



AX VO API 文档

文档版本: V1.3

发布日期: 2021/05/13

AXERA CONFIDENTIAL FOR Sipeed

| | |
|--------------------------------|----|
| 前 言 | 4 |
| 目 录 | |
| 修订历史 | 5 |
| 1 概述 | 6 |
| 1.1 概述 | 7 |
| 1.2 功能描述 | 9 |
| 1.2.1 重要概念 | 9 |
| 2 API 简介 | 10 |
| 3 API 定义 | 12 |
| AX_VO_Init | 12 |
| AX_VO_Deinit | 13 |
| AX_VO_SetPubAttr | 14 |
| AX_VO_GetPubAttr | 15 |
| AX_VO_Enable | 16 |
| AX_VO_Disable | 17 |
| AX_VO_SetVideoLayerAttr | 18 |
| AX_VO_GetVideoLayerAttr | 19 |
| AX_VO_EnableVideoLayer | 20 |
| AX_VO_DisableVideoLayer | 21 |
| AX_VO_BindVideoLayer | 22 |
| AX_VO_UnBindVideoLayer | 23 |
| AX_VO_BindGraphicLayer | 24 |
| AX_VO_UnBindGraphicLayer | 25 |

| | |
|--------------------------------|-----------|
| AX_VO_GetLayerFd | 26 |
| AX_VO_GetLayerFrame | 27 |
| AX_VO_ReleaseLayerFrame | 29 |
| AX_VO_EnableChn | 30 |
| AX_VO_DisableChn | 31 |
| AX_VO_SetChnAttr | 32 |
| AX_VO_GetChnAttr | 33 |
| AX_VO_SendFrame | 34 |
| 4 数据结构 | 36 |
| VO_INTF_TYPE_E | 36 |
| AX_INTF_SYNC_E | 38 |
| VO_RESO_S | 41 |
| VO_PUB_ATTR_S | 42 |
| VO_RECT_S | 43 |
| VO_LAYER_SYNC_E | 45 |
| VO_LAYER_BUF_E | 47 |
| VO_LAYER_DISPATCH_MODE_E | 48 |
| VO_VIDEO_FRAME_S | 49 |
| VO_SIZE_S | 52 |
| VO_VIDEO_LAYER_ATTR_S | 53 |
| VO_CHN_ATTR_S | 55 |
| 5 错误码 | 56 |
| 6 调试信息 | 57 |

权利声明

爱芯元智半导体（上海）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。

AXERA CONFIDENTIAL FOR Speed

适用产品



爱芯 AX620A

前言

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/AX620A_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。 |

| 文档版本 | 发布时间 | 修订说明 |
|------|------------|----------------------------------|
| V1.0 | 2021/12/23 | 文档初版 |
| V1.1 | 2022/01/17 | 更新 GraphicLayer bind/unbind 接口说明 |
| V1.2 | 2022/04/13 | 增加通道属性说明、更新 VO_INTF_TYPE_E 结构体 |
| V1.3 | 2022/05/13 | 增加获取 Layer 对应 fd 的接口说明 |

修订历史

本章节包含：

[1.1 概述](#)

[表 1-2](#)

1 概述

AXERA CONFIDENTIAL FOR Sipeed

芯片显示输出能力

| 芯片型号 | 接口类型 | 最大输出能力 | 是否支持双显示 |
|--------|----------|------------------------------------|---------|
| AX620A | MIPI DSI | 单屏 4K DCI 30 Hz 双屏 2K DCI 60 Hz | 支持 |
| | DPI | 1080p 60Hz | 不支持 |
| | BT.1120 | 1080p 60Hz | 不支持 |
| | BT.656 | 1080p 60Hz | 不支持 |
| | BT.601 | 1080p 60Hz | 不支持 |

功能描述

AXERA CONFIDENTIAL FOR Sipeed

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。

不同型号的芯片支持不同的解码规格。芯片支持的解码规格如下表所示。

表1-1 芯片对图层的支持

| 芯片型号 | 图层 | 输入图像格式 | 通道 | 缩放 |
|--------|-------------|---|-------------------------|-----|
| AX620A | 视频层 (V0) | NV12 RGB888 RGB565 | 硬件支持单通道 软件支持最大 32 通道 | 不支持 |
| | 图形层 (G0) | ARGB1555 ARGB4444 RGB565_A8 RGB565 RGB888 RGBA4444 RGBA5551 RGBA8888 | 32 通道 | 不支持 |

表1-2 芯片显示输出能力

| 芯片型号 | 接口类型 | 最大输出能力 | 是否支持双显示 |
|--------|----------|------------------------------------|---------|
| AX620A | MIPI DSI | 单屏 4K DCI 30 Hz 双屏 2K DCI 60 Hz | 支持 |
| | DPI | 1080p 60Hz | 不支持 |
| | BT.1120 | 1080p 60Hz | 不支持 |
| | BT.656 | 1080p 60Hz | 不支持 |
| | BT.601 | 1080p 60Hz | 不支持 |

1.2 功能描述

1.2.1 重要概念

显示设备（VO DEV）

显示设备代表硬件显示控制器，用于将图像输出到显示终端上。

通道（Channel）

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）

图层可以理解为 VO 的输出图像，使用图层可以管理多个通道。图层与通道之间的关系如图所示。



图1-1 图层与通道之间的关系

VO 模块提供的 API 接口如下：

2 API 简介

- **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_Enable**: 使能指定的显示设备。
- **AX_VO_Disable**: 关闭指定的显示设备。
- **AX_VO_SetVideoLayerAttr**: 设置 VideoLayer 的属性。
- **AX_VO_GetVideoLayerAttr**: 获取 VideoLayer 已设置所属性。
- **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_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。

AXERA CONFIDENTIAL FOR Sipeed

AX_VO_Init

3 API 定义

【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及所有的 Layer 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 VO_PUB_ATTR_S *pstPubAttr)
```

【参数】

| 参数名称 | 描述 | 输入/输出 |
|------------|-------------|-------|
| VoDev | 显示设备号 | 输入 |
| pstPubAttr | 显示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】

- 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。
- 设备号必须为实际所支持的设备的对应的设备号。

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, VO_PUB_ATTR_S *pstPubAttr)
```

【参数】

| 参数名称 | 描述 | 输入/输出 |
|------------|-------------|-------|
| VoDev | 显示设备号 | 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】

- 设备号必须为实际所支持的设备的对应的设备号。

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 | 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】

- 设备号必须为实际所支持的设备的对应的设备号。
- 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 | 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】

- 设备号必须为实际所支持的设备的对应的设备号。

AX_VO_SetVideoLayerAttr

【描述】

设置指定 VideoLayer 的属性。

【语法】

```
AX_S32 AX_VO_SetVideoLayerAttr(VO_LAYER VoLayer, const VO_VIDEO_LAYER_ATTR_S
*pstLayerAttr)
```

【参数】

| 参数名称 | 描述 | 输入/输出 |
|--------------|---------------|-------|
| VoLayer | VoLayer 编号 | 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】

- VoLayer 编号必须在所支持的范围内。
- 调用此接口前必须确保所对应的 VoLayer 没有被使用。

AX_VO_GetVideoLayerAttr

【描述】

获取指定 VideoLayer 的属性。

【语法】

```
AX_S32 AX_VO_GetVideoLayerAttr(VO_LAYER VoLayer, VO_VIDEO_LAYER_ATTR_S
*pstLayerAttr)
```

【参数】

| 参数名称 | 描述 | 输入/输出 |
|--------------|---------------|-------|
| VoLayer | VoLayer 编号 | 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】

- VoLayer 编号必须在所支持的范围内。

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 | VoLayer 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 编号必须在所支持的范围内。
- 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 | VoLayer 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】

- VideoLayer 编号必须在所支持的范围内。

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 | VideoLayer 编号 | 输入 |
| VoDev | 显示设备编号 | 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 必须在所支持的范围内。
- VoDev 必须在所支持的范围内。
- 支持重复绑定，但不支持同一个 VideoLayer 绑定到多个 VoDev 上。如需换 VoDev，需先与前一个 VoDev 解绑。

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 | 解码通道号 | 输入 |
| VoDev | 协议相关参数 | 输入 |

【返回值】

| 返回值 | 描述 |
|---------|---------------|
| VoLayer | VideoLayer 编号 |
| VoDev | 显示设备编号 |

【需求】

- 头文件：ax_vo_api.h
- 库文件：libax_vo.so

【注意】

- VoLayer 必须在所支持的范围内。
- 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】

- GraphicLayer 必须在所支持的范围内。
- VoDev 必须在所支持的范围内。
- 支持重复绑定，但不支持同一个 GraphicLayer 绑定到多个 VoDev 上。如需换 VoDev，需先与前一个 VoDev 解绑。

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】

- GraphicLayer 必须在所支持的范围内。
- VoDev 必须在所支持的范围内。

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 | VideoLayer 编号 | 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】

- VoLayer 必须在所支持的范围内。
- 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_S *pstVoFrame,  
AX_S32 s32MS)
```

【参数】

| 参数名称 | 描述 | 输入/输出 |
|------------|--|-------|
| VoLayer | VideoLayer 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】

- VoLayer 必须在所支持的范围内。
- 目前参数 s32MS 用于将来使用，暂时不支持，该接口暂时为非阻塞类型，因此使用时需根据返回值来判断是否获取到有效帧。

AXERA CONFIDENTIAL FOR Sipeed

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_S
*pstVoFrame)
```

【参数】

| 参数名称 | 描述 | 输入/输出 |
|------------|------------------------|-------|
| VoLayer | VideoLayer 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_GetLayerFrame 配对使用。

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 | VideoLayer 编号 | 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 必须在所支持的范围内且所对应的 VideoLayer 必须已使能。
- VoChn 必须在所对应 VideoLayer 所支持的通道范围内，VideoLayer 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 | VideoLayer 编号 | 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 必须在所支持的范围内且所对应的 VideoLayer 必须已使能。
- VoChn 必须在所对应 VideoLayer 所支持的通道范围内，VideoLayer 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 VO_CHN_ATTR_S
*pstChnAttr)
```

【参数】

| 参数名称 | 描述 | 输入/输出 |
|------------|---------------|-------|
| VoLayer | VideoLayer 编号 | 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 必须在所支持的范围内且所对应的 VideoLayer 必须已使能。
- VoChn 必须在所对应 VideoLayer 所支持的通道范围内，VideoLayer 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, VO_CHN_ATTR_S  
*pstChnAttr)
```

【参数】

| 参数名称 | 描述 | 输入/输出 |
|------------|---------------|-------|
| VoLayer | VideoLayer 编号 | 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 必须在所支持的范围内且所对应的 VideoLayer 必须已使能。
- VoChn 必须在所对应 VideoLayer 所支持的通道范围内，VideoLayer 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_S  
*pstVoFrame, AX_S32 s32MS)
```

【参数】

| 参数名称 | 描述 | 输入/输出 |
|------------|--|-------|
| VoLayer | VideoLayer 编号 | 输入 |
| VoChn | 通道编号 | 输入 |
| pstVoFrame | 图像帧结构体指针 | 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 必须在所支持的范围内且所对应的 VideoLayer 必须已使能。
- VoChn 必须在所对应 VideoLayer 所支持的通道范围内且对应的通道必须已使能，VideoLayer 所支持的通道数可通过相关属性来指定。
- 目前参数 s32MS 用于将来使用，暂时不支持，该接口暂时为非阻塞类型。

AXERA CONFIDENTIAL FOR Sipeed

VO_INTF_TYPE_E

4 数据结构

【说明】

枚举显示设备接口类型。

【定义】

```
typedef enum {  
  
    VO_INTF_VIRT = 0,  
  
    VO_INTF_DPI,  
  
    VO_INTF_DSI0,  
  
    VO_INTF_DSI1,  
  
    VO_INTF_2DSI,  
  
    VO_INTF_BT656,  
  
    VO_INTF_BT1120,  
  
    VO_INTF_BUTT  
} VO_INTF_TYPE_E;
```

【成员】

| 成员名称 | 描述 |
|--------------|-----------------------|
| VO_INTF_VIRT | 虚拟类型，目前不支持 |
| VO_INTF_DPI | 显示硬件设备通过 DPI 输出 |
| VO_INTF_DSI0 | 显示硬件设备通过 MIPI DSI0 输出 |

| 成员名称 | 描述 |
|----------------|-----------------------------|
| VO_INTF_DSI1 | 显示硬件设备通过 MIPI DSI1 输出 |
| VO_INTF_2DSI | 显示硬件设备通过 MIPI DSI0/DSI1 双输出 |
| VO_INTF_BT656 | 显示硬件设备通过 BT656 输出 |
| VO_INTF_BT1120 | 显示硬件设备通过 BT1120 输出 D |

【相关数据类型及接口】[AX_VO_SetPubAttr](#)

AXERA CONFIDENTIAL FOR Sipeed

AX_INTF_SYNC_E

【说明】

枚举显示设备的分辨率类型。

【定义】

```
typedef enum {  
  
    VO_OUTPUT_USER = 1,  
  
    VO_OUTPUT_480P60,  
  
    VO_OUTPUT_576P50,  
  
    VO_OUTPUT_720P24,  
  
    VO_OUTPUT_720P25,  
  
    VO_OUTPUT_720P30,  
  
    VO_OUTPUT_720P50,  
  
    VO_OUTPUT_720P60,  
  
    VO_OUTPUT_800P60,  
  
    VO_OUTPUT_1080P24,  
  
    VO_OUTPUT_1080P25,  
  
    VO_OUTPUT_1080P30,  
  
    VO_OUTPUT_1080P50,  
  
    VO_OUTPUT_1080P60,  
  
    VO_OUTPUT_3840x2160_24,  
  
    VO_OUTPUT_3840x2160_25,  
  
    VO_OUTPUT_3840x2160_30,  
  
}
```

```
VO_OUTPUT_3840x2160_50,  
VO_OUTPUT_3840x2160_60,  
VO_OUTPUT_4096x2160_24,  
VO_OUTPUT_4096x2160_25,  
VO_OUTPUT_4096x2160_30,  
VO_OUTPUT_4096x2160_50,  
VO_OUTPUT_4096x2160_60,  
VO_OUTPUT_BUTT,  
} VO_INTF_SYNC_E;
```

【成员】

| 成员名称 | 描述 |
|-------------------|----------------------------------|
| VO_OUTPUT_USER | 自定义分辨率类型，当用户所选的分辨率不在此枚举支持的类型中时使用 |
| VO_OUTPUT_480P60 | 分辨率为 720x480，刷新率为 60Hz |
| VO_OUTPUT_576P50 | 分辨率为 720x576，刷新率为 50Hz |
| VO_OUTPUT_720P24 | 分辨率为 1280x720，刷新率为 24Hz |
| VO_OUTPUT_720P25 | 分辨率为 1280x720，刷新率为 25Hz |
| VO_OUTPUT_720P30 | 分辨率为 1280x720，刷新率为 30Hz |
| VO_OUTPUT_720P50 | 分辨率为 1280x720，刷新率为 50Hz |
| VO_OUTPUT_720P60 | 分辨率为 1280x720，刷新率为 60Hz |
| VO_OUTPUT_800P60 | 分辨率为 1280x800，刷新率为 60Hz |
| VO_OUTPUT_1080P24 | 分辨率为 1920x1080，刷新率为 24Hz |
| VO_OUTPUT_1080P25 | 分辨率为 1920x1080，刷新率为 25Hz |
| VO_OUTPUT_1080P30 | 分辨率为 1920x1080，刷新率为 30Hz |

| 成员名称 | 描述 |
|------------------------|--------------------------|
| VO_OUTPUT_1080P50 | 分辨率为 1920x1080，刷新率为 50Hz |
| VO_OUTPUT_1080P60 | 分辨率为 1920x1080，刷新率为 60Hz |
| VO_OUTPUT_3840x2160_24 | 分辨率为 3840x2160，刷新率为 24Hz |
| VO_OUTPUT_3840x2160_25 | 分辨率为 3840x2160，刷新率为 25Hz |
| VO_OUTPUT_3840x2160_30 | 分辨率为 3840x2160，刷新率为 30Hz |
| VO_OUTPUT_3840x2160_50 | 分辨率为 3840x2160，刷新率为 50Hz |
| VO_OUTPUT_3840x2160_60 | 分辨率为 3840x2160，刷新率为 60Hz |
| VO_OUTPUT_4096x2160_24 | 分辨率为 4096x2160，刷新率为 24Hz |
| VO_OUTPUT_4096x2160_25 | 分辨率为 4096x2160，刷新率为 25Hz |
| VO_OUTPUT_4096x2160_30 | 分辨率为 4096x2160，刷新率为 30Hz |
| VO_OUTPUT_4096x2160_50 | 分辨率为 4096x2160，刷新率为 50Hz |
| VO_OUTPUT_4096x2160_60 | 分辨率为 4096x2160，刷新率为 60Hz |

【相关数据类型及接口】[AX_VO_SetPubAttr](#)

VO_RESO_S

【说明】

描述 VO 分辨率、刷新率相关的结构体定义。

【定义】

```
typedef struct axVO_RESO_S {  
  
    AX_U32 u32Width;  
  
    AX_U32 u32Height;  
  
    AX_U32 u32RefreshRate;  
  
} VO_RESO_S ;
```

【成员】

| 成员名称 | 描述 |
|----------------|------|
| u32Width | 分辨率宽 |
| u32Height | 分辨率高 |
| u32RefreshRate | 刷新率 |

【相关数据类型及接口】

➤ [AX_VO_SetPubAttr](#)

VO_PUB_ATTR_S

【说明】

显示设备相关属性结构体定义。

【定义】

```
typedef struct axVO_PUB_ATTR_S {  
  
    VO_INTF_TYPE_E enIntfType;  
  
    VO_INTF_SYNC_E enIntfSync;  
  
    VO_RESO_S stReso;  
  
} VO_PUB_ATTR_S;
```

【成员】

| 成员名称 | 描述 |
|------------|--|
| enIntfType | 显示设备接口类型 |
| enIntfSync | 显示设备的分辨率类型 |
| stReso | 用户指定的分辨率类型，当 enIntfSync 指定为 VO_OUTPUT_USER 时，使用此成员来指定分辨率及刷新率 |

VO_RECT_S

【说明】

描述 VO 矩形区域相关的结构体定义。

【定义】

```
typedef struct axVO_RECT_S {  
  
    AX_U32 u32X;  
  
    AX_U32 u32Y;  
  
    AX_U32 u32Width;  
  
    AX_U32 u32Height;  
  
} VO_RECT_S ;
```

【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](#)

AXERA CONFIDENTIAL FOR Sipeed

VO_LAYER_SYNC_E

【说明】

VideoLayer 的 SYNC 类型定义，用于指示 Layer 何时去提交相关的数据到硬件。

【定义】

```
typedef enum {
```

```
    VO_LAYER_SYNC_NORMAL,
```

```
    VO_LAYER_SYNC_SHUTTLE,
```

```
    VO_LAYER_SYNC_GROUPING,
```

```
    VO_LAYER_SYNC_PRIMARY,
```

```
    VO_LAYER_SYNC_BUTT,
```

```
} VO_LAYER_SYNC_E;
```

【成员】

| 成员名称 | 描述 |
|------------------------|--|
| VO_LAYER_SYNC_NORMAL | 该模式下，所有使能的通道中均有数据才去提交硬件 |
| VO_LAYER_SYNC_SHUTTLE | 该模式下，Layer 将按其属性设置的帧率去提交所使能通道中的数据到硬件，目前版本暂未支持 |
| VO_LAYER_SYNC_GROUPING | 该模式下，Layer 将根据所有使能通道中帧信息的时间戳来判断，所有的时间戳一致时提交到硬件，目前版本暂未支持 |
| VO_LAYER_SYNC_PRIMARY | 该模式下，Layer 属性中会指定一个通道当作 PRIMARY 通道，只要当 PRIMARY 通道中有数据送来时就会提交硬件 |

AXERA CONFIDENTIAL FOR Sipeed

VO_LAYER_BUF_E

【说明】

定义 VideoLayer 输出 buffer 的来源。

【定义】

```
typedef enum {  
  
    VO_LAYER_OUT_BUF_POOL,  
  
    VO_LAYER_OUT_BUF_INPLACE,  
  
    VO_LAYER_OUT_BUF_BUTT,  
  
} VO_LAYER_BUF_E;
```

【成员】

| 成员名称 | 描述 |
|--------------------------|----------------------------------|
| VO_LAYER_OUT_BUF_POOL | VideoLayer 的输出 buffer 从 pool 中获取 |
| VO_LAYER_OUT_BUF_INPLACE | VideoLayer 的输出 buffer 来自于输入通道 |

VO_LAYER_DISPATCH_MODE_E

【说明】

定义 VideoLayer 输出的分发模式。

【定义】

```
typedef enum {  
  
    VO_LAYER_OUT_TO_FIFO = 1,  
  
    VO_LAYER_OUT_TO_LINK = 2,  
  
} VO_LAYER_DISPATCH_MODE_E;
```

【成员】

| 成员名称 | 描述 |
|----------------------|---------------------------------|
| VO_LAYER_OUT_TO_FIFO | VideoLayer 的输出分发到其自带的 FIFO 中 |
| VO_LAYER_OUT_TO_LINK | VideoLayer 的输出分发到 VO 已 link 的模块 |

VO_VIDEO_FRAME_S

【说明】

定义每个输入通道解码图像信息。

【定义】

```
typedef struct axVIDEO_FRAME_S {  
  
    AX_U32          u32Width;  
  
    AX_U32          u32Height;  
  
    AX_IMG_FORMAT_E  enImgFormat;  
  
    AX_VSCAN_FORMAT_E  enVscanFormat;  
  
    AX_COMPRESS_MODE_E  enCompressMode;  
  
    AX_U32          u32HeaderStride[3];  
  
    AX_U32          u32PicStride[3];  
  
    AX_U32          u32ExtStride[3];  
  
    AX_U64          u64HeaderPhyAddr[3];  
  
    AX_U64          u64HeaderVirAddr[3];  
  
    AX_U64          u64PhyAddr[3];  
  
    AX_U64          u64VirAddr[3];  
  
    AX_U64          u64ExtPhyAddr[3];  
  
    AX_U64          u64ExtVirAddr[3];  
  
    AX_S16          s16OffsetTop;  
  
    AX_S16          s16OffsetBottom;  
  
    AX_S16          s16OffsetLeft;
```

```

    AX_S16          s16OffsetRight;

    AX_U32          u32TimeRef;

    AX_U64          u64PTS;

    AX_U64          u64SeqNum;

    AX_U64          u64UserData;

    AX_U64          u64PrivateData;

    AX_U32          u32FrameFlag;

    AX_U8           u8MetaDataInternal[128];

    AX_U8           u8MetaDataUser[128];

    AX_U32          u32FrameSize;

} AX_VIDEO_FRAME_S;

```

【成员】

| 成员名称 | 描述 |
|---------------------|--|
| u32Width | 解码图像的宽度 |
| u32Height | 解码图像的高度 |
| enImgFormat | 解码图像输出格式 |
| enVscanFormat | 视频图像像素格式 |
| enCompressMode | 解码图像的压缩模式 |
| u32HeaderStride[3] | 图像压缩头跨距 |
| u32PicStride[3] | 图像数据跨距 |
| u32ExtStride[3] | 10bit 数据位宽的图像，有些数据格式的存储方式是前 8bit 和后 2bit 分开存储，这里指后 2bit 数据跨距 |
| u64HeaderPhyAddr[3] | 压缩头物理地址 |
| u64HeaderVirAddr[3] | 压缩头虚拟地址 |

| 成员名称 | 描述 |
|------------------|---|
| u64PhyAddr[3] | 物理地址 |
| u64VirAddr[3] | 虚拟地址 |
| u64ExtPhyAddr[3] | 10bit 数据位宽的图像，有些数据格式的存储方式是前 8bit 和后 2bit 分开存储，这里指后 2bit 数据的物理地址 |
| u64ExtVirAddr[3] | 10bit 数据位宽的图像，有些数据格式的存储方式是前 8bit 和后 2bit 分开存储，这里指后 2bit 数据的虚拟地址 |
| s16OffsetTop | 图像顶部裁剪宽度 |
| s16OffsetBottom | 图像底部裁剪宽度 |
| s16OffsetLeft | 图像左侧裁剪宽度 |
| s16OffsetRight | 图像右侧裁剪宽度 |
| u32TimeRef | 图像帧序列号 |
| u64PTS | 图像时间戳 |
| u64SeqNum | 图像序列号 |
| u64UserData | 用户数据 |
| u64PrivateData | 私有数据 |
| u32FrameFlag | 当前帧标记 |
| u32FrameSize | 图像帧大小 |

VO_SIZE_S

【说明】

定义 VO 图像 SIZE 相关的结构体。

【定义】

```
typedef struct axVO_SIZE_S {  
  
    AX_U32 u32Width;  
  
    AX_U32 u32Height;  
  
} VO_SIZE_S;
```

【成员】

| 成员名称 | 描述 |
|-----------|------|
| u32Width | 图像的宽 |
| u32Height | 图像的高 |

VO_VIDEO_LAYER_ATTR_S

【说明】

定义 VideoLayer 相关的属性。

【定义】

```
typedef struct axVO_VIDEO_LAYER_ATTR_S {  
  
    VO_RECT_S stDispRect;  
  
    VO_SIZE_S stImageSize;  
  
    AX_IMG_FORMAT_E enPixFmt;  
  
    VO_LAYER_SYNC_E enLayerSync;  
  
    AX_U32 u32PrimaryChnId;  
  
    AX_U32 u32FrameRate;  
  
    AX_U32 u32FifoDepth;  
  
    AX_U32 u32ChnNr;  
  
    AX_U32 u32BkClr;  
  
    VO_LAYER_BUF_E enLayerBuf;  
  
    AX_U32 u32InplaceChnId;  
  
    AX_POOL u32PoolId;  
  
    VO_LAYER_DISPATCH_MODE_E enDispatchMode;  
  
    AX_U64 u64KeepChnPrevFrameBitmap;  
  
} VO_VIDEO_LAYER_ATTR_S;
```

【成员】

| 成员名称 | 描述 |
|---------------------------|--|
| stDispRect | 指定 VideoLayer 输出图像用于显示的区域范围，该范围不能超出 stImageSize 所限定的范围 |
| stImageSize | 指定 VideoLayer 输出图像的宽和高 |
| enPixFmt | 指定 VideoLayer 输出图像的格式，只支持 NV12/NV21 |
| enLayerSync | 指定 VideoLayer 的 Sync 类型 |
| u32PrimaryChnId | 当 enLayerSync 为 VO_LAYER_SYNC_PRIMARY 时，此成员用于指定对应的通道 ID |
| u32FrameRate | 当 enLayerSync 为 VO_LAYER_SYNC_SHUTTLE 时，此成员用于指定 VideoLayer 的帧率 |
| u32FifoDepth | 指定 VideoLayer 输出 FIFO 的深度 |
| u32ChnNr | 指定 VideoLayer 支持的通道数，目前最大为 32 |
| u32BkClr | 指定 VideoLayer 的背景色，格式为 RGB888 |
| enLayerBuf | 指定 VideoLayer 的输出 buffer 从哪里获取 |
| u32InplaceChnId | 当 enLayerBuf 为 VO_LAYER_OUT_BUF_INPLACE 时，此成员用于指定对应的通道 ID |
| u32PoolId | 当 enLayerBuf 为 VO_LAYER_OUT_BUF_POOL 时，此成员用于指定对应的 Pool ID |
| enDispatchMode | 指定 VideoLayer 的分发模式 |
| u64KeepChnPrevFrameBitmap | 用于指定 VideoLayer 上通道在没有获取到数据时是否使用前一帧，如 $1 \ll (n)$ 表示通道 n 使用前一帧 |

VO_CHN_ATTR_S

【说明】

定义 VideoLayer 上通道相关的属性。

【定义】

```
typedef struct axVO_CHN_ATTR_S {  
  
    VO_RECT_S stRect;  
  
    AX_U32 u32FifoDepth;  
  
} VO_CHN_ATTR_S;
```

【成员】

| 成员名称 | 描述 |
|--------------|--------------------------------------|
| stRect | 指定通道在 VideoLayer 上的区域范围 |
| u32FifoDepth | 指定通道上 FIFO 的深度，范围[0-8]，等于 0 时使用默认值 3 |

VO API 错误码如下表所示。

5 错误码

表5-1 VO API 错误码

| 错误代码 | 宏定义 | 描述 |
|------------|---------------------------|------------------|
| 0x80038A02 | AX_ERR_VO_INVALID_DEVID | 无效显示设备 ID |
| 0x80038A04 | AX_ERR_VO_INVALID_CHNID | 无效的通道 ID |
| 0x80038A0A | AX_ERR_VO_ILLEGAL_PARAM | 参数非法 |
| 0x80038A0B | AX_ERR_VO_NULL_PTR | 空指针 |
| 0x80038A15 | AX_ERR_VO_NOT_PERM | 操作不允许 |
| 0x80038A17 | AX_ERR_VO_NODEV | 没有 VO DEV |
| 0x80038A18 | AX_ERR_VO_NOMEM | 没有 Memory |
| 0x80038A80 | AX_ERR_VO_DEV_OP_FAIL | VO DEV 相关的操作失败 |
| 0x80038A81 | AX_ERR_VO_LAYER_OP_FAIL | VO LAYER 相关的操作失败 |
| 0x80038A82 | AX_ERR_VO_CHN_OP_FAIL | VO CHN 相关的操作失败 |
| 0x80038A83 | AX_ERR_VO_INVALID_LAYERID | 无效的 LAYER ID |
| 0x80038A84 | AX_ERR_VO_OPEN_FILE_FAIL | 打开 VO 相关的文件失败 |

6 调试信息

AXERA CONFIDENTIAL FOR Sipeed