....	78
AX_VO_SIZE_T .....	79
AX_VO_VIDEO_LAYER_ATTR_T .....	80
AX_VO_CHN_ATTR_T .....	85
AX_VO_QUERY_STATUS_T .....	86
AX_VO_WBC_ATTR_T .....	87
AX_CSC_MATRIX_T .....	89
<b>5 应用实例 .....</b>	<b>90</b>
5.1 Layer 使用说明 .....	90
5.2 VO 显示使用说明 .....	92
5.3 PIP 相关的应用 .....	94
5.4 帧控使用说明 .....	94

6 错误码.....	96
7 调试信息 .....	97
7.1 Layer 相关的信息查询.....	97
7.2 Display 相关的信息查询 .....	100
7.3 用户态相关 LOG 的控制.....	101
7.4 固定时序相关说明.....	103

AEXRA CONFIDENTIAL FOR SIPEED

## 权利声明

爱芯元智半导体股份有限公司或其许可人保留一切权利。

非经权利人书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## 注意

您购买的产品、服务或特性等应受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的购买或使用范围之内。除非商业合同另有约定，本公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 前言



## 适用产品

AX620E 系列产品（AX630C、AX620Q）

## 适读人群

- 软件开发工程师
- 技术支持工程师

## 符号与格式定义

符号/格式	说明
<code>xxx</code>	表示您可以执行的命令行。
<i>斜体</i>	表示变量。如，“安装目录/AX620E_SDK_Vx.x.x/build 目录”中的“安装目录”是一个变量，由您的实际环境决定。
 说明/备注：	表示您在使用产品的过程中，我们向您说明的事项。
 注意：	表示您在使用产品的过程中，需要您特别注意的事项。

## 修订历史

文档版本	发布时间	修订说明
V1.0	2023/8/31	文档初版
V1.1	2023/10/08	增加 layer 输出 fifo 深度和 VB 个数限制关系说明
V1.2	2023/11/14	更新 proc 信息 更新 AX_VO_PUB_ATTR_T，新增在线离线模式
V1.3	2023/11/27	删除 AX_VO_SetChnFrameRate、X_VO_GetChnFrameRate 接口，增加帧控使用说明
V1.4	2023/12/5	修改接口时序定义，将 dts 时序参数配置移动到 SDK 中配置，实现 lvds、dpi、mipi 设备自适应
V1.5	2024/1/29	增加 AX_VO_CscMatrix 接口说明
V1.6	2024/1/31	增加 1.2.1 硬件功能框图 增加 1.2.3 限制 增加 5.3 PIP 相关应用 增加 7.3 固定时序相关说明 修改 API 注意事项说明
V1.7	2024/3/6	AX_VO_ReleaseLayerFrame/AX_VO_ReleaseChnFrame 增加注意事项说明
V1.8	2024/3/22	增加 enBlendMode 枚举参数项和描述说明 5.4 节帧控补充注意项
V1.9	2024/3/27	调试信息增加 display 相关的信息查询
V2.0	2024/4/11	修改 AX_VO_EnumMode 注意事项
V2.1	2024/5/11	增加 VO 最大通道个数和最大画框线宽配置说明
V2.2	2024/9/10	新增表 7-5 Display 图层相关信息描述



# 1 概述

## 1.1 概述

视频输出模块简称 VO（Video Output），该模块主动从内存相应位置读取视频和图形数据，并通过相应的显示设备输出视频和图形。

芯片支持的显示能力如下所示。

表1-1 芯片显示输出能力

设备编号	接口类型	最大输出能力	备注
VO0	MIPI DSI	1080p 60Hz	
	LVDS	1080p 60Hz	
	DPI	1080p 120Hz	SDR 能力到 60HZ; DDR 能力到 120HZ
	BT(656/601/1120)	1080p 120Hz	SDR 能力到 60HZ; DDR 能力到 120HZ
VO1	DPI	1080p 120Hz	SDR 能力到 60HZ; DDR 能力到 120HZ
	BT(656/601/1120)	1080p 120Hz	SDR 能力到 60HZ; DDR 能力到 120HZ

## 1.2 功能描述

### 1.2.1 硬件功能框图

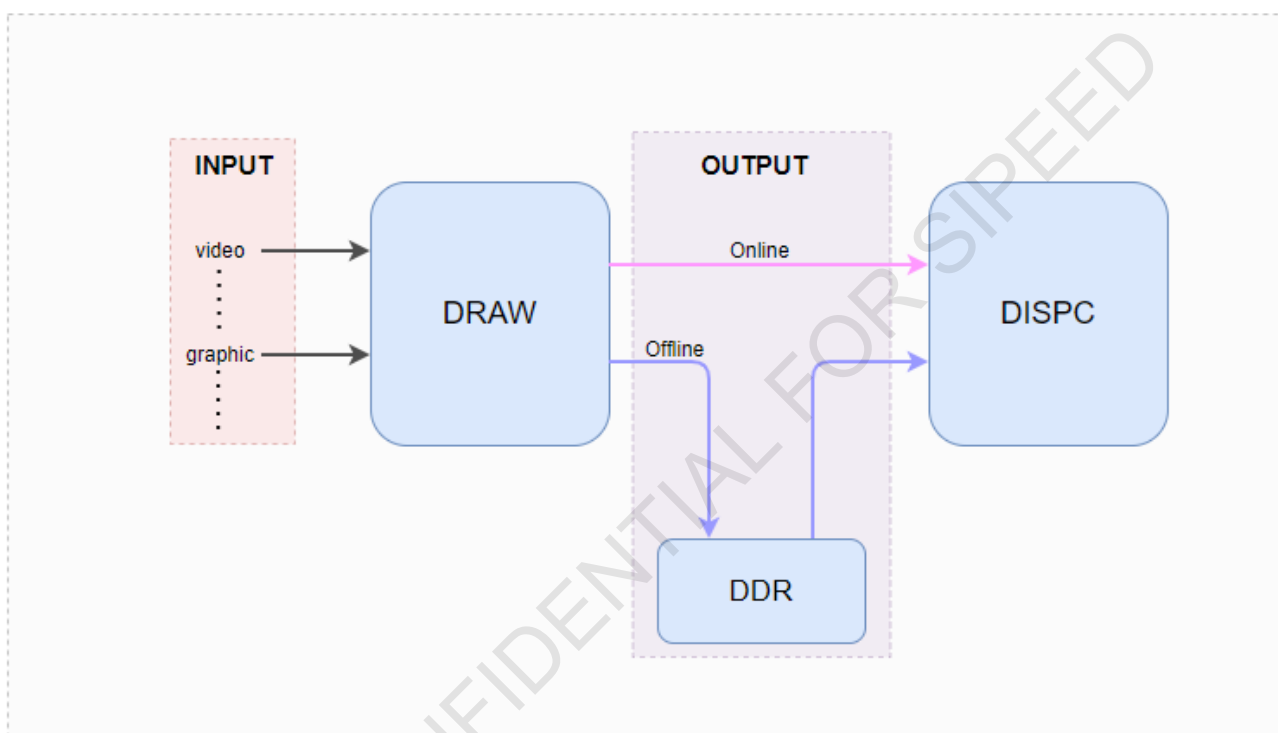


图1-1 VO 硬件功能框图

VO 硬件模块分为 DRAW 和 DISPC 二个子模块，DRAW 模块负责将输入的 VIDEO/GRAPHIC 图像合成后输出到 DDR 或 DISPC。DISPC 模块负责从 DDR 中读取图像或接收 DRAW 送过来的图像并将其转化成相应的硬件时序信号送到显示终端。

支持 2 种工作模式，即：Online 模式和 Offline 模式。如上图所示，Offline 模式 DRAW 的输出直接写到 DDR，DISPC 需要从 DDR 中读取图像。Online 模式 DRAW 的输出直接送到 DISPC。2 种模式各有优缺点，Online 模式优点是省带宽，但不够灵活。Offline 模式优点是灵活，二个子模块可以当成独立模块来使用，缺点则是占用更多的带宽。用户可根据自己的使用场景选择合适的场景。

## 1.2.2 重要概念

### 显示设备（VO DEV）

显示设备代表硬件显示控制器，它是对 DISPC 进行的软件抽象，用于将图像输出到显示终端上。

### 通道（Channel）

通道是对 VO 硬件 DRAW-INPUT 的软件抽象，SDK 将通道归属于图层管理，一个图层上可显示多个视频，每个视频显示区域称为一个通道，视频被限制在通道内，通道被限制在图层内。对于一个图层，其上面的通道都是独立的，不同的图层上的通道也是独立的。

### 图层（Layer）

图层是对 VO 硬件 DRAW-OUTPUT 的软件抽象，使用图层可以管理多个通道。本芯片上支持的 LAYER 数由 AX\_VO\_LAYER\_MAX 定义，图层与通道之间的关系如图所示。



图1-2 图层与通道之间的关系

### 1.2.3 通道个数配置

VO 支持通过 ko 加载和修改 dts 的方式修改最大支持通道个数，默认值为 16，可根据需要自行更改，配置更小的通道个数可以节省部分 cmm 和 os 内存。

#### ko 方式

insmod /soc/ko/ax\_vo.ko vo\_chn\_max = 12, 此命令可将最大通道个数配置为 12。不指定 vo\_chn\_max 参数时将尝试从 dts 中读取通道个数配置项。

#### dts 方式

读取 dtsi 文件中 VO 的 vo-chn-max 参数，如果该参数存在，使用配置值作为 VO 最大通道个数，如果不存在，则使用默认值 16。

```
vo0: vo@4407000 {
    .....
    vo-chn-max = <16>;
    vo-poly-line-width = <16>;
    operating-points-v2 = <&vo0_dfs>;
    status = "disabled";
};
```

### 1.2.4 画框线条宽度配置

VO 支持向 TDP 提供画框功能，通过 ko 加载和修改 dts 的方式修改矩形框最大线宽，默认值为 16，可根据需要自行更改，配置更小的线宽可节省部分 cmm 内存。

#### ko 方式

insmod /soc/ko/ax\_vo.ko vo\_poly\_line\_width = 8, 此命令可将矩形框最大支持线宽配置为 8。不指定 vo\_poly\_line\_width 参数时将尝试从 dts 中读取矩形框最大支持线宽配置项。

#### dts 方式

读取 dtsi 文件中 VO 的 vo-poly-line-width 参数，如果该参数存在，使用配置值作为 VO 矩形框最大线宽，如果不存在，则使用默认值 16。

```
vo0: vo@4407000 {
    .....
    vo-chn-max = <16>;
```

```

vo-poly-line-width = <16>;
operating-points-v2 = <&vo0_dfs>;
status = "disabled";
};

```

## 1.2.5 相关限制

本节主要描述 VO 使用的相关限制。

表1-2 通道(Chn)视频层和图形层格式限制

图像格式	FBDC	最小分辨率 (像素)	最大分辨率 (像素)	Stride 对齐限制 (字节)	
NV12	Y	2x2	4096x4096	FBDC 使能: 128 FBDC 关闭: 8	
ARGB1555	N	2x2	4096x4096	8	
ARGB4444	N	2x2	4096x4096	8	
ARGB8565	N	2x2	4096x4096	8	
ARGB8888	N	2x2	4096x4096	8	
RGBA5551	N	2x2	4096x4096	8	
RGBA4444	N	2x2	4096x4096	8	
RGBA5658	N	2x2	4096x4096	8	
RGBA8888	N	2x2	4096x4096	8	
RGB565	N	2x2	4096x4096	8	
BGR565	N	2x2	4096x4096	8	
RGB888	N	2x2	4096x4096	8	
BGR888	N	2x2	4096x4096	8	
NV16	N	2x2	4096x4096	8	

图像格式	FBDC	最小分辨率 (像素)	最大分辨率 (像素)	Stride 对齐限制 (字节)	
BITMAP	N	2x2	4096x4096	8	Chn 工作 V(视频层)时 不支持此格式

表1-3 图层(Layer)相关的限制

图像格式	FBC	最小分辨率 (像素)	最大分辨率 (像素)	Stride 对齐限制 (字节)	备注
NV12	Y	2x2	4096x4096	FBC 使能: 128 FBC 关闭: 8	
ARGB1555	N	2x2	4096x4096	8	Offline 不支持
ARGB4444	N	2x2	4096x4096	8	Offline 不支持
ARGB8565	N	2x2	4096x4096	8	Offline 不支持
ARGB8888	N	2x2	4096x4096	8	Offline 不支持
RGBA5551	N	2x2	4096x4096	8	Offline 不支持
RGBA4444	N	2x2	4096x4096	8	Offline 不支持
RGBA5658	N	2x2	4096x4096	8	Offline 不支持
RGBA8888	N	2x2	4096x4096	8	Offline 不支持
RGB565	N	2x2	4096x4096	8	Offline 不支持
BGR565	N	2x2	4096x4096	8	Offline 不支持
RGB888	N	2x2	4096x4096	8	Offline 不支持
BGR888	N	2x2	4096x4096	8	Offline 不支持
NV16	N	2x2	4096x4096	8	Offline 不支持

## 2 API 简介

VO 模块提供的 API 接口如下：

- [AX\\_VO\\_Init](#)：初始化 VO 相关的软硬件资源。
- [AX\\_VO\\_Deinit](#)：释放 VO 相关的软硬件资源。
- [AX\\_VO\\_SetPubAttr](#)：设置显示设备的公共属性。
- [AX\\_VO\\_GetPubAttr](#)：获取显示设备已设置的公共属性。
- [AX\\_VO\\_SetCSC](#)：设置显示设备 CSC。
- [AX\\_VO\\_GetCSC](#)：获取显示设备 CSC。
- [AX\\_VO\\_Enable](#)：使能指定的显示设备。
- [AX\\_VO\\_Disable](#)：关闭指定的显示设备。
- [AX\\_VO\\_EnumMode](#)：枚举指定显示设备所支持的显示模式。
- [AX\\_VO\\_CreateVideoLayer](#)：创建一个 Video Layer。
- [AX\\_VO\\_DestroyVideoLayer](#)：销毁一个 Video Layer。
- [AX\\_VO\\_SetVideoLayerAttr](#)：设置 VideoLayer 的属性。
- [AX\\_VO\\_GetVideoLayerAttr](#)：获取 VideoLayer 已设置所属性。
- [AX\\_VO\\_SetVideoLayerCSC](#)：设置指定 Video layer 的 CSC。
- [AX\\_VO\\_GetVideoLayerCSC](#)：获取指定 Video layer 的 CSC。
- [AX\\_VO\\_BatchBegin](#)：设置指定 Video layer 上的通道的设置属性开始。
- [AX\\_VO\\_BatchEnd](#)：设置指定 Video layer 上的通道的设置属性结束。
- [AX\\_VO\\_EnableVideoLayer](#)：使能相应的 VideoLayer。
- [AX\\_VO\\_DisableVideoLayer](#)：关闭相应的 VideoLayer。

- [AX\\_VO\\_BindVideoLayer](#): 绑定 VideoLayer 到指定的显示设备。
- [AX\\_VO\\_UnBindVideoLayer](#): VideoLayer 与显示设备的解绑。
- [AX\\_VO\\_BindGraphicLayer](#): 绑定 GraphicLayer 到指定的显示设备。
- [AX\\_VO\\_UnBindGraphicLayer](#): GraphicLayer 与显示设备的解绑。
- [AX\\_VO\\_GetLayerFd](#): 获取指定 Layer 对应的文件描述符。
- [AX\\_VO\\_GetLayerFrame](#): 获取 VO 输出的 LayerFrame。
- [AX\\_VO\\_ReleaseLayerFrame](#): 释放已获取的 LayerFrame。
- [AX\\_VO\\_EnableChn](#): 使能相应 Layer 上的通道。
- [AX\\_VO\\_DisableChn](#): 关闭相应 Layer 上的通道。
- [AX\\_VO\\_SetChnAttr](#): 设置通道的属性。
- [AX\\_VO\\_GetChnAttr](#): 获取通道上已设置的属性。
- [AX\\_VO\\_SendFrame](#): 发送 Frame 到指定的通道。
- [AX\\_VO\\_ShowChn](#): 显示指定通道。
- [AX\\_VO\\_HideChn](#): 隐藏指定通道。
- [AX\\_VO\\_QueryChnStatus](#): 查询视频输出通道状态。
- [AX\\_VO\\_ClearChnBuf](#): 清空指定输出通道的缓存数据。
- [AX\\_VO\\_GetChnFrame](#): 获取指定通道帧数据。
- [AX\\_VO\\_ReleaseChnFrame](#): 释放获取的帧数据。
- [AX\\_VO\\_EnableWBC](#): 使能回写设备。
- [AX\\_VO\\_DisableWBC](#): 禁用回写设备。
- [AX\\_VO\\_SetWBCAttr](#): 设置回写设备属性。
- [AX\\_VO\\_GetWBCAttr](#): 获取回写设备属性。
- [AX\\_VO\\_GetWBCFrame](#): 获取回写设备的输出图像数据。



- [AX\\_VO\\_ReleaseWBCFrame](#): 释放回写设备的输出图像数据。
- [AX\\_VO\\_GetWbcFd](#): 获取回写设备对应的文件描述符。
- [AX\\_VO\\_CscMatrix](#): 通过自定义转换参数实现图像调整

AEXRA CONFIDENTIAL FOR SIPEED

## 3 API 定义

### AX\_VO\_Init

#### 【描述】

初始化 VO 模块的软硬件资源。

#### 【语法】

```
AX_S32 AX_VO_Init(AX_VOID)
```

#### 【参数】

无

#### 【返回值】

返回值	描述
非 0	失败
0	成功

#### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

#### 【注意】

- 如果硬件初始化失败或未预期的错误，返回 AX\_ERR\_VO\_DEV\_OP\_FAIL。

## AX\_VO\_Deinit

### 【描述】

释放 VO 模块的软硬件资源。

### 【语法】

```
AX_S32 AX_VO_Deinit(AX_VOID)
```

### 【参数】

无

### 【返回值】

返回值	描述
非 0	失败
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 调用此接口前需确保所有的显示设备均已关闭。

## AX\_VO\_SetPubAttr

### 【描述】

设置指定显示设备的公共属性。

### 【语法】

```
AX_S32 AX_VO_SetPubAttr(VO_DEV VoDev, const AX\_VO\_PUB\_ATTR\_T *pstPubAttr)
```

### 【参数】

参数名称	描述	输入/输出
VoDev	显示设备号。 取值范围：[0, AX_VO_DEV_MAX)	输入
pstPubAttr	显示设备属性结构体指针，结构体实参使用前需先清 0。	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 调用此接口前确保 AX\_VO\_Init 已成功调用且对应的设备没有使能，即：AX\_VO\_Enable 没有被调用。

## AX\_VO\_GetPubAttr

### 【描述】

获取指定显示设备的属性。

### 【语法】

```
AX_S32 AX_VO_GetPubAttr(VO_DEV VoDev, AX\_VO\_PUB\_ATTR\_T *pstPubAttr)
```

### 【参数】

参数名称	描述	输入/输出
VoDev	显示设备号。 取值范围：[0, AX_VO_DEV_MAX)	输入
pstPubAttr	显示设备属性结构体指针	输出

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

暂无

## AX\_VO\_SetCSC

### 【描述】

设置显示设备 CSC。

### 【语法】

```
AX_S32 AX_VO_SetCSC(VO_DEV VoDev, const AX\_VO\_CSC\_T *pstVideoCSC)
```

### 【参数】

参数名称	描述	输入/输出
VoDev	显示设备号。 取值范围：[0, AX_VO_DEV_MAX)	输入
pstVideoCSC	设备图像输出效果结构体指针	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 此接口支持 CSC 的设置以及亮度、色度、饱和度、对比度的设置，如设置的范围超出最大值则以支持的最大值配置到设备
- 此接口必须在成功调用 AX\_VO\_SetPubAttr 之后设置才有效
- 显示设备默认没有开启亮度、色度、饱和度、对比度功能。如开启，则推荐亮度 50，色度 50，饱和度 25，对比度 25

- 显示设备 CSC 默认配置为 AX\_VO\_CSC\_MATRIX\_BT601\_TO\_BT601
- 显示设备 CSC 存在如下限制：
  1. 如工作在 Online 模式下并且输出为 YUV 格式，CSC 不生效。输出为 RGB 格式时，YUV 转 RGB 转换矩阵才生效。
  2. 如工作在 Offline 模式下，通常都是将 LAYER 绑定到显示设备，这时如果输入为 RGB 或者输出为 RGB 格式时，转换类型参数才会生效。

AEXRA CONFIDENTIAL FOR SIPEED

## AX\_VO\_GetCSC

### 【描述】

获取显示设备 CSC。

### 【语法】

```
AX_S32 AX_VO_GetCSC(VO_DEV VoDev, AX\_VO\_CSC\_T *pstVideoCSC)
```

### 【参数】

参数名称	描述	输入/输出
VoDev	显示设备号。 取值范围：[0, AX_VO_DEV_MAX)	输入
pstVideoCSC	设备图像输出效果结构体指针	输出

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

暂无



## AX\_VO\_Enable

### 【描述】

使能指定的显示设备。

### 【语法】

```
AX_S32 AX_VO_Enable (VO_DEV VoDev)
```

### 【参数】

参数名称	描述	输入/输出
VoDev	显示设备号。 取值范围：[0, AX_VO_DEV_MAX)	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 调用此接口前确保 AX\_VO\_SetPubAttr 已成功调用。

## AX\_VO\_Disable

### 【描述】

关闭指定的显示设备。

### 【语法】

```
AX_S32 AX_VO_Disable (VO_DEV VoDev)
```

### 【参数】

参数名称	描述	输入/输出
VoDev	显示设备号。 取值范围：[0, AX_VO_DEV_MAX)	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

暂无

## AX\_VO\_EnumMode

### 【描述】

枚举指定显示设备所支持的显示模式。

### 【语法】

```
AX_S32 AX_VO_EnumMode (VO_DEV VoDev, AX\_VO\_DISPLAY\_MODE\_T *pstMode)
```

### 【参数】

参数名称	描述	输入/输出
VoDev	显示设备号。 取值范围：[0, AX_VO_DEV_MAX)	输入
pstMode	显示模式	输出

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 该接口输出的是所指设备所支持的所有显示模式，可能包含有 LVDS、DPI-BT、DSI。
- 该接口不支持查询分辨率能力，本芯片支持的分辨率可参考 AX\_VO\_INTF\_SYNC\_E 枚举定义，除此之外还支持用户自定义模式。

## AX\_VO\_CreateVideoLayer

### 【描述】

分配创建一个未使用的 LAYER。

### 【语法】

```
AX_S32 AX_VO_CreateVideoLayer(VO_LAYER *u32VoLayer)
```

### 【参数】

参数名称	描述	输入/输出
u32VoLayer	存储成功分配创建 LAYER 的句柄	输出

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 在使用 LAYER 提供的相关功能之前必须先调用此接口分配创建一个 LAYER，成功调用后获得一个与 LAYER 对应的句柄。

## AX\_VO\_DestroyVideoLayer

### 【描述】

销毁一个不使用的 LAYER。

### 【语法】

```
AX_S32 AX_VO_DestroyVideoLayer(VO_LAYER u32VoLayer)
```

### 【参数】

参数名称	描述	输入/输出
u32VoLayer	LAYER 句柄	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 即将要销毁 LAYER 的句柄必须是已成功调用 AX\_VO\_CreateVideoLayer 之后获取到的并且所对应的 LAYER 不能处在 Enable 状态。

## AX\_VO\_SetVideoLayerAttr

### 【描述】

设置指定 VideoLayer 的属性。

### 【语法】

```
AX_S32 AX_VO_SetVideoLayerAttr(VO_LAYER VoLayer, const  
AX\_VO\_VIDEO\_LAYER\_ATTR T *pstLayerAttr)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
pstLayerAttr	Layer 属性结构体指针	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 传入的属性结构体未使用成员须赋值成 0。
- 调用此接口前必须确保所对应的 Layer 处于初始状态，即刚通过 AX\_VO\_CreateVideoLayer 之后的 Layer。
- 部分属性只支持 VO 工作在 Offline 模式，具体请参考属性结构体介绍。

## AX\_VO\_GetVideoLayerAttr

### 【描述】

获取 Layer 的属性。

### 【语法】

```
AX_S32 AX_VO_GetVideoLayerAttr(VO_LAYER VoLayer, AX\_VO\_VIDEO\_LAYER\_ATTR\_T
*pstLayerAttr)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
pstLayerAttr	Layer 属性结构体指针	输出

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- Layer 句柄必须有效。

## AX\_VO\_SetVideoLayerCSC

### 【描述】

设置视频层 CSC。

### 【语法】

```
AX_S32 AX_VO_SetVideoLayerCSC (VO_LAYER VoLayer, const AX\_VO\_CSC\_T
*pstVideoCSC)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
pstVideoCSC	图像输出效果结构体指针	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- Layer 句柄必须有效。
- 此接口只支持 CSC 的设置，不支持亮度、色度、饱和度、对比度的设置
- Layer 默认配置为 AX\_VO\_CSC\_MATRIX\_BT601\_TO\_BT601
- 如果此 Layer 需要绑定到显示，推荐使用 AX\_VO\_CSC\_MATRIX\_BT601\_TO\_BT601 或 AX\_VO\_CSC\_MATRIX\_BT709\_TO\_BT709



- 如果 Layer 上有 Chn 是 RGB 格式，则 AX\_VO\_CSC\_MATRIX\_BT601\_TO\_XXX 类型的 CSC 有意义。如果 Layer 的输出为 RGB 格式，则 AX\_VO\_CSC\_MATRIX\_RGB\_TO\_XXX 类型的有意义。

AEXRA CONFIDENTIAL FOR SIPEED

## AX\_VO\_GetVideoLayerCSC

### 【描述】

获取视频层 CSC。

### 【语法】

```
AX_S32 AX_VO_GetVideoLayerCSC (VO_LAYER VoLayer, AX\_VO\_CSC\_T *pstVideoCSC)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
pstVideoCSC	图像输出效果结构体指针	输出

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- Layer 句柄必须有效。

## AX\_VO\_BatchBegin

### 【描述】

设置视频层上的通道的设置属性开始。

### 【语法】

```
AX_S32 AX_VO_BatchBegin(VO_LAYER VoLayer)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- Layer 句柄必须有效。
- BEGIN 和 END 接口要配对使用，否则 BEGIN 后设置的通道属性不会生效。
- 此接口可对通道的动态操作进行批处理，例如设置通道属性、通道显示、通道隐藏、使能通道、禁止通道等。当 VO 切换画面大小或显示隐藏时，需要重新准备内存并保证切换的及时性，因此建议的调用顺序为：在 BEGIN 和 END 接口之间调用设置通道属性、通道显示、通道隐藏等接口，对设备的多个通道进行批处理操作。
- 批处理中的通道操作都是在 AX\_VO\_BatchEnd 接口调用之后生效，建议合理控制 BEGIN 和 END 接口的范围。例如希望批处理禁止通道，并禁止视频层。由于禁止视频层的调用

必须在禁止通道成功后生效，禁止通道必须在 `AX_VO_BatchEnd` 接口调用后生效，因此要按照如下顺序调用 `AX_VO_BatchBegin`、`AX_VO_DisableChn`、`AX_VO_BatchEnd` 和 `AX_VO_DisableVideoLayer`。如果是按照以下顺序调用 `AX_VO_BatchBegin`、`AX_VO_DisableChn`、`AX_VO_DisableVideoLayer` 和 `AX_VO_BatchEnd`，将导致禁止视频层失败。

- 重复调用该接口会把之前的 `cmd` 清除掉再重新开始记录 `cmd`。
- 批处理操作中，对通道相关的操作不会立即进行，如果该操作不合法也不会立刻返回错误，直到调用了 `AX_VO_BatchEnd` 之后才会返回错误。

AEXRA CONFIDENTIAL FOR SIPRED

## AX\_VO\_BatchEnd

### 【描述】

设置视频层上的通道的设置属性结束。

### 【语法】

```
AX_S32 AX_VO_BatchEnd(VO_LAYER VoLayer)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- Layer 句柄必须有效。
- 调用前需保证 AX\_VO\_BatchBegin 已经调用。
- BEGIN 和 END 接口要配对使用，否则 BEGIN 后设置的通道属性不会生效。
- 在调用 END 接口时，必须要保证此接口及相应的批处理操作的正确性。

## AX\_VO\_EnableVideoLayer

### 【描述】

使能指定的 VideoLayer。

### 【语法】

```
AX_S32 AX_VO_EnableVideoLayer(VO_LAYER VoLayer)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- Layer 句柄必须有效。
- 调用此接口前必须确保所对应的 VideoLayer 相关的属性已设置，即：  
AX\_VO\_SetVideoLayerAttr 已成功调用。

## AX\_VO\_DisableVideoLayer

### 【描述】

关闭指定的 VideoLayer。

### 【语法】

```
AX_S32 AX_VO_DisableVideoLayer(VO_LAYER VoLayer)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- Layer 句柄必须有效。
- Layer 关闭前必须确保其上所有通道都已关闭，否则无法成功关闭。

## AX\_VO\_BindVideoLayer

### 【描述】

绑定指定的 VideoLayer 到指定的显示设备。

### 【语法】

```
AX_S32 AX_VO_BindVideoLayer(VO_LAYER VoLayer, VO_DEV VoDev)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
VoDev	显示设备号 取值范围: [0, AX_VO_DEV_MAX)	输入

### 【返回值】

返回值	描述
非 0	失败, 其值为错误码
0	成功

### 【需求】

- 头文件: ax\_vo\_api.h
- 库文件: libax\_vo.so

### 【注意】

- Layer 句柄必须有效。
- VoDev 必须在所支持的范围内。
- 支持重复绑定, 支持同一个 VideoLayer 绑定到多个 VoDev 上。



## AX\_VO\_UnBindVideoLayer

### 【描述】

解绑指定的 VideoLayer。

### 【语法】

```
AX_S32 AX_VO_UnBindVideoLayer(VO_LAYER VoLayer, VO_DEV VoDev)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
VoDev	显示设备号 取值范围: [0, AX_VO_DEV_MAX)	输入

### 【返回值】

返回值	描述
非 0	失败, 其值为错误码
0	成功

### 【需求】

- 头文件: ax\_vo\_api.h
- 库文件: libax\_vo.so

### 【注意】

- Layer 句柄必须有效。
- VoDev 必须在所支持的范围内。

## AX\_VO\_BindGraphicLayer

### 【描述】

绑定指定的 GraphicLayer 到指定的显示设备，GraphicLayer 与 FB 对应。如编号为 0 的 GraphicLayer 对应 FB0。

### 【语法】

```
AX_S32 AX_VO_BindGraphicLayer (GRAPHIC_LAYER GraphicLayer, VO_DEV VoDev)
```

### 【参数】

参数名称	描述	输入/输出
GraphicLayer	GraphicLayer 编号	输入
VoDev	显示设备号 取值范围：[0, AX_VO_DEV_MAX)	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- GraphicLayer 必须在所支持的范围内，0 对应 fb0、1 对应 fb1。
- 目前同一个显示设备最多只支持 1 个 GUI 层，1 个鼠标层。
- 在绑定之前需先将 fb 的分辨率、图像格式等配置好，不允许绑定之后动态改变。
- 支持重复绑定，不支持同一个 GraphicLayer 绑定到多个 VoDev 上。

## AX\_VO\_UnBindGraphicLayer

### 【描述】

解绑指定的 GraphicLayer。

### 【语法】

```
AX_S32 AX_VO_UnBindGraphicLayer(GRAPHIC_LAYER GraphicLayer, VO_DEV VoDev)
```

### 【参数】

参数名称	描述	输入/输出
GraphicLayer	GraphicLayer 编号	输入
VoDev	显示设备号 取值范围: [0, AX_VO_DEV_MAX)	输入

### 【返回值】

返回值	描述
非 0	失败, 其值为错误码
0	成功

### 【需求】

- 头文件: ax\_vo\_api.h
- 库文件: libax\_vo.so

### 【注意】

- GraphicLayer 必须在所支持的范围内。
- 需与 AX\_VO\_BindGraphicLayer 配对使用。

## AX\_VO\_GetLayerFd

### 【描述】

获取指定 Layer 对应的文件描述符，此文件描述符主要用于提供给 select 接口去监听。

### 【语法】

```
AX_S32 AX_VO_GetLayerFd(VO_LAYER VoLayer, AX_S32 *s32Fd)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
s32Fd	保存输出的文件描述符	输出

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- Layer 句柄必须有效。
- 获取到的 fd 不使用后不可直接调用 close 相关的操作。

## AX\_VO\_GetLayerFrame

### 【描述】

获取指定 VideoLayer 上输出的帧。

### 【语法】

```
AX_S32 AX_VO_GetLayerFrame (VO_LAYER VoLayer, AX_VIDEO_FRAME_T *pstVoFrame,  
AX_S32 s32MS)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
pstVoFrame	VideoLayer 输出帧结构体指针	输出
s32MS	超时选项，取值范围：  -1：阻塞  0：非阻塞  正值：超时时间，没有上限值，以 ms 为单位	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- Layer 句柄必须有效。

## AX\_VO\_ReleaseLayerFrame

### 【描述】

释放指定 VideoLayer 上的帧。

### 【语法】

```
AX_S32 AX_VO_ReleaseLayerFrame(VO_LAYER VoLayer, const AX_VIDEO_FRAME_T
*pstVoFrame)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
pstVoFrame	待释放的 VideoLayer 帧结构体指针	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 此接口与 AX\_VO\_GetLayerFrame 配对使用。
- 实际上，此接口的 VoLayer 参数无实际用途，可在取值范围内任意设置。
- 不允许重复释放。

## AX\_VO\_EnableChn

### 【描述】

使能对应 VideoLayer 上指定的通道。

### 【语法】

```
AX_S32 AX_VO_EnableChn(VO_LAYER VoLayer, VO_CHN VoChn)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
VoChn	通道编号	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- VoLayer 所对应的 Layer 必须已使能。
- VoChn 必须在所对应 Layer 所支持的通道范围内，Layer 所支持的通道数可通过相关属性来指定。
- 调用此接口前必须要确保此通道相关的属性已设置。

## AX\_VO\_DisableChn

### 【描述】

关闭对应 VideoLayer 上指定的通道。

### 【语法】

```
AX_S32 AX_VO_DisableChn(VO_LAYER VoLayer, VO_CHN VoChn)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
VoChn	通道编号	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- VoLayer 所对应的 Layer 必须已使能。
- VoChn 必须在所对应 Layer 所支持的通道范围内，Layer 所支持的通道数可通过相关属性来指定。



## AX\_VO\_SetChnAttr

### 【描述】

设置指定通道的相关属性。

### 【语法】

```
AX_S32 AX_VO_SetChnAttr(VO_LAYER VoLayer, VO_CHN VoChn, const  
AX\_VO\_CHN\_ATTR\_T *pstChnAttr)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
VoChn	通道编号	输入
pstChnAttr	通道属性结构体指针	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- VoLayer 所对应的 Layer 必须已使能。
- VoChn 必须在所对应 Layer 所支持的通道范围内，Layer 所支持的通道数可通过相关属性来指定。
- 此接口支持动态调用，如在通道已使能的情况调用此接口需确保属性中的 u32Width、u32Height 与使能前保持一致。

## AX\_VO\_GetChnAttr

### 【描述】

获取指定通道的相关属性。

### 【语法】

```
AX_S32 AX_VO_GetChnAttr(VO_LAYER VoLayer, VO_CHN VoChn, AX\_VO\_CHN\_ATTR\_T
*pstChnAttr)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
VoChn	通道编号	输入
pstChnAttr	属性结构体指针	输出

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- VoLayer 所对应的 Layer 必须已使能。
- VoChn 必须在所对应 Layer 所支持的通道范围内，Layer 所支持的通道数可通过相关属性来指定。

## AX\_VO\_SendFrame

### 【描述】

发送 Frame 到指定的通道。

### 【语法】

```
AX_S32 AX_VO_SendFrame(VO_LAYER VoLayer, VO_CHN VoChn, AX_VIDEO_FRAME_T  
*pstVoFrame, AX_S32 s32MS)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
VoChn	通道编号	输入
pstVoFrame	图像帧结构体指针，结构体实参使用前建议先清 0。	输入
s32MS	超时选项，取值范围： -1：阻塞 0：非阻塞 正值：超时时间，没有上限值，以 ms 为单位	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- VoLayer 所对应的 Layer 必须已使能。
- VoChn 必须在所对应 Layer 所支持的通道范围内且对应的通道必须已使能，Layer 所支持的通道数可通过相关属性来指定。

AEXRA CONFIDENTIAL FOR SIPEED

## AX\_VO\_ShowChn

### 【描述】

显示指定通道。

### 【语法】

```
AX_S32 AX_VO_ShowChn (VO_LAYER VoLayer, VO_CHN VoChn)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
VoChn	通道编号	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 调用前需保证视频输出通道所在的视频层已经使能。
- 默认情况下通道处于显示状态。

## AX\_VO\_HideChn

### 【描述】

隐藏指定通道。

### 【语法】

```
AX_S32 AX_VO_HideChn(VO_LAYER VoLayer, VO_CHN VoChn)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
VoChn	通道编号	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 调用前需保证视频输出通道所在的视频层已经使能。

## AX\_VO\_QueryChnStatus

### 【描述】

查询视频输出通道状态，获取通道当前占用的待显示的图像数目。

### 【语法】

```
AX_S32 AX_VO_QueryChnStatus (VO_LAYER VoLayer, VO_CHN VoChn,  
AX\_VO\_QUERY\_STATUS\_T *pstChnStatus)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
VoChn	通道编号	输入
pstChnStatus	通道状态结构体指针。	输出

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- VoLayer 对应的 Layer 必须已使能。
- VoChn 必须在所对应 Layer 所支持的通道范围内，Layer 所支持的通道数可通过相关属性来指定。

## AX\_VO\_ClearChnBuf

### 【描述】

清空指定输出通道的缓存数据。

### 【语法】

```
AX_S32 AX_VO_ClearChnBuf(VO_LAYER VoLayer, VO_CHN VoChn, AX_BOOL bClrAll)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
VoChn	通道编号	输入
bClrAll	是否将通道中的数据全部清空。取值范围： AX_TRUE: 将通道中的数据全部清空，屏幕上该通道的区域将显示背景色，直到有新的图像数据到来。 AX_FALSE: 在通道 buffer 中保留一幅图像数据，而将其他数据清空。	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- VoLayer 所对应的 Layer 必须已使能。



- VoChn 必须在所对应 Layer 所支持的通道范围内且对应的通道必须已使能。

AEXRA CONFIDENTIAL FOR SIPEED

## AX\_VO\_GetChnFrame

### 【描述】

获取通道帧。

### 【语法】

```
AX_S32 AX_VO_GetChnFrame(VO_LAYER VoLayer, VO_CHN VoChn, AX_VIDEO_FRAME_T  
*pstVoFrame, AX_S32 s32MS)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
VoChn	通道编号	输入
pstVoFrame	VideoLayer 输出帧结构体指针	输出
s32MS	超时选项，取值范围： -1：阻塞 0：非阻塞 正值：超时时间，没有上限值，以 ms 为单位	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

**【注意】**

- 调用该接口前必须保证通道已经使能且通道属性 `bInUseFrOutput` 成员必须为 `AX_TRUE`。
- 该接口支持阻塞式获取通道帧，`s32MilliSec` 为-1 时表示阻塞获取，一直等到通道有图像为止；`s32MilliSec` 为 0 时表示非阻塞获取；`s32MilliSec > 0` 时，表示阻塞等待时间，超过该时间则不再等待。
- 获取后应保证及时的释放，调用 `AX_VO_ReleaseChnFrame` 可释放通道帧。

AEXRA CONFIDENTIAL FOR SIPEED

## AX\_VO\_ReleaseChnFrame

### 【描述】

释放输出通道图像数据。

### 【语法】

```
AX_S32 AX_VO_ReleaseChnFrame(VO_LAYER VoLayer, VO_CHN VoChn, const  
AX_VIDEO_FRAME_T *pstVoFrame)
```

### 【参数】

参数名称	描述	输入/输出
VoLayer	Layer 句柄	输入
VoChn	通道编号	输入
pstVoFrame	待释放的 VideoLayer 帧结构体指针	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 此接口与 AX\_VO\_GeChnFrame 配对使用。
- 实际上，此接口的 VoLayer 和 VoChn 参数无实际用途，可在取值范围内任意设置。
- 不允许重复调用本接口。

## AX\_VO\_EnableWBC

### 【描述】

使能回写设备。

### 【语法】

```
AX_S32 AX_VO_EnableWBC (VO_WBC VoWbc)
```

### 【参数】

参数名称	描述	输入/输出
VoWbc	回写设备号。 取值范围：[0, AX_VO_DEV_MAX)	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 回写功能在使能前必须先设置其相关属性。
- 不允许回写设备使能后再去设置其属性。
- 回写设备必须工作在 Offline 模式。

## AX\_VO\_DisableWBC

### 【描述】

禁用回写设备。

### 【语法】

AX\_S32 AX\_VO\_DisableWBC (VO\_WBC VoWbc)

### 【参数】

参数名称	描述	输入/输出
VoWbc	回写设备号。 取值范围：[0, AX_VO_DEV_MAX)	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 回写设备处在关闭状态的情况下调用此接口不返回错误。
- 回写设备必须工作在 Offline 模式。

## AX\_VO\_SetWBCAttr

### 【描述】

设置回写设备属性。

### 【语法】

```
AX_S32 AX_VO_SetWBCAttr(VO_WBC VoWbc, const AX\_VO\_WBC\_ATTR\_T *pstWbcAttr)
```

### 【参数】

参数名称	描述	输入/输出
VoWbc	回写设备号。 取值范围：[0, AX_VO_DEV_MAX)	输入
pstWbcAttr	回写设备属性结构体指针。	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 回写功能在使能前必须先设置其相关属性。
- 不允许回写设备使能后再去设置其属性。
- 属性中的帧率功能暂不支持。
- 属性中的回写模式只支持 NORMAL。
- 回写设备必须工作在 Offline 模式。

## AX\_VO\_GetWBCAttr

### 【描述】

获取回写设备属性。

### 【语法】

```
AX_S32 AX_VO_GetWBCAttr(VO_WBC VoWbc, AX\_VO\_WBC\_ATTR\_T *pstWbcAttr)
```

### 【参数】

参数名称	描述	输入/输出
VoWbc	回写设备号。 取值范围：[0, AX_VO_DEV_MAX)	输入
pstWbcAttr	回写设备属性结构体指针。	输出

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

暂无



## AX\_VO\_GetWBCFrame

### 【描述】

获取回写设备的输出图像数据。

### 【语法】

```
AX_S32 AX_VO_GetWBCFrame (VO_WBC VoWbc, AX_VIDEO_FRAME_T *pstVFrame, AX_S32  
s32MilliSec)
```

### 【参数】

参数名称	描述	输入/输出
VoWbc	回写设备号。 取值范围：[0, AX_VO_DEV_MAX)	输入
pstVFrame	获取的回写输出图像数据信息结构体指针。	输出
S32MilliSec	超时时间。单位：ms。 超时参数 s32MilliSec 设为-1 时，为阻塞接口，为 0 时为非阻塞接口，大于 0 时为超时等待时间	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 获取操作应保证与释放操作配对。
- 不允许重复释放

- 调用前确保回写已使能。
- 获取后应保证及时的释放。
- 回写设备必须工作在 Offline 模式。

AEXRA CONFIDENTIAL FOR SIPEED

## AX\_VO\_ReleaseWBCFrame

### 【描述】

释放回写设备的输出图像数据。

### 【语法】

```
AX_S32 AX_VO_ReleaseWBCFrame(VO_WBC VoWbc, const AX_VIDEO_FRAME_T *pstVFrame)
```

### 【参数】

参数名称	描述	输入/输出
VoWbc	回写设备号。 取值范围：[0, AX_VO_DEV_MAX)	输入
PstVFrame	释放的输出通道图像数据信息结构体指针。	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 获取操作应保证与释放操作配对。
- 不允许重复释放。

## AX\_VO\_GetWBCFd

### 【描述】

获取回写设备对应的文件描述符。

### 【语法】

```
AX_S32 AX_VO_GetWbcFd(VO_WBC VoWbc, AX_S32 *s32Fd)
```

### 【参数】

参数名称	描述	输入/输出
VoWbc	回写设备号。 取值范围：[0, AX_VO_DEV_MAX)	输入
s32Fd	获取的回写设备文件描述符指针。	输出

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 获取到的 fd 不使用后不可直接调用 close 相关的操作。

## AX\_VO\_CscMatrix

### 【描述】

用户可通过自定义转换参数实现图像调整。

### 【语法】

```
AX_S32 AX_VO_CscMatrix(const AX_VIDEO_FRAME_T *ptSrc, AX_VIDEO_FRAME_T  
*ptDst, const AX\_CSC\_MATRIX\_T *ptCscMatrix)
```

### 【参数】

参数名称	描述	输入/输出
ptSrc	待转换的源图像帧结构体指针。	输入
ptDst	转换后的目的图像帧结构体指针。	输入/输出
ptCscMatrix	用户自定义的转换参数	输入

### 【返回值】

返回值	描述
非 0	失败，其值为错误码
0	成功

### 【需求】

- 头文件：ax\_vo\_api.h
- 库文件：libax\_vo.so

### 【注意】

- 该接口为独立接口，不需要依赖任何 vo 的 api。
- 接口仅支持 yuv420 同格式转换。
- 该接口仅支持转换矩阵类型为用户自定义模式，ntype 为 AX\_CSC\_MATRIX\_USER。
- 该接口中的内存由调用者申请释放。

## 4 数据结构

### AX\_VO\_PUB\_ATTR\_T

#### 【说明】

显示设备相关属性结构体定义。

#### 【定义】

```
typedef struct axVO_PUB_ATTR_S {  
    AX_VO_MODE_E enMode;  
    AX_VO_INTF_TYPE_E enIntfType;  
    AX_VO_OUT_FMT_E enIntfFmt;  
    AX_VO_INTF_SYNC_E enIntfSync;  
    AX_VO_SYNC_INFO_T stSyncInfo;  
} AX_VO_PUB_ATTR_T;
```

#### 【成员】

成员名称	描述
enMode	VO 工作模式配置，原型定义： typedef enum { AX_VO_MODE_OFFLINE, AX_VO_MODE_ONLINE, AX_VO_MODE_BUTT } AX_VO_MODE_E; 只有 VO0 能支持 ONLINE 模式，VO1 不支持
enIntfType	接口类型典型配置，原型定义： typedef enum {

成员名称	描述
	<pre>AX_VO_INTF_DPI, AX_VO_INTF_BT601, AX_VO_INTF_BT656, AX_VO_INTF_BT1120, AX_VO_INTF_DSI, AX_VO_INTF_LVDS, AX_VO_INTF_BUTT } AX_VO_INTF_TYPE_E;</pre> <p>需要注意只有 DPU1</p>
enIntfFmt	<p>接口格式典型配置，此配置主要只对 BT/DPI 输出有效，原型定义：</p> <pre>typedef enum {     AX_VO_OUT_FMT_UNUSED = 0,     AX_VO_OUT_FMT_RGB565,     AX_VO_OUT_FMT_RGB666,     AX_VO_OUT_FMT_RGB666LP,     AX_VO_OUT_FMT_RGB888,     AX_VO_OUT_FMT_YUV422,     AX_VO_OUT_FMT_BUTT } AX_VO_OUT_FMT_E;</pre>
enIntfSync	<p>接口时序典型配置，原型定义：</p> <pre>typedef enum {     AX_VO_OUTPUT_576P50,     AX_VO_OUTPUT_576I50     AX_VO_OUTPUT_480P60,     AX_VO_OUTPUT_480I60,</pre>

成员名称	描述
	<p>AX_VO_OUTPUT_720P25, AX_VO_OUTPUT_720P30, AX_VO_OUTPUT_720P50, AX_VO_OUTPUT_720P60, AX_VO_OUTPUT_1080P24, AX_VO_OUTPUT_1080P25, AX_VO_OUTPUT_1080P30, AX_VO_OUTPUT_1080P50, AX_VO_OUTPUT_1080P60, AX_VO_OUTPUT_640x480_60, AX_VO_OUTPUT_800x480_60, AX_VO_OUTPUT_800x600_60, AX_VO_OUTPUT_1024x600_60, AX_VO_OUTPUT_1024x768_60, AX_VO_OUTPUT_1280x1024_60, AX_VO_OUTPUT_1366x768_60, AX_VO_OUTPUT_1280x800_60, AX_VO_OUTPUT_1440x900_60, AX_VO_OUTPUT_1600x1200_60, AX_VO_OUTPUT_1680x1050_60, AX_VO_OUTPUT_1920x1200_60, AX_VO_OUTPUT_2560x1600_60, AX_VO_OUTPUT_3840x2160_24, AX_VO_OUTPUT_3840x2160_25, AX_VO_OUTPUT_3840x2160_30,</p>



成员名称	描述
	<pre> AX_VO_OUTPUT_3840x2160_50, AX_VO_OUTPUT_3840x2160_60, AX_VO_OUTPUT_4096x2160_24, AX_VO_OUTPUT_4096x2160_25, AX_VO_OUTPUT_4096x2160_30, AX_VO_OUTPUT_4096x2160_50, AX_VO_OUTPUT_4096x2160_60, AX_VO_OUTPUT_720x1280_60, AX_VO_OUTPUT_1080x1920_60, AX_VO_OUTPUT_1080x1920_30, AX_VO_OUTPUT_USER, AX_VO_OUTPUT_BUTT } AX_VO_INTF_SYNC_E; </pre>
stSyncInfo	<p>接口时序结构体，原型定义：</p> <pre> typedef struct axVO_SYNC_INFO_T {     AX_BOOL bSynm;     AX_BOOL bIop;     AX_U16 u16Vact;     AX_U16 u16Vbb;     AX_U16 u16Vfb;     AX_U16 u16Hact;     AX_U16 u16Hbb;     AX_U16 u16Hfb;     AX_U16 u16Hmid;     AX_U16 u16Bvact; </pre>

成员名称	描述
	<pre>AX_U16 u16Bvbb; AX_U16 u16Bvfb; AX_U16 u16Hpw; AX_U16 u16Vpw; AX_U32 u32Pclk; AX_BOOL bIdv; AX_BOOL bIhs; AX_BOOL bIvs; } AX_VO_SYNC_INFO_T;</pre> <p>在使能用户时序时，该结构体生效，用户须自行保证时序配置的正确性。</p>

#### 【相关数据类型及接口】

- [AX\\_VO\\_SetPubAttr](#)
- [AX\\_VO\\_GetPubAttr](#)

## AX\_VO\_CSC\_T

### 【说明】

定义图像输出效果结构体。

### 【定义】

```
typedef struct {  
    AX_VO_CSC_MATRIX_E enCscMatrix;  
    AX_U32 u32Luma;          /* luminance: 0 ~ 100 */  
    AX_U32 u32Contrast;      /* contrast : 0 ~ 100 */  
    AX_U32 u32Hue;          /* hue : 0 ~ 100 */  
    AX_U32 u32Satuature;     /* satuation: 0 ~ 100 */  
} AX_VO_CSC_T;
```

### 【成员】

成员名称	描述
enCscMatrix	CSC 矩阵类型，原型定义： typedef enum { AX_VO_CSC_MATRIX_IDENTITY = 0, AX_VO_CSC_MATRIX_BT601_TO_BT601, AX_VO_CSC_MATRIX_BT601_TO_BT709, AX_VO_CSC_MATRIX_BT709_TO_BT709, AX_VO_CSC_MATRIX_BT709_TO_BT601, AX_VO_CSC_MATRIX_BT601_TO_RGB_PC, AX_VO_CSC_MATRIX_BT709_TO_RGB_PC, AX_VO_CSC_MATRIX_RGB_TO_BT601_PC, AX_VO_CSC_MATRIX_RGB_TO_BT709_PC, AX_VO_CSC_MATRIX_RGB_TO_BT2020_PC, AX_VO_CSC_MATRIX_BT2020_TO_RGB_PC, 

成员名称	描述
	<code>AX_VO_CSC_MATRIX_RGB_TO_BT601_TV,</code> <code>AX_VO_CSC_MATRIX_RGB_TO_BT709_TV,</code> <code>AX_VO_CSC_MATRIX_BUTT</code> <code>} AX_VO_CSC_MATRIX_E;</code>
u32Luma	亮度值 范围: [0, 100]
u32Contrast	对比度值 范围: [0, 100]
u32Hue	色调值 范围: [0, 100]
u32Satuature	饱和度值 范围: [0, 100]

**【相关数据类型及接口】**

- [AX\\_VO\\_SetVideoLayerCSC](#)
- [AX\\_VO\\_GetVideoLayerCSC](#)

## AX\_VO\_DISPLAY\_MODE\_T

### 【说明】

定义显示模式结构体。

### 【定义】

```
typedef struct axVO_DISPLAY_MODE_T {  
    AX_U16 u16ModesNum;  
    struct {  
        AX_U16 u16Width;  
        AX_U16 u16Height;  
        AX_U16 u16Refresh;  
        AX_U16 u16Type;  
        AX_U32 u32Flags;  
    } stModes[AX_VO_MODE_MAX];  
} AX_VO_DISPLAY_MODE_T;
```

### 【成员】

成员名称	描述
u16ModesNum	有效模式的数量
stModes	有效的模式，成员含义：  u16Width 表示分辨率宽 u16Height 表示分辨率高 u16Refresh 表示刷新率 u16Type 表示该输出模式支持的接口，可取值为：  VO_DISPLAY_TYPE_LVDS、 VO_DISPLAY_TYPE_VIRTUAL、 VO_DISPLAY_TYPE_DSI  u32Flags 表示该输出模式下的一些标志，如水平、垂直同步的极

成员名称	描述
	性，隔行、逐行等，可使用正面的宏进行判断：  VO_DISPLAY_MODE_FLAG_PHSYNC、 VO_DISPLAY_MODE_FLAG_NHSYNC、 VO_DISPLAY_MODE_FLAG_PVSYNC、 VO_DISPLAY_MODE_FLAG_NVSYNC、 VO_DISPLAY_MODE_FLAG_INTERLACE

**【相关数据类型及接口】**

- [AX\\_VO\\_EnumMode](#)

## AX\_VO\_RECT\_T

### 【说明】

描述 VO 矩形区域相关的结构体定义。

### 【定义】

```
typedef struct axVO_RECT_T {  
    AX_U32 u32X;  
    AX_U32 u32Y;  
    AX_U32 u32Width;  
    AX_U32 u32Height;  
} AX_VO_RECT_T;
```

### 【成员】

成员名称	描述
u32X	矩形区域的起始横坐标
u32Y	矩形区域的起始纵坐标
u32Width	矩形区域的宽
u32Height	矩形区域的高

### 【相关数据类型及接口】

- [AX\\_VO\\_SetVideoLayerAttr](#)
- [AX\\_VO\\_GetVideoLayerAttr](#)

## AX\_VO\_SIZE\_T

### 【说明】

定义 VO 图像 SIZE 相关的结构体。

### 【定义】

```
typedef struct axVO_SIZE_T {  
    AX_U32 u32Width;  
    AX_U32 u32Height;  
} AX_VO_SIZE_T;
```

### 【成员】

成员名称	描述
u32Width	图像的宽
u32Height	图像的高

### 【相关数据类型及接口】

- [AX\\_VO\\_SetVideoLayerAttr](#)
- [AX\\_VO\\_GetVideoLayerAttr](#)



## AX\_VO\_VIDEO\_LAYER\_ATTR\_T

### 【说明】

定义 VideoLayer 相关的属性。

### 【定义】

```
typedef struct axVO_VIDEO_LAYER_ATTR_T {  
    AX_VO_SIZE_T stImageSize;  
    AX_FRAME_COMPRESS_INFO_T stCompressInfo;  
    AX_IMG_FORMAT_E enPixFmt;  
    AX_VO_LAYER_SYNC_MODE_E enSyncMode;  
    AX_U32 u32PrimaryChnId;  
    AX_U32 u32FifoDepth;  
    AX_U32 u32BkClr;  
    AX_U32 u32DispatchMode;  
    AX_VO_LAYER_WB_MODE_E enWBMode;  
    AX_U32 u32InplaceChnId;  
    AX_POOL u32PoolId;  
    AX_VO_PART_MODE_E enPartMode;  
    AX_VO_BLEND_MODE_E enBlendMode;  
    AX_VO_ENGINE_MODE_E enEngineMode;  
    AX_U32 u32EngineId;  
    AX_U32 u32Toleration;  
    AX_F32 f32FrmRate;  
} AX_VO_VIDEO_LAYER_ATTR_T;
```

### 【成员】

成员名称	描述
stImageSize	图像分辨率结构体，即合成画面尺寸
stCompressInfo	视频层输出是否使能压缩

成员名称	描述
enPixFmt	视频层输出的像素格式：NV12/NV21
enSyncMode	<p>视频层的同步模式，原型如下：</p> <pre>typedef enum {     AX_VO_LAYER_SYNC_NORMAL,     AX_VO_LAYER_SYNC_SHUTTLE,     AX_VO_LAYER_SYNC_GROUPING,     AX_VO_LAYER_SYNC_PRIMARY,     AX_VO_LAYER_SYNC_BUTT, } AX_VO_LAYER_SYNC_MODE_E;</pre> <p>目前只支持 NORMAL、PRIMARY 模式，NORMAL 模式表示所有通道均有图像帧时才会去触发硬件进行图像绘制，PRIMARY 模式表示只要 Primary Chn 中有新的图像帧送来时就会去触发硬件进行图像绘制</p>
u32PrimaryChnId	当 enSyncMode 为 AX_VO_LAYER_SYNC_PRIMARY 时，此成员用于指定对应的通道 ID
u32FifoDepth	指定视频层输出 FIFO 的深度，推荐值 3，该值应小于 layer 输出 VB 个数，否则会在帧写满 FIFO 后出现获取不到 VB 的情况。
u32BkClr	<p>指定视频层的背景色，格式为 RGB888。</p> <p>背景色支持动态设置。</p>
u32DispatchMode	<p>指定视频层输出的分发模式，可能组合：</p> <p>FIFO/LINK/FIFO+LINK</p> <p>此属性只在 Layer 工作在 OFFLINE 模式下的 VO 设备时才有效。</p>
enWBMode	<p>指定 LAYER 输出 buffer 的来源，原型如下：</p> <pre>typedef enum {     AX_VO_LAYER_WB_POOL,     AX_VO_LAYER_WB_INPLACE,</pre>

成员名称	描述
	<pre>AX_VO_LAYER_WB_BUF_BUTT,</pre> <pre>} AX_VO_LAYER_WB_MODE_E;</pre> <ol style="list-style-type: none"> <li>1. 设置成 AX_VO_LAYER_WB_POOL 时，则输出的 buffer 来源于 u32PoolId 指定的 pool，此时 u32PoolId 的值必须有效</li> <li>2. 设置成 AX_VO_LAYER_WB_INPLACE 时，则输出的 buffer 来源于 u32InplaceChnId 指定通道上的图像帧，此时 LAYER 的背景色不生效。</li> </ol> <p>此属性只在 VO 工作在 Offline 模式下有效。</p>
u32InplaceChnId	当 enWBMode 为 AX_VO_LAYER_WB_INPLACE 时，此成员用于指定对应的通道 ID
u32PoolId	当 enWBMode 为 AX_VO_LAYER_WB_POOL 时，此成员用于指定对应的 Pool ID
enPartMode	<p>指定视频层的分隔处理模式，原型如下：</p> <pre>typedef enum axVO_PART_MODE_E {</pre> <pre>    AX_VO_PART_MODE_SINGLE = 0,</pre> <pre>    AX_VO_PART_MODE_MULTI = 1,</pre> <pre>    AX_VO_PART_MODE_BUTT</pre> <pre>} AX_VO_PART_MODE_E;</pre> <ol style="list-style-type: none"> <li>1. 当设置为 AX_VO_PART_MODE_SINGLE，允许通道可重叠，重叠次序由各通道的优先级指定</li> </ol> <p>本芯片不支持 AX_VO_PART_MODE_MULTI 模式</p>
enBlendMode	<p>指定混合模式，原型如下：</p> <pre>typedef enum axVO_BLEND_MODE_E {</pre> <pre>    AX_VO_BLEND_MODE_DEFAULT = 0,</pre> <pre>    AX_VO_BLEND_MODE_INDEPENDENT = 1,</pre>

成员名称	描述
	<pre>AX_VO_BLEND_MODE_GLOBAL = 2, AX_VO_BLEND_MODE_BUTT } AX_VO_BLEND_MODE_E;</pre> <p>描述视频层和图形层叠加时 alpha 的选择方式，目前只支持 AX_VO_BLEND_MODE_DEFAULT：当图形层为 ARGB 格式时，使用像素自带的 alpha。当图形层为 RGB 格式时，使用图形层 fb 设置的 alpha 参数。 AX_VO_BLEND_MODE_GLOBAL 变量未合入 1.7.0_P2.X 分支。</p>
enEngineMode	<p>指定 Layer 使用的引擎，原型如下：</p> <pre>typedef enum axVO_ENGINE_MODE_E {     AX_VO_ENGINE_MODE_AUTO = 0,     AX_VO_ENGINE_MODE_FORCE,     AX_VO_ENGINE_MODE_BUTT } AX_VO_ENGINE_MODE_E;</pre> <p>选择 AX_VO_ENGINE_MODE_AUTO 时，如果 Layer 已绑定到显示设备，则会选择绑定的显示设备硬件作为引擎，如果 Layer 未绑定显示设备或绑定的显示设备是 LITE 类型，则会默认用 DPU0 的硬件做 DRAW 处理。</p> <p>选择 AX_VO_ENGINE_MODE_FORCE 时，需要指定有效的 u32EngineId。</p> <p>需要注意，当显示设备工作在 Online 模式时，最多只允许 1 个 Layer 选择该显示设备作为其引擎。</p>
u32EngineId	<p>当 enEngineMode 指定为 AX_VO_ENGINE_MODE_FORCE 时，u32EngineId 用于指定具体的引擎，目前本芯片上支持的引擎号分别为： 0 - VO0，1 - VO1</p>

成员名称	描述
u32Toleration	设置播放容忍度，以毫秒为单位，默认值为 10000
f32FrmRate	<ol style="list-style-type: none"><li>1. 当 enLayerSync 为 AX_VO_LAYER_SYNC_SHUTTLE 时，此成员用于指定视频层的帧率</li><li>2. 当 enLayerSync 为 AX_VO_LAYER_SYNC_NORMAL 并且该 layer 需要绑定到显示，则此成员需设成与显示刷新率相同</li></ol> 其它情况下，该成员无意义

#### 【相关数据类型及接口】

- [AX\\_VO\\_SetVideoLayerAttr](#)
- [AX\\_VO\\_GetVideoLayerAttr](#)

## AX\_VO\_CHN\_ATTR\_T

### 【说明】

定义视频层上通道相关的属性。

### 【定义】

```
typedef struct axVO_CHN_ATTR_T {  
    AX_VO_RECT_S stRect;  
    AX_U32 u32FifoDepth;  
    AX_U32 u32Priority;  
    AX_BOOL bKeepPrevFr;  
    AX_BOOL bInUseFrOutput;  
} AX_VO_CHN_ATTR_T;
```

### 【成员】

成员名称	描述
stRect	指定通道在视频层上的区域范围，只有起始坐标支持动态属性
u32FifoDepth	指定通道上 FIFO 的深度，范围[0-8]，等于 0 时使用默认值 3。 不支持动态设置
u32Priority	指定通道的优先级，不支持动态设置
bKeepPrevFr;	指定通道是否开启保持前一帧的功能，不支持动态设置
bInUseFrOutput	指定通道是否支持输出正在使用的帧，不支持动态设置

### 【相关数据类型及接口】

- [AX\\_VO\\_SetChnAttr](#)
- [AX\\_VO\\_GetChnAttr](#)

## AX\_VO\_QUERY\_STATUS\_T

### 【说明】

定义视频输出通道状态。

### 【定义】

```
typedef struct axVO_QUERY_STATUS_T {  
    AX_U32 u32ChnBufUsed;  
} AX_VO_QUERY_STATUS_T;
```

### 【成员】

成员名称	描述
u32ChnBufUsed	视频输出通道当前占用的 buf 数目

### 【相关数据类型及接口】

➤ [AX\\_VO\\_QueryChnStatus](#)

## AX\_VO\_WBC\_ATTR\_T

### 【说明】

定义回写设备的属性。

### 【定义】

```
typedef struct axVO_WBC_ATTR_T {  
    AX_VO_WBC_SOURCE_TYPE_E enSourceType;  
    AX_VO_WBC_MODE_E enMode;  
    AX_U32 u32FifoDepth;  
    AX_F32 f32FrameRate;  
} AX_VO_WBC_ATTR_T;
```

### 【成员】

成员名称	描述
enSourceType	回写源选择，原型定义如下：  typedef enum axVO_WBC_SOURCE_TYPE_E { AX_VO_WBC_SOURCE_DEV = 0x0, AX_VO_WBC_SOURCE_VIDEO = 0x1, AX_VO_WBC_SOURCE_BUTT } AX_VO_WBC_SOURCE_TYPE_E;
enMode	回写模式配置，原型定义如下：  typedef enum axVO_WBC_MODE_E { AX_VO_WBC_MODE_NORMAL = 0, AX_VO_WBC_MODE_DROP_REPEAT, AX_VO_WBC_MODE_BUTT, } AX_VO_WBC_MODE_E;  目前只支持 NORMAL



成员名称	描述
u32FifoDepth	设置回写 FIFO 的深度
f32FrameRate	设置回写的帧率，暂不支持

**【相关数据类型及接口】**

- [AX\\_VO\\_SetWBCAttr](#)
- [AX\\_VO\\_GetWBCAttr](#)

AEXRA CONFIDENTIAL FOR SIPEED

## AX\_CSC\_MATRIX\_T

见《AX 公共数据结构文档.docx》中的结构体定义说明。

AEXRA CONFIDENTIAL FOR SIPEED

## 5 应用实例

### 5.1 Layer 使用说明

Layer/Chn 相关的接口与显示接口之间是独立的，Layer/Chn 相关的状态转化关系如下图所示：

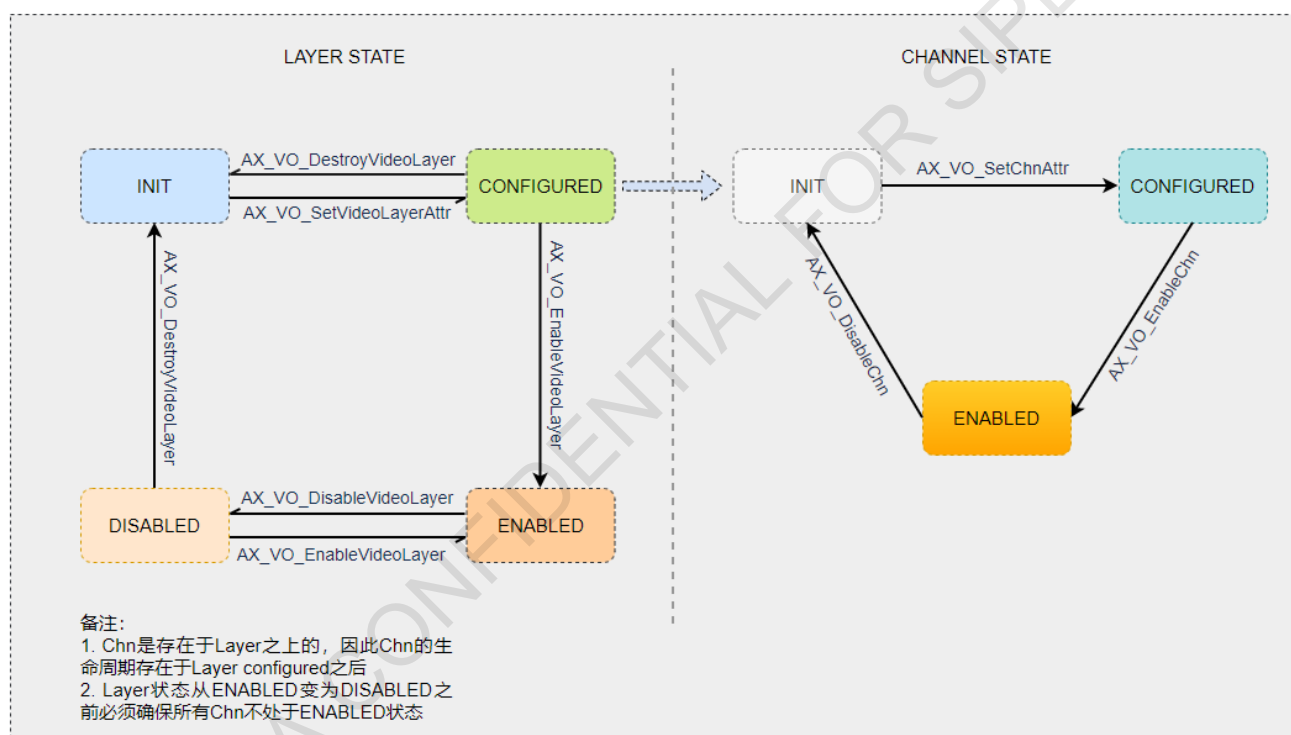


图5-1 Layer/Chn 状态转化关系

#### Layer 的相关状态：

**INIT：** 初始状态，成功调用 `AX_VO_CreateVideoLayer` 之后，此 LAYER 则处于 INIT 状态，当 LAYER 处于 DISABLED 状态时，成功调用 `AX_VO_DestroyVideoLayer` 后 LAYER 所处的状态也是 INIT 状态。

**CONFIGURED：** 已配置状态，成功调用 `AX_VO_SetVideoLayerAttr` 之后所处的状态。

**ENABLED：** 已使能状态，成功调用 `AX_VO_EnableVideoLayer` 之后所处的状态。

**DISABLED：** 已关闭状态，成功调用 `AX_VO_DisableVideoLayer` 之后所处的状态，从使能状

态到关闭状态必须确保 LAYER 上所有 Chn 都处在非使能状态。

### CHN 的相关状态：

INIT：初始状态，成功调用 AX\_VO\_SetVideoLayerAttr 之后，此 LAYER 上所有的 CHN 均处于 INIT 状态。

CONFIGURED：已配置状态，成功调用 AX\_VO\_SetChnAttr 之后所处的状态。

ENABLED：已使能状态，成功调用 AX\_VO\_EnableChn 之后所处的状态。

Layer/Chn 相关的使用流程可参考下图：

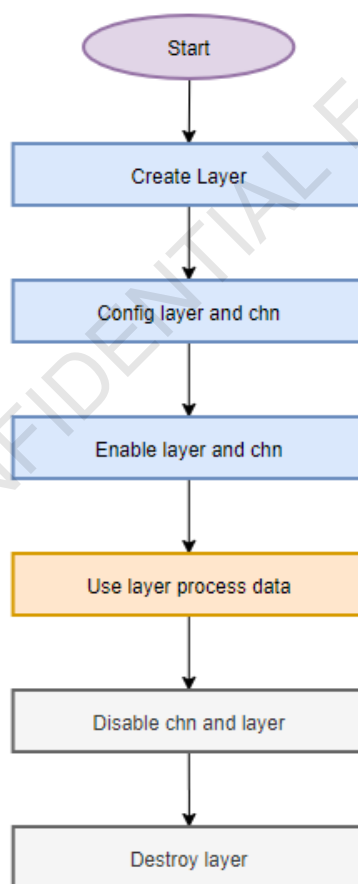


图5-2 Layer 的使用流程

具体的使用方法可参考 SDK 包，sample\_vo 示例程序。

## 5.2 VO 显示使用说明

显示设备的状态转化关系如下图所示：

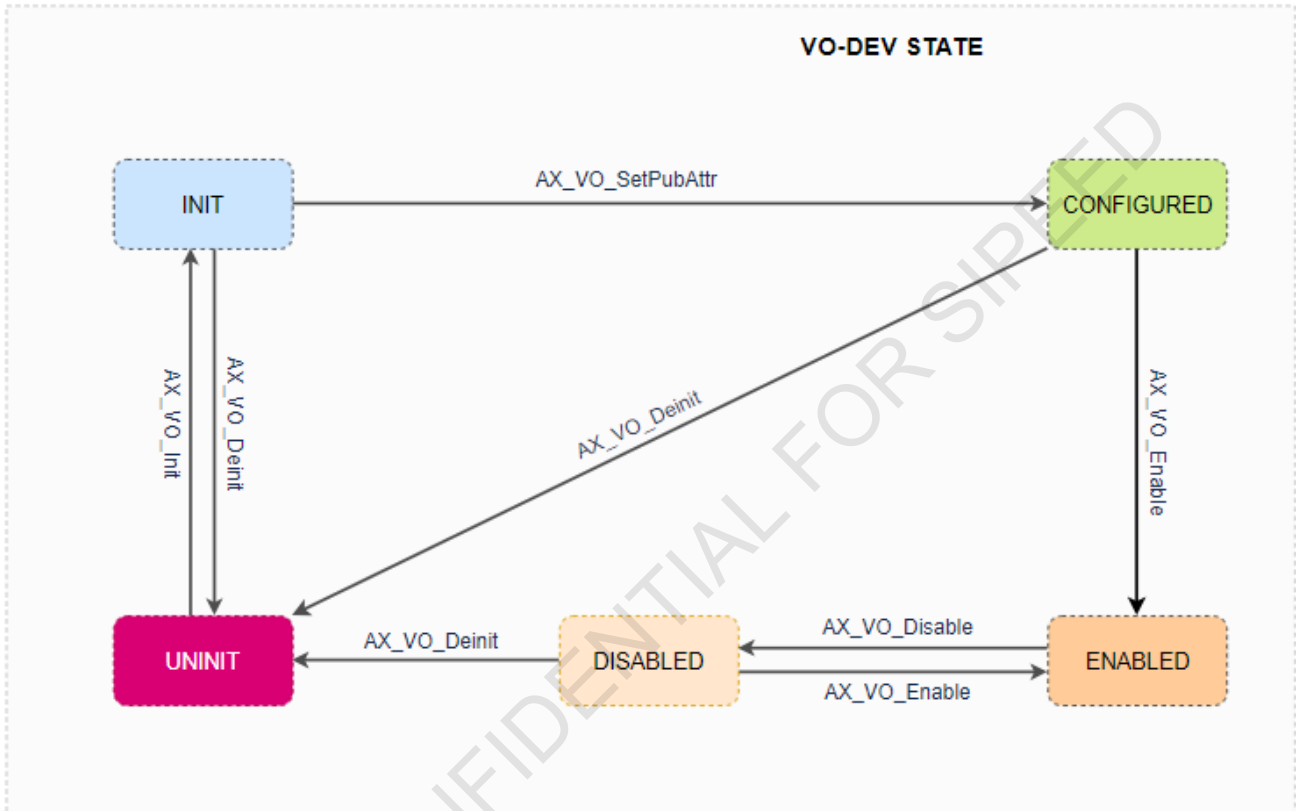


图5-3 显示设备状态转化关系

使用 VO 显示的基本流程如下图所示：

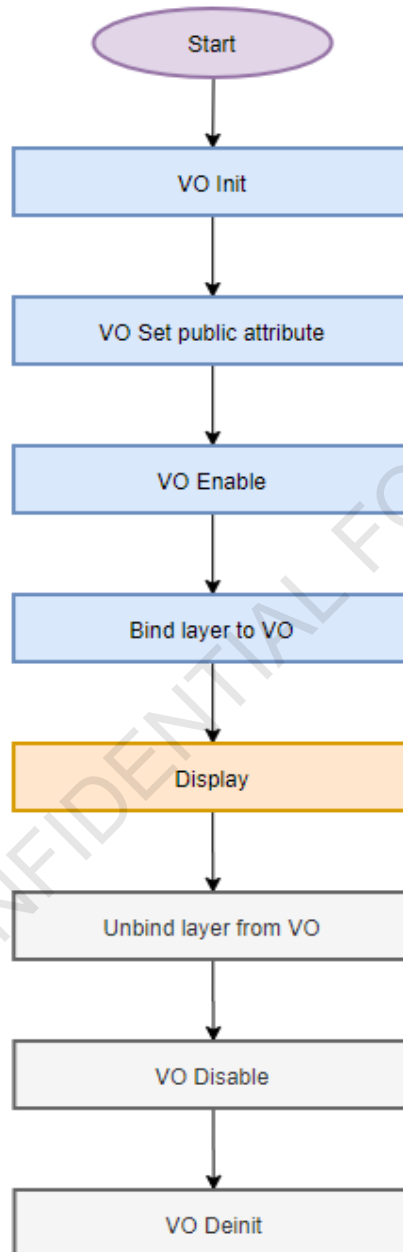


图5-4 VO 显示的使用流程

具体的使用方法可参考 SDK 包，sample\_vo 示例程序。

## 5.3 PIP 相关的应用

基于 VO 的硬件机制实现 PIP 功能，下面举例 PIP 场景来说明使用方法。

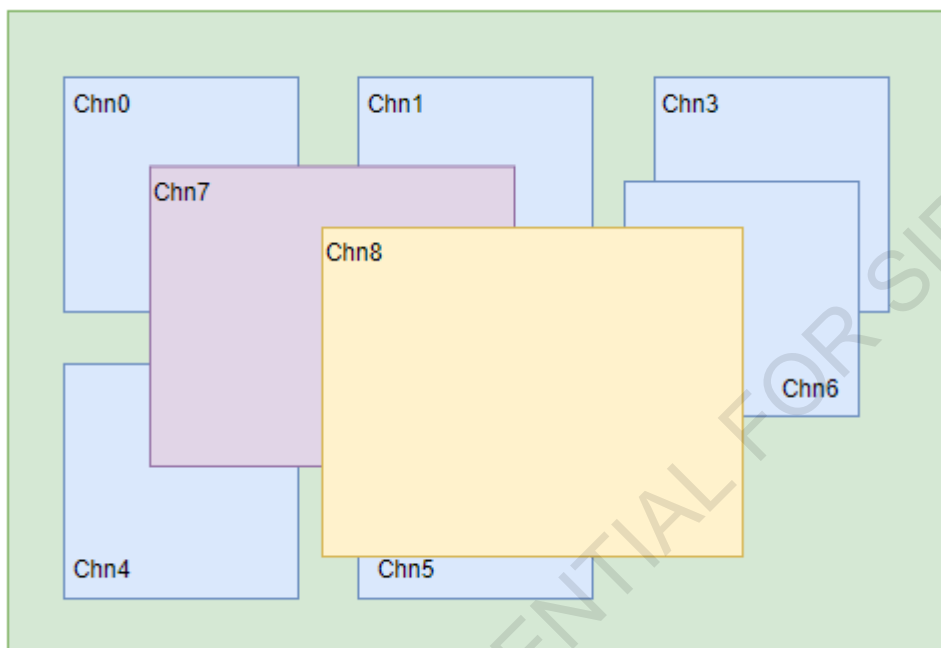


图5-5 通道之间任意重叠场景

其中 Chn7/Chn8 均为 PIP 通道，Chn0~6 为非 PIP 通道，因为本芯片不支持 AX\_VO\_PART\_MODE\_MULTI，所以 PIP 业务只能 Layer 工作在 Single 模式下来实现，大概方法如下：

1. Layer 属性相关的配置，enPartMode 属性设置成 AX\_VO\_PART\_MODE\_SINGLE，其他属性则根据实际使用情况来配置。
2. Chn 属性相关的配置，所有通道 Chn0~8 的 u32Priority 依据叠加顺序来设置，上层的数值大。

这种工作方式的优点是支持任意重叠。缺点则是性能差，占用带宽大。

## 5.4 帧控使用说明

VO 对于已绑定显示设备的 Layer，其通道可以支持帧率控制，具体方法：

1. 当检测到通道接收的帧处于回放模式时，帧控使能，帧率会依据前后帧的 PTS 动态控制。
2. 当检测到通道接收帧的 `u32FrameFlag` 中设置了 `AX_FRM_FLG_FR_CTRL` 时，帧控使能，帧率会依据前后帧的 PTS 动态控制。

帧控范围：前后帧 PTS 的间隔所表征的帧率小于显示设备的刷新率时，通道以 PTS 间隔所表征的帧率来输出帧，当其大于显示设备的刷新率时，通道会以显示设备刷新率来输出帧。

**！ 注意：**

- 帧控使能时，要求上游送帧尽可能的快。
- 帧控使能时，通道的深度尽可能大，以免通道深度过小引起阻塞。
- 如果送来的帧的 PTS 为 0，则 VO 会自动打上合适的 PTS，如果用户希望使用自己设置的 PTS，则设置的 PTS 不能出现 0 和非 0 混合的情况。
- 如果送来的帧的 PTS 为 -1，则 VO 会直接丢掉不显示。
- 使用帧控功能，Layer 不能处于 PRIMARY 模式。



## 6 错误码

---

错误码详见《55 - AX 软件错误码文档》文档。

AEXRA CONFIDENTIAL FOR SIPEED

# 7 调试信息

## 7.1 Layer 相关的信息查询

查询方法：cat /proc/ax\_proc/vo/layer\_status，如下图所示：

```
/root # cat /proc/ax_proc/vo/layer_status

----- VO INFO -----
[Axera version]: ax_vo V0.14.0_20231102170228 Nov  2 2023 17:17:04 JK

----- layer0 info -----
ID PoolId Stat SyncMode PrimChn InplceChn DispatchMode OutBufSel Reso BLK-Miss FpsSet FpsAct Interval WorkIn FBC LEVEL
0 1 EN NORMAL 0 0 FIFO POOL 800x480 0 60 55.43 16666 0 NONE 0

ID PartMode Fmt BgClrPx LQDepth LQFill LQTotal
0 SINGLE NV12 0x0 3 3 23288

Chn Stat Prior Pos(x,y) Reso Hide Pause KeepPrev FrCtrl Fmt FBDC LEVEL InuseFrOut PTS SeqNum
0 EN 0 0,0 800x480 N N Y N NV12 NONE 0 N 1 0

Chn CQDepth CQFill InFps OutFps TotalFrames RecvFrames DiscFrames ReptMax ADJ-CNT
0 3 1 33.33 33.38 14008 14008 0 2 0
```

表7-1 Layer 相关信息描述

字段名	描述	备注
ID	Layer 编号	
PoolId	当前 layer 输出所使用的 pool，只有 OutBufSel 为 AX_VO_LAYER_WB_POOL 才有意义	
Stat	Layer 的状态，如使能或关闭等	
SyncMode	Layer 的同步模式，由 Layer 属性中 enSyncMode 成员指定	
PrimChn	主通道号，只有同步模式为 AX_VO_LAYER_SYNC_PRIMARY 时才有意义	
InplceChn	Inplace 通道号，只有 OutBufSel 为 AX_VO_LAYER_WB_INPLACE 时才有意义	
DispatchMode	输出 buf 的分发方式，如 link 或 out fifo	
OutBufSel	输出 buf 的来源描述	
Reso	Layer 输出分辨率	

字段名	描述	备注
BLK-Miss	Layer 从 pool 中获取 VB 出错的次数统计	
FpsSet	Layer 所设置的帧率	
FpsAct	Layer 实际的帧率	
Interval	Layer 的帧间隔，此数值参与通道的帧控	
WorkIn	当前 Layer 工作在哪个硬件模块上	
FBC	输出帧的压缩类型	
LEVEL	输出帧的压缩等级	
PartMode	Layer 的分隔模式	
Fmt	Layer 输出格式	
BgClrPx	Layer 输出的背景色像素值	
LQDepth	Layer 输出队列的深度	
LQFill	Layer 输出队列上元素的个数	
LQTotal	Layer 输出的帧数统计	

表7-2 Chn 相关信息描述

字段名	描述	备注
Chn	通道号	
Stat	通道状态，如使能或关闭等	
Prior	通道的优先级信息	
Pos(x,y)	通道在 Layer 的位置信息	
Reso	通道的分辨率信息	
Hide	通道处于 Hide 状态	
Pause	通道处于 Pause 状态	
KeepPrev	通道处于 Keep-prev 状态	
FrCtrl	通道处于 Framerate-ctrl 状态	

字段名	描述	备注
Fmt	通道当前流转帧的格式	
FBDC	输入帧的压缩类型	
LEVEL	输入帧的压缩等级	
InuseFrOut	通道是否支持输出正在使用的帧	
PTS	图像时间戳	
SeqNum	帧序列号	

表7-3 Chn 相关信息描述 2

字段名	描述	备注
Chn	通道号	
CQDepth	通道输入队列的深度数	
CQFill	通道输入队列上元素的个数	
InFps	帧率统计，发送到通道的帧率，也为进入 FIFO 队列帧率统计。	
OutFps	帧率统计，通道输出的帧率，也为从 FIFO 队列拿出的帧率统计。	
TotalFrames	发送到通道的总帧数	
RecvFrames	成功接收到的总帧数	
DiscFrames	丢掉的总帧数	
ReptMax	重复的总帧数及单帧的最大重复次数统计	
ADJ-CNT	采样的起始点调整次数统计	

## 7.2 Display 相关的信息查询

查询方法：cat /proc/ax\_proc/vo/display\_status，如下图所示：

```
/root # cat /proc/ax_proc/vo/display_status

----- VO VERSION -----
[Axera version]: ax_vo V2.13.0_20240905213915 Sep  5 2024 21:52:06 JK

----- display0 stats -----
Mode      OutInfo      ComposeInfo  Irqs      IrqInterval  Refresh      HTimings      VTimings      Pclk      Polar      ScanType
OFFLINE    DPI-RG16      V0+G         19946      16664        60.06(60 )   800 -810 -840 -928  480 -493 -501 -533  29700      N-P-P      P
Layer Pos(x,y)  Reso      Fmt
V      0,0        800x480    NV12
Layer Pos(x,y)  Reso      Fmt      CkEn  CkInv  CkValL  CkValH
G      0,0        800x480    AR24     Y      N      65793   65793
Layer Pos(x,y)  Reso      Fmt      Stat Alpha Pixel  Inv InvPixel  InvThr
M      200,200    20x50     AR24     EN    0      0      DIS 0      0
```

表7-4 Display 相关信息描述

字段名	描述	备注
Mode	当前显示设备的工作模式，ONLINE 或 OFFLINE	
OutInfo	当前显示设备的输出接口类型及输出格式	
ComposeInfo	当前显示设备使用的硬件图层，如 V0、G0 或 V0+G0	
Irqs	当前显示设备产生的中断数	
IrqInterval	当前显示设备最大中断间隔，单位微秒，每次 Cat 后会清 0	
Refresh	当前显示设备的刷新率，括号外为实际刷新率，括号内为期望刷新率	
HTimings	当前显示设备所配置的水平时序信息	
VTimings	当前显示设备所配置的垂直时序信息	
Pclk	当前显示设备所配置的像素时钟信息，单位 kHz	
Polar	当前显示设备所配置的 DE/VSYNC/HSYNC 信号的极性，P 表示正极性，N 表示负极性	
ScanType	当前显示设备的扫描模式，P 表示逐行，I 表示隔行	

表7-5 Display 图层相关信息描述

字段名	描述	备注
Layer	图层类型，V 表示视频，G 表示图形，M 表示鼠标	
Pos(x,y)	图层在显示器上的位置信息	
Reso	图层分辨率信息	
Fmt	图层格式信息	
CkEn	图形层关键色使能状态，EN 表示使能，DIS 表示未使能	仅图形层有
CkInv	图形层关键色是否使能反向模式	仅图形层有且 CkEn 使能了才有意义
CkValL	图形层关键色的低临界	仅图形层有且 CkEn 使能了才有意义
CkValH	图形层关键色的高临界	仅图形层有且 CkEn 使能了才有意义
Stat	鼠标层使能状态，EN 表示使能，DIS 表示未使能	仅鼠标层有
Alpha	鼠标层的全局 Alpha，针对非 ARGB 格式才有意义	仅鼠标层有
Pixel	鼠标层的像素值，针对非 BITMAP 格式才有意义	仅鼠标层有
Inv	反色使能状态	仅鼠标层有
InvPixel	反色像素值	仅鼠标层有
InvThr	反色检测的阈值	仅鼠标层有

## 7.3 用户态相关 LOG 的控制

VO 用户态 LOG 主要是 libax\_vo 产生的，libax\_vo 支持动态控制 LOG 的输出，详细使用参考：

```

/ # cat /proc/ax_proc/logctl
***** Axera log control *****
Usage:
echo [ulog/klog] [id] [level] > /proc/ax_proc/logctl
ulog: user log klog: kernel log
id: [0-40, all] level: [0-7])
echo ulog/klog/all [on, off] > /proc/ax_proc/logctl
echo target [file, console, null] > /proc/ax_proc/logctl(user log only)
-----
klog_state: on
ulog_state: on
ulog_target: file
-----

```

module	id	level(u)	level(k)
ISP	1	4	4
CE	2	4	4
VO	3	4	4
VDSP	4	4	4
EFUSE	5	4	4
NPU	6	4	4
VENC	7	4	4
VDEC	8	4	4
JENC	9	4	4
JDEC	10	4	4
SYS	11	4	4
AENC	12	4	4
IVPS	13	4	4
MIPI	14	4	4
ADEC	15	4	4
DMA	16	4	4
VIN	17	4	4
USER	18	4	4
IVES	19	4	4
SKEL	20	4	4
IVE	21	4	4
3A	25	4	4
AUDIO	26	4	4
AI	32	4	4
AO	33	4	4
SENSOR	34	4	4
NT	35	4	4
TDP	36	4	4
VPP	37	4	4
VGP	38	4	4
GDC	39	4	4
BASE	40	4	4

```

/ #

```

从上图可以看出 log 的状态是打开的，默认存储到文件，VO 的当前 log 过滤等级为 4，即 VO\_WARN/VO\_ERROR 的均会输出。VO 目前使用的 log 等有 4 个，从小到大分别为：

SYS\_LOG\_ERROR(3)、SYS\_LOG\_WARN(4)、SYS\_LOG\_INFO(6)、SYS\_LOG\_DEBUG(7)

表7-6 AX\_VO 日志相关控制

命令	描述	备注
echo ulog 3 3 > /proc/ax_proc/logctl	VO 模块只输出错误信息。	Log 量最小
echo ulog 3 4 > /proc/ax_proc/logctl	VO 模块会输出错误和告警信息。	
echo ulog 3 6 > /proc/ax_proc/logctl	VO 模块会输出错误、告警和一般信息。	
echo ulog 3 7 > /proc/ax_proc/logctl	VO 模块会输出所有的信息，包含错误、告警、一般信息和调试信息。	Log 量最大
echo ulog console > /proc/ax_proc/logctl	用于控制 log 从控制台输出	

## 7.4 固定时序相关说明

为了用户使用方便，本芯片支持固定时序输出，具体的由 AX\_VO\_INTF\_SYNC\_E 定义，下表列出固定时序对应的像素时钟：

表7-7 VO 固定时序

时序名称	像素时钟(kHz)	备注
AX_VO_OUTPUT_576P50	27000	
AX_VO_OUTPUT_576I50	27000	
AX_VO_OUTPUT_480P60	27000	
AX_VO_OUTPUT_480I60	27000	
AX_VO_OUTPUT_720P25	37125	
AX_VO_OUTPUT_720P30	37125	
AX_VO_OUTPUT_720P50	74250	
AX_VO_OUTPUT_720P60	74250	
AX_VO_OUTPUT_1080P24	74250	



AX_VO_OUTPUT_1080P25	74250	
AX_VO_OUTPUT_1080P30	74250	
AX_VO_OUTPUT_1080P50	148500	
AX_VO_OUTPUT_1080P60	148500	
AX_VO_OUTPUT_640x480_60	25175	
AX_VO_OUTPUT_800x480_60	29232	
AX_VO_OUTPUT_800x600_60	40000	
AX_VO_OUTPUT_1024x600_60	52000	
AX_VO_OUTPUT_1024x768_60	65000	
AX_VO_OUTPUT_1280x1024_60	108000	
AX_VO_OUTPUT_1366x768_60	85500	
AX_VO_OUTPUT_1280x800_60	83500	
AX_VO_OUTPUT_1440x900_60	106500	
AX_VO_OUTPUT_1600x1200_60	162000	
AX_VO_OUTPUT_1680x1050_60	146250	
AX_VO_OUTPUT_1920x1200_60	154000	
AX_VO_OUTPUT_2560x1600_60	348500	
AX_VO_OUTPUT_3840x2160_24	297000	
AX_VO_OUTPUT_3840x2160_25	297000	
AX_VO_OUTPUT_3840x2160_30	297000	
AX_VO_OUTPUT_3840x2160_50	594000	
AX_VO_OUTPUT_3840x2160_60	594000	
AX_VO_OUTPUT_4096x2160_24	297000	
AX_VO_OUTPUT_4096x2160_25	297000	
AX_VO_OUTPUT_4096x2160_30	297000	

AX_VO_OUTPUT_4096x2160_50	594000	
AX_VO_OUTPUT_4096x2160_60	594000	
AX_VO_OUTPUT_720x1280_60	74250	
AX_VO_OUTPUT_1080x1920_60	148500	
AX_VO_OUTPUT_1080x1920_60	74250	

AEXRA CONFIDENTIAL FOR SIPEED