



AX 公共数据结构文档

文档版本：V1.6

发布日期：2024/08/08

目 录

前 言	4
修订历史.....	5
1 概述.....	6
1.1 概述.....	6
1.2 重要概念.....	6
2 数据结构	7
AX_FRAME_RATE_CTRL_T.....	7
AX_LOG_LEVEL_E	8
AX_LOG_TARGET_E	9
AX_CHIP_TYPE_E	10
AX_PAYLOAD_TYPE_E	11
AX_COMPRESS_MODE_E	14
AX_FRAME_COMPRESS_INFO_T	15
AX_DYNAMIC_RANGE_E.....	16
AX_COLOR_GAMUT_E.....	17
AX_ROTATION_E.....	18
AX_IMG_FORMAT_E	19
AX_VIN_IVPS_MODE_E	24
AX_VIDEO_FRAME_T	25
AX_MOD_ID_E.....	28
AX_LINK_MODE_E	30
AX_VIDEO_FRAME_INFO_T	31
AX_NOTIFY_EVENT_E.....	32

AX_SYS_CLK_ID_E.....	33
AX_MOD_INFO_T.....	34
AX_LINK_DEST_T.....	35
AX_MEMORY_SOURCE_E.....	36
AX_MEMORY_ADDR_T.....	37
AX OSD BMP ATTR_T	38
AX_ERR_CODE_E.....	40
AX_COORD_E.....	41
AX_BGCOLOR_T.....	42
AX_COLORKEY_T.....	43
AX_BITCOLOR_T.....	44
AX_OVERLAY_T.....	45
AX_CSC_MATRIX_TYPE_E.....	46
AX_CSC_MATRIX_T.....	47
AX_MAX_COLOR_COMPONENT.....	48
AX_VSCAN_FORMAT_E.....	49
AX_GDC_MODE_E.....	50
AX_VDSP_MODE_E.....	51
AX_WARP_ENGINE_E.....	52
AX_WARP_MODE_U.....	53

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

- 技术支持工程师
- 软件开发工程师

符号与格式定义

符号/格式	说明
xxx	表示您可以执行的命令行。
斜体	表示变量。如，“安装目录/AX630C_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	2023/10/31	文档初版
V1.1	2024/1/29	新增 AX_CSC_MATRIX_T 结构体
V1.2	2024/3/15	更改 AX OSD BMP ATTR T 的 Alpha 变量的范围[0,255] ->(0,255]
V1.3	2024/07/05	修改 AX_CHIP_TYPE_E 枚举成员
V1.4	2024/07/25	修改 AX_VIDEO_FRAME_T, 完善相关成员变量
V1.5	2024/08/07	修改 AX_ERR_CODE_E 枚举成员
V1.6	2024/08/08	修改 AX_IMG_FORMAT_E 数据格式

1 概述

1.1 概述

本文档旨在介绍公共数据结构。

1.2 重要概念

表1-1 重要概念

缩写	全称
VIN	Video Input 视频输入模块
VDEC	Video Decoder 视频解码模块
VENC	Video Encoder 视频编码模块
IVPS	Image Video Process System 视频图像处理子系统
VB	Video Buffer , ISP Pipeline 使用帧 Buffer

2 数据结构

AX_FRAME_RATE_CTRL_T

【说明】

帧率控制信息。

【定义】

```
typedef struct _AX_FRAME_RATE_CTRL_T_ {  
    AX_F32 fSrcFrameRate;  
    AX_F32 fDstFrameRate;  
} AX_FRAME_RATE_CTRL_T;
```

【成员】

成员名称	描述
fSrcFrameRate	输入帧率控制参数。
fDstFrameRate	输出帧率控制参数。

【注意】

- 可使用浮点进行帧率控制。
- 可以使用宏 AX_INVALID_FRMRATE 赋值 0。
- 没有帧率控制时，fSrcFrameRate、fDstFrameRate 均设置为 0。
- 抓拍机方案中，建议不要使用帧率控制，否则会造成闪光信号触发的延时，不能及时响应抓拍。
- 组帧率控制和通道帧控控制可以同时生效，通道的输入帧率需要和组的输出帧率相同。
- fSrcFrameRate、fDstFrameRate 不可以小于 0

AX_LOG_LEVEL_E

【说明】

定义 log 输出级别。

【定义】

```
typedef enum {
    SYS_LOG_MIN          = -1,
    SYS_LOG_EMERGENCY    = 0,
    SYS_LOG_ALERT         = 1,
    SYS_LOG_CRITICAL      = 2,
    SYS_LOG_ERROR         = 3,
    SYS_LOG_WARN          = 4,
    SYS_LOG_NOTICE        = 5,
    SYS_LOG_INFO          = 6,
    SYS_LOG_DEBUG         = 7,
    SYS_LOG_MAX
};
```

【注意】

无

AX_LOG_TARGET_E

【说明】

定义 log 输出目标。

【定义】

```
typedef enum {
    SYS_LOG_TARGET_MIN = 0,
    SYS_LOG_TARGET_STDERR = 1,
    SYS_LOG_TARGET_SYSLOG = 2,
    SYS_LOG_TARGET_SYSLOG = 3,
    SYS_LOG_TARGET_MAX
} AX_LOG_TARGET_E;
```

【成员】

成员名称	描述
SYS_LOG_TARGET_STDERR	日志输出到标准错误。
SYS_LOG_TARGET_SYSLOG	日志输出到 syslog 文件。
SYS_LOG_TARGET_SYSLOG	不输出日志，等价于用户态日志关闭。

【注意】

无

AX_CHIP_TYPE_E

【说明】

定义芯片类型

【定义】

```
typedef enum {
    NONE_CHIP_TYPE = 0x0,
    AX620Q_CHIP = 0x1,
    AX620QX_CHIP = 0x2,
    AX630C_CHIP = 0x4,
    AX620E_CHIP_MAX
} AX_CHIP_TYPE_E;
```

【注意】

无

AX_PAYLOAD_TYPE_E

【说明】

定义解码通道类型枚举。

【定义】

```
typedef enum
{
    PT_PCMU          = 0,
    PT_1016          = 1,
    PT_G721          = 2,
    PT_GSM           = 3,
    PT_G723          = 4,
    PT_DVI4_8K       = 5,
    PT_DVI4_16K      = 6,
    PT_LPC           = 7,
    PT_PCMA          = 8,
    PT_G722          = 9,
    PT_S16BE_STEREO = 10,
    PT_S16BE_MONO   = 11,
    PT_QCELP          = 12,
    PT_CN            = 13,
    PT_MPEGAUDIO     = 14,
    PT_G728          = 15,
    PT_DVI4_3         = 16,
    PT_DVI4_4         = 17,
    PT_G729          = 18,
    PT_G711A          = 19,
    PT_G711U          = 20,
    PT_G726          = 21,
    PT_G729A          = 22,
    PT_LPCM          = 23,
    PT_CelB          = 25,
    PT_JPEG          = 26,
    PT_CUSM          = 27,
    PT_NV            = 28,
    PT_PICW          = 29,
    PT_CPV           = 30,
}
```

PT_H261	= 31,
PT_MPEGVIDEO	= 32,
PT_MPEG2TS	= 33,
PT_H263	= 34,
PT_SPEG	= 35,
PT_MPEG2VIDEO	= 36,
PT_AAC	= 37,
PT_WMA9STD	= 38,
PT_HEAAC	= 39,
PT_PCM_VOICE	= 40,
PT_PCM_AUDIO	= 41,
PT_AACLC	= 42,
PT_MP3	= 43,
PT_ADPCM	= 49,
PT_AEC	= 50,
PT_X_LD	= 95,
PT_H264	= 96,
PT_D_GSM_HR	= 200,
PT_D_GSM_EFR	= 201,
PT_D_L8	= 202,
PT_D_RED	= 203,
PT_D_VDVI	= 204,
PT_D_BT656	= 220,
PT_D_H263_1998	= 221,
PT_D_MP1S	= 222,
PT_D_MP2P	= 223,
PT_D_BMPEG	= 224,
PT_MP4VIDEO	= 230,
PT_MP4AUDIO	= 237,
PT_VC1	= 238,
PT_JVC ASF	= 255,
PT_D_AVI	= 256,
PT_DIVX3	= 257,
PT_AVIS	= 258,
PT_REAL8	= 259,
PT_REAL9	= 260,
PT_VP6	= 261,
PT_VP6F	= 262,
PT_VP6A	= 263,

```
PT_SORENSEN      = 264,
PT_H265          = 265,
PT_VP8           = 266,
PT_MVC           = 267,
PT_PNG           = 268,
PT_AVS2          = 269,
PT_VP7           = 270,
PT_VP9           = 271,
PT_AMR           = 1001,
PT_MJPEG         = 1002,
PT_AMRWB         = 1003,
PT_PRORES        = 1006,
PT_OPUS          = 1007,
PT_BUTT          =
} AX_PAYLOAD_TYPE_E;
```

【注意】

目前 VDEC 模块仅支持 PT_H264、PT_JPEG、PT_MJPEG。

AX_COMPRESS_MODE_E

【说明】

定义图像压缩模式。

【定义】

```
typedef enum
{
    AX_COMPRESS_MODE_NONE = 0,
    AX_COMPRESS_MODE_LOSSLESS,
    AX_COMPRESS_MODE_LOSSY,
    AX_COMPRESS_MODE_BUTT
} AX_COMPRESS_MODE_E;
```

【成员】

成员名称	描述
AX_COMPRESS_MODE_NONE	无压缩
AX_COMPRESS_MODE_LOSSLESS	无损压缩，省带宽但不省内存
AX_COMPRESS_MODE_LOSSY	有损压缩，即省带宽又省内存

【注意】

无

AX_FRAME_COMPRESS_INFO_T

【说明】

定义图像压缩模式和等级。

【定义】

```
typedef struct axFRAME_COMPRESS_INFO_T {  
    AX_COMPRESS_MODE_E enCompressMode;  
    AX_U32    u32CompressLevel;  
} AX_FRAME_COMPRESS_INFO_T;
```

【成员】

成员名称	描述
enCompressMode	压缩模式
u32CompressLevel	有损压缩模式的压缩等级，范围：[1, 8]

【注意】

无

AX_DYNAMIC_RANGE_E

【说明】

动态范围类型。

【定义】

```
typedef enum axDYNAMIC_RANGE_E
{
    AX_DYNAMIC_RANGE_SDR8 = 0,
    AX_DYNAMIC_RANGE_SDR10,
    AX_DYNAMIC_RANGE_HDR10,
    AX_DYNAMIC_RANGE_HLG,
    AX_DYNAMIC_RANGE_SLF,
    AX_DYNAMIC_RANGE_XDR,
    AX_DYNAMIC_RANGE_BUTT
} AX_DYNAMIC_RANGE_E;
```

【成员】

成员名称	描述
AX_DYNAMIC_RANGE_SDR8	8bit 数据的标准动态范围
AX_DYNAMIC_RANGE_SDR10	10bit 数据的标准动态范围
AX_DYNAMIC_RANGE_HDR10	10bit 数据的高动态范围
AX_DYNAMIC_RANGE_HLG	10bit 数据的高动态范围
AX_DYNAMIC_RANGE_SLF	暂时无效
AX_DYNAMIC_RANGE_XDR	10bit 数据，算法处理的一个中间类型数据，用户无需关系

【注意】

无

AX_COLOR_GAMUT_E

【说明】

视频输出标准

【定义】

```
typedef enum axCOLOR_GAMUT_E
{
    AX_COLOR_GAMUT_BT601 = 0,
    AX_COLOR_GAMUT_BT709,
    AX_COLOR_GAMUT_BT2020,
    AX_COLOR_GAMUT_USER,
    AX_COLOR_GAMUT_BUTT
} AX_COLOR_GAMUT_E;
```

【成员】

成员名称	描述
AX_COLOR_GAMUT_BT601	BT601 色域
AX_COLOR_GAMUT_BT709	BT709 色域
AX_COLOR_GAMUT_BT2020	BT2020 色域
AX_COLOR_GAMUT_USER	用户自定义的色域, 非标准色域

【注意】

无

AX_ROTATION_E

【说明】

定义编码旋转角度。

【定义】

```
typedef enum
{
    AX_ROTATION_0    = 0,
    AX_ROTATION_90   = 1,
    AX_ROTATION_180  = 2,
    AX_ROTATION_270  = 3,
    AX_ROTATION_BUTT
} AX_ROTATION_E;
```

【成员】

成员名称	描述
AX_ROTATION_0	不旋转
AX_ROTATION_90	顺时针旋转 90 度
AX_ROTATION_180	顺时针旋转 180 度
AX_ROTATION_270	顺时针旋转 270 度

【注意】

无

AX_IMG_FORMAT_E

【说明】

像素格式定义。

【定义】

```
typedef enum
{
    AX_FORMAT_INVALID = -1,
    /* YUV400 8 bit */
    AX_FORMAT_YUV400 = 0x0,
    /* YUV420 8 bit */
    AX_FORMAT_YUV420_PLANAR = 0x1,
    AX_FORMAT_YUV420_PLANAR_VU = 0x2,
    AX_FORMAT_YUV420_SEMIPLANAR = 0x3,
    AX_FORMAT_YUV420_SEMIPLANAR_VU = 0x4,
    /* YUV422 8 bit */
    AX_FORMAT_YUV422_PLANAR = 0x8,
    AX_FORMAT_YUV422_PLANAR_VU = 0x9,
    AX_FORMAT_YUV422_SEMIPLANAR = 0xA,
    AX_FORMAT_YUV422_SEMIPLANAR_VU = 0xB,
    AX_FORMAT_YUV422_INTERLEAVED_YUVY = 0xC,
    AX_FORMAT_YUV422_INTERLEAVED_YUYV = 0xD,
    AX_FORMAT_YUV422_INTERLEAVED_UYVY = 0xE,
    AX_FORMAT_YUV422_INTERLEAVED_VYUY = 0xF,
    AX_FORMAT_YUV422_INTERLEAVED_VYYU = 0x10,
    /* YUV444 8 bit */
    AX_FORMAT_YUV444_PLANAR = 0x14,
    AX_FORMAT_YUV444_PLANAR_VU = 0x15,
    AX_FORMAT_YUV444_SEMIPLANAR = 0x16,
    AX_FORMAT_YUV444_SEMIPLANAR_VU = 0x17,
    AX_FORMAT_YUV444_PACKED = 0x18,
    /* YUV 10 bit */
    AX_FORMAT_YUV400_10BIT = 0x20,
    AX_FORMAT_YUV420_PLANAR_10BIT_UV_PACKED_4Y5B = 0x24,
```

```
AX_FORMAT_YUV420_PLANAR_10BIT_I010          = 0x25,  
AX_FORMAT_YUV420_SEMIPLANAR_10BIT_P101010    = 0x28,  
AX_FORMAT_YUV420_SEMIPLANAR_10BIT_P010        = 0x2A,  
AX_FORMAT_YUV420_SEMIPLANAR_10BIT_P016        = 0x2C,  
AX_FORMAT_YUV420_SEMIPLANAR_10BIT_I016        = 0x2E,  
AX_FORMAT_YUV420_SEMIPLANAR_10BIT_12P16B      = 0x2F,  
AX_FORMAT_YUV444_PACKED_10BIT_P010            = 0x30,  
AX_FORMAT_YUV444_PACKED_10BIT_P101010          = 0x32,  
AX_FORMAT_YUV422_SEMIPLANAR_10BIT_P101010      = 0x33,  
AX_FORMAT_YUV422_SEMIPLANAR_10BIT_P010          = 0x34,  
  
/* BAYER RAW */  
AX_FORMAT_BAYER_RAW_8BPP                      = 0x80,  
AX_FORMAT_BAYER_RAW_10BPP                     = 0x81,  
AX_FORMAT_BAYER_RAW_12BPP                     = 0x82,  
AX_FORMAT_BAYER_RAW_14BPP                     = 0x83,  
AX_FORMAT_BAYER_RAW_16BPP                     = 0x84,  
AX_FORMAT_BAYER_RAW_10BPP_PACKED              = 0x85,  
AX_FORMAT_BAYER_RAW_12BPP_PACKED              = 0x86,  
AX_FORMAT_BAYER_RAW_14BPP_PACKED              = 0x87,  
  
/* RGB Format */  
AX_FORMAT_RGB565                            = 0xA0,  
AX_FORMAT_RGB888                            = 0xA1,  
AX_FORMAT_KRGB444                           = 0xA2,  
AX_FORMAT_KRGB555                           = 0xA3,  
AX_FORMAT_KRGB888                           = 0xA4,  
AX_FORMAT_BGR888                            = 0xA5,  
AX_FORMAT_BGR565                            = 0xA6,  
  
AX_FORMAT_ARGB444                           = 0xC5,  
AX_FORMAT_ARGB1555                          = 0xC6,  
AX_FORMAT_ARGB8888                         = 0xC7,  
AX_FORMAT_ARGB8565                          = 0xC8,  
AX_FORMAT_RGBA8888                         = 0xC9,  
AX_FORMAT_RGBA5551                          = 0xCA,  
AX_FORMAT_RGBA4444                          = 0xCB,  
AX_FORMAT_RGBA5658                          = 0xCC,  
AX_FORMAT_ABGR4444                          = 0xCD,
```

```

AX_FORMAT_ABGR1555          = 0xCE,
AX_FORMAT_ABGR8888          = 0xCF,
AX_FORMAT_ABGR8565          = 0xD0,
AX_FORMAT_BGRA8888          = 0xD1,
AX_FORMAT_BGRA5551          = 0xD2,
AX_FORMAT_BGRA4444          = 0xD3,
AX_FORMAT_BGRA5658          = 0xD4,
AX_FORMAT_BITMAP             = 0xE0,
AX_FORMAT_MAX                = 0xE1

} AX_IMG_FORMAT_E;

```

【成员】

成员名称	描述
AX_FORMAT_YUV400	Y...
AX_FORMAT_YUV420_PLANAR	YYYY... UUUU ... VVVV... I420/YU12
AX_FORMAT_YUV420_PLANAR_VU	YYYY... VVVV... UUUU... YV12
AX_FORMAT_YUV420_SEMIPLANAR	YYYY... UVUVUV... NV12
AX_FORMAT_YUV420_SEMIPLANAR_VU	YYYY... VUVUVU... NV21
AX_FORMAT_YUV422_PLANAR	YYYY... UUUU... VVVV... I422
AX_FORMAT_YUV422_PLANAR_VU	YYYY... VVVV... UUUU... YV16
AX_FORMAT_YUV422_SEMIPLANAR	YYYY... UVUVUV... NV61
AX_FORMAT_YUV422_SEMIPLANAR_VU	YYYY... VUVUVU... NV16
AX_FORMAT_YUV422_INTERLEAVED_YUVY	YUVYYUVY... YUVY
AX_FORMAT_YUV422_INTERLEAVED_YUYV	YUYVYUYV... YUYV
AX_FORMAT_YUV422_INTERLEAVED_UYVY	UYVYUYVY... UYVY
AX_FORMAT_YUV422_INTERLEAVED_VYUY	VYUYVYUY... VYUY
AX_FORMAT_YUV422_INTERLEAVED_VYVU	VYUYVYUY... VYVU
AX_FORMAT_YUV444_PLANAR	YYYY... UUUU... VVVV... I444

成员名称	描述
AX_FORMAT_YUV444_PLANAR_VU	YYYY... VVVV... UUUU... YV24
AX_FORMAT_YUV444_SEMIPLANAR	YYYY... UVUVUV... NV24
AX_FORMAT_YUV444_SEMIPLANAR_VU	YYYY... VUVUVU... NV42
AX_FORMAT_YUV444_PACKED	YUV YUV YUV ...
AX_FORMAT_YUV400_10BIT	
AX_FORMAT_YUV420_PLANAR_10BIT_UV_PACKED_4Y5B	YYYY... UUUU... VVVV..., 4 Y pixels in 5 bytes, UV packed
AX_FORMAT_YUV420_PLANAR_10BIT_I010	16 bit pixel, low 10bits valid, high 6 bits invalid
AX_FORMAT_YUV420_SEMIPLANAR_10BIT_P101010	YYYY... UVUVUV..., Y/U/V 4 pixels in 5 bytes
AX_FORMAT_YUV420_SEMIPLANAR_10BIT_P010	16 bit pixel, high 10bits valid, low 6 bits invalid
AX_FORMAT_YUV420_SEMIPLANAR_10BIT_P016	16 bit pixel, low 10bits valid, high 6 bits invalid
AX_FORMAT_YUV420_SEMIPLANAR_10BIT_I016	16 bit pixel, high 10bits valid, low 6 bits invalid
AX_FORMAT_YUV420_SEMIPLANAR_10BIT_12P16B	12 pixels in 16bytes, low 120bits valid, high 8 bits invalid
AX_FORMAT_YUV444_PACKED_10BIT_P010	YUV YUV YUV ..., 16 bit pixel, high 10bits valid, low 6 bits invalid
AX_FORMAT_YUV444_PACKED_10BIT_P101010	YUV YUV YUV ..., 4 pixels storage in 5 bytes
AX_FORMAT_YUV422_SEMIPLANAR_10BIT_P101010	YYYY... UVUVUV..., Y/U/V 4 pixels in 5 bytes

成员名称	描述
AX_FORMAT_YUV422_SEMIPLANAR_10BIT_P010	16 bit pixel, high 10bits valid, low 6 bits invalid
AX_FORMAT_RGB565	RGB565 16bpp
AX_FORMAT_RGB888	RGB888 24bpp
AX_FORMAT_BGR888	RGBRGRB..., BGR888 32bpp
AX_FORMAT_BGR565	RGBRGRB..., BGR565 16bpp
AX_FORMAT_ARGB4444	ARGB4444 16bpp
AX_FORMAT_ARGB1555	ARGB1555 16bpp
AX_FORMAT_ARGB8888	ARGB8888 32bpp
AX_FORMAT_ARGB8565	ARGB8565 24bpp
AX_FORMAT_RGBA8888	ABGRABGR..., RGBA8888 32bpp
AX_FORMAT_RGBA5551	ABGRABGR..., RGBA5551 16bpp
AX_FORMAT_RGBA4444	ABGRABGR..., RGBA4444 16bpp
AX_FORMAT_RGBA5658	ABGRABGR..., RGBA5658 24bpp
AX_FORMAT_ABGR4444	RGBARGBA..., ABGR4444 16bpp
AX_FORMAT_ABGR1555	RGBARGBA..., ABGR1555 16bpp
AX_FORMAT_ABGR8888	RGBARGBA..., ABGR8888 32bpp
AX_FORMAT_ABGR8565	RGBARGBA..., ABGR8565 24bpp
AX_FORMAT_BGRA8888	ARGBARGB..., BGRA8888 32bpp
AX_FORMAT_BGRA5551	ARGBARGB..., BGRA5551 16bpp
AX_FORMAT_BGRA4444	ARGBARGB..., BGRA4444 16bpp
AX_FORMAT_BGRA5658	ARGBARGB..., BGRA5658 24bpp
AX_FORMAT_BITMAP	每个像素占用 1bit 1 bpp

【注意】

无

AX_VIN_IVPS_MODE_E

【说明】

定义 VIN 各个 PIPE 和 IVPS 各个组的工作模式。

【定义】

```
typedef enum
{
    AX_ITP_OFFLINE_VPP = 0,
    AX_GDC_ONLINE_VPP = 1,
    AX_ITP_ONLINE_VPP = 2,
    AX_VIN_IVPS_MODE_BUTT
} AX_VIN_IVPS_MODE_E;
```

【成员】

成员名称	描述
AX_ITP_OFFLINE_VPP	VIN(ITP)与 IVPS(VPP)以离线模式连接
AX_GDC_ONLINE_VPP	VIN(GDC)与 IVPS(VPP)以在线模式连接
AX_ITP_ONLINE_VPP	VIN(ITP)与 IVPS(VPP)以在线模式连接

【注意】

具体参数《22 - AX IVPS API 文档》文档中 [1.4 硬件单元](#)。

AX_VIDEO_FRAME_T

【说明】

定义图像属性。

【定义】

```
typedef struct axVIDEO_FRAME_T
{
    AX_U32                      u32Width;
    AX_U32                      u32Height;
    AX_IMG_FORMAT_E              enImgFormat;
    AX_VSCAN_FORMAT_E            enVscanFormat;
    AX_FRAME_COMPRESS_INFO_T    stCompressInfo;
    AX_DYNAMIC_RANGE_E           stDynamicRange;
    AX_COLOR_GAMUT_E              stColorGamut;

    AX_U32                      u32PicStride[AX_MAX_COLOR_COMPONENT];
    AX_U32                      u32ExtStride[AX_MAX_COLOR_COMPONENT];

    AX_U64                      u64PhyAddr[AX_MAX_COLOR_COMPONENT];
    AX_U64                      u64VirAddr[AX_MAX_COLOR_COMPONENT];
    AX_U64                      u64ExtPhyAddr[AX_MAX_COLOR_COMPONENT];
    AX_U64                      u64ExtVirAddr[AX_MAX_COLOR_COMPONENT];

    AX_U32                      u32HeaderSize[AX_MAX_COLOR_COMPONENT];
    AX_U32                      u32BlkId[AX_MAX_COLOR_COMPONENT];
    AX_S16                       s16CropX;
    AX_S16                       s16CropY;
    AX_S16                       s16CropWidth;
    AX_S16                       s16CropHeight;
    AX_U32                      u32TimeRef;
    AX_U64                      u64PTS;
    AX_U64                      u64SeqNum;
    AX_U64                      u64UserData;
    AX_U64                      u64PrivateData;
    AX_U32                      u32FrameFlag;
    AX_U32                      u32FrameSize;
} AX_VIDEO_FRAME_T;
```

【成员】

成员名称	描述
u32Width	图像宽度
u32Height	图像高度
enImgFormat	图像像素格式
enVscanFormat	视频扫描模式
stCompressInfo	FBC 压缩模式
stDynamicRange	动态范围
stColorGamut	色域范围
u32PicStride	图像数据横方向跨度
u32ExtStride	10bit 数据位宽的图象,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.
u64PhyAddr	图像数据物理地址
u64VirAddr	图像数据虚拟地址
u64ExtPhyAddr	10bit 数据位宽的图象,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	10bit 数据位宽的图象,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.
u32HeaderSize	帧头大小
u32BlkId	Block id, u32BlkId[0]的 metadata 存放帧结构体
s16CropX	有效图像的顶部相对原点竖坐标
s16CropY	有效图像的底部相对原点竖坐标
s16CropWidth	有效图像的宽度
s16CropHeight	有效图像的高度
u32TimeRef	图象帧序列号
u64PTS	图像时间戳

成员名称	描述
u64SeqNum	图像帧序列号
u64UserData	用户私有数据
u64PrivateData	SDK 内部私有数据, 保留
u32FrameFlag	SDK 内部私有数据, 保留
u32FrameSize	帧数据所占用存储空间大小

【注意】

无

AX_MOD_ID_E

【说明】

定义模块 ID。

【定义】

```
typedef enum
{
    AX_ID_MIN      = 0x00,
    AX_ID_ISP      = 0x01,
    AX_ID_CE       = 0x02,
    AX_ID_VO       = 0x03,
    AX_ID_VDSP     = 0x04,
    AX_ID_EFUSE    = 0x05,
    AX_ID_NPU      = 0x06,
    AX_ID_VENC     = 0x07,
    AX_ID_VDEC     = 0x08,
    AX_ID_JENC     = 0x09,
    AX_ID_JDEC     = 0x0a,
    AX_ID_SYS      = 0x0b,
    AX_ID_AENC     = 0x0c,
    AX_ID_IVPS     = 0x0d,
    AX_ID_MIPI     = 0x0e,
    AX_ID_ADEC     = 0x0f,
    AX_ID_DMA      = 0x10,
    AX_ID_VIN      = 0x11,
    AX_ID_USER     = 0x12,
    AX_IDIVES     = 0x13,
    AX_ID_SKEL     = 0x14,
    AX_ID_IVE      = 0x15,
    AX_ID_AVSCALI  = 0x17,
    AX_ID_AUDIO    = 0x1a,
    AX_ID_ALGO     = 0x1b,
    AX_ID_ENGINE   = 0x1c,
    AX_ID_ACODEC   = 0x1f,
    AX_ID_AI       = 0x20,
    AX_ID_AO       = 0x21,
```

```
AX_ID_SENSOR      = 0x22,  
AX_ID_NT          = 0x23,  
AX_ID_TDP         = 0X24,  
AX_ID_VPP         = 0X25,  
AX_ID_GDC         = 0x27,  
AX_ID_BASE        = 0x28,  
AX_ID_THERMAL     = 0x29,  
AX_ID_3A_AE        = 0x2a,  
AX_ID_3A_AWB       = 0x2b,  
AX_ID_3A_AF        = 0x2c,  
AX_ID_AXGZIPD     = 0x2d,  
AX_ID_RESERVE      = 0x2e,  
AX_ID_BUTT         =  
AX_ID_CUST_MIN    = 0x80,  
AX_ID_MAX          = 0xFF  
} AX_MOD_ID_E;
```

【注意】

无

AX_LINK_MODE_E

【说明】

LINK 模式。

【定义】

```
typedef enum
{
    AX_UNLINK_MODE = 0,
    AX_LINK_MODE = 1,
} AX_LINK_MODE_E;
```

【成员】

成员名称	描述
AX_UNLINK_MODE	非 link 模式
AX_LINK_MODE	link 模式

【注意】

无

AX_VIDEO_FRAME_INFO_T

【说明】

图像属性信息。

【定义】

```
typedef struct axVIDEO_FRAME_INFO_T {  
    AX_VIDEO_FRAME_T     stVFrame;  
    AX_MOD_ID_E          enModId;  
    AX_BOOL               bEndOfStream;  
} AX_VIDEO_FRAME_INFO_T;
```

【成员】

成员名称	描述
stVFrame	图像属性。
enModId	模块 ID。
bEndOfStream	是否发完全部码流

【注意】

无

AX_NOTIFY_EVENT_E

【说明】

NOTIFY 事件类型定义

【定义】

```
typedef enum {  
    AX_NOTIFY_EVENT_SLEEP    = 0,  
    AX_NOTIFY_EVENT_WAKEUP   = 1,  
    AX_NOTIFY_EVENT_MAX  
} AX_NOTIFY_EVENT_E;
```

【成员】

成员名称	描述
AX_NOTIFY_EVENT_SLEEP	睡眠
AX_NOTIFY_EVENT_WAKEUP	唤醒

【注意】

无

AX_SYS_CLK_ID_E

【说明】

定义 CLK 模块 ID

【定义】

```
typedef enum {
    AX_CPU_CLK_ID      = 0,
    AX_BUS_CLK_ID      = 1,
    AX_NPU_CLK_ID      = 2,
    AX_ISP_CLK_ID      = 3,
    AX_MM_CLK_ID       = 4,
    AX_VPU_CLK_ID      = 5,
    AX_SYS_CLK_MAX_ID  = 6,
} AX_SYS_CLK_ID_E
```

【注意】

无

AX_MOD_INFO_T

【说明】

定义模块组通道结构体。

【定义】

```
typedef struct axMOD_INFO_T {  
    AX_MOD_ID_E enModId;  
    AX_S32      s32GrpId;  
    AX_S32      s32ChnId;  
} AX_MOD_INFO_T;
```

【成员】

成员名称	描述
enModId	模块号。
s32GrpId	组号。
s32ChnId	通道号。

【注意】

无

AX_LINK_DEST_T

【说明】

定义数据源绑定的所有数据接收者信息集合。

【定义】

```
typedef struct axLINK_DEST_S{  
    AX_U32 u32DestNum;  
    AX_MOD_INFO_T astDestMod[AX_LINK_DEST_MAXNUM];  
} AX_LINK_DEST_T;
```

【成员】

成员名称	描述
u32DestNum	绑定的数据接收者个数。
astDestMod	数据接收者模块-组-通道信息。

【注意】

无

AX_MEMORY_SOURCE_E

【说明】

输出 buffer 的内存分配模式

【定义】

```
typedef enum {  
    AX_MEMORY_SOURCE_CMM = 0,  
    AX_MEMORY_SOURCE_POOL = 1,  
    AX_MEMORY_SOURCE_OS = 2,  
    AX_MEMORY_SOURCE_BUTT,  
} AX_MEMORY_SOURCE_E;
```

【成员】

成员名称	描述
AX_MEMORY_SOURCE_CMM	CMM 内存分配方式
AX_MEMORY_SOURCE_POOL	POOL 内存分配方式
AX_MEMORY_SOURCE_OS	OS 内存分配方式

【注意】

无

AX_MEMORY_ADDR_T

【说明】

输出 buffer 地址

【定义】

```
typedef struct {  
    AX_U64 u64PhyAddr;  
    AX_VOID *pVirAddr;  
} AX_MEMORY_ADDR_T;
```

【成员】

成员名称	描述
u64PhyAddr	物理地址
pVirAddr	虚拟地址

【注意】

无

AX OSD BMP ATTR T

【说明】

定义 OSD 图像属性。

【定义】

```
typedef struct axOSD_BMP_ATTR_T {  
    AX_U16 u16Alpha;  
    AX_IMG_FORMAT_E enRgbFormat;  
    AX_U8 *pBitmap;  
    AX_U64 u64PhyAddr;  
    AX_U32 u32BmpWidth;  
    AX_U32 u32BmpHeight;  
    AX_U32 u32DstXoffset;  
    AX_U32 u32DstYoffset;  
    AX_U32 u32Color;  
    AX_BOOL bColorInv;  
    AX_U32 u32ColorInv;  
    AX_U32 u32ColorInvThr;  
} AX OSD BMP ATTR T;
```

【成员】

成员名称	描述
u16Alpha	OSD 区域透明度等级，范围(0, 255]。其中 0: 透明；255: 不透明。仅 ARGB、RGBA、ABGR、BGRA 格式生效
enRgbFormat	OSD 区域格式
pBitmap	OSD 区域起始虚拟地址
u64PhyAddr	OSD 区域起始物理地址
u32BmpWidth	OSD 区域宽度
u32BmpHeight	OSD 区域高度
u32DstXoffset	OSD 区域相对于原图的横坐标 Xoffset
u32DstYoffset	OSD 区域相对于原图的纵坐标 Yoffset

成员名称	描述
u32Color	当前 OSD 颜色，仅使用于 bitmap-1 格式
bColorInv	OSD 反色是否使能，仅使用于 bitmap-1 格式
u32ColorInv	反色后的颜色，仅使用于 bitmap-1 格式
u32ColorInvThr	反色的阈值，仅使用于 bitmap-1 格式

【注意】

- u16Alpha 表示 region 通道属性的透明度，可以和 ARGB 格式的 alpha 值同时生效。

AX_ERR_CODE_E

【说明】

错误类型定义。

【定义】

```
typedef enum {  
    AX_ERR_INVALID_MODID      = 0x01,  
    AX_ERR_INVALID_DEVID       = 0x02,  
    AX_ERR_INVALID_GRPID       = 0x03,  
    AX_ERR_INVALID_CHNID       = 0x04,  
    AX_ERR_INVALID_PIPEID       = 0x05,  
    AX_ERR_INVALID_STITCHGRPID = 0x06,  
    AX_ERR_ILLEGAL_PARAM        = 0x0A,  
    AX_ERR_NULL_PTR            = 0x0B,  
    AX_ERR_BAD_ADDR             = 0x0C,  
    AX_ERR_SYS_NOTREADY        = 0x10,  
    AX_ERR_BUSY                 = 0x11,  
    AX_ERR_NOT_INIT              = 0x12,  
    AX_ERR_NOT_CONFIG            = 0x13,  
    AX_ERR_NOT_SUPPORT           = 0x14,  
    AX_ERR_NOT_PERM              = 0x15,  
    AX_ERR_EXIST                 = 0x16,  
    AX_ERR_UNEXIST               = 0x17,  
    AX_ERR_NOMEM                 = 0x18,  
    AX_ERR_NOBUF                  = 0x19,  
    AX_ERR_NOT_MATCH              = 0x1A,  
    AX_ERR_BUF_EMPTY              = 0x20,  
    AX_ERR_BUF_FULL                = 0x21,  
    AX_ERR_QUEUE_EMPTY             = 0x22,  
    AX_ERR_QUEUE_FULL              = 0x23,  
    AX_ERR_TIMED_OUT                = 0x27,  
    AX_ERR_FLOW_END                  = 0x28,  
    AX_ERR_UNKNOWN                  = 0x29,  
    AX_ERR_OS_FAIL                    = 0x30,  
    AX_ERR_BUTT                     = 0x7F,  
} AX_ERR_CODE_E;
```

AX_COORD_E

【说明】

定义坐标模式。

【定义】

```
typedef enum {  
    AX_COORD_ABS = 0,  
    AX_COORD_RATIO,  
    AX_COORD_BUTT  
} AX_COORD_E;
```

【成员】

成员名称	描述
AX_COORD_ABS	绝对坐标
AX_COORD_RATIO	相对坐标

【注意】

- 相对坐标，即坐标值是以当前图像宽高的比率来表示，具体比率转换与实际使用接口为准。

AX_BGCOLOR_T

【说明】

定义背景色的属性

【定义】

```
typedef struct {  
    AX_BOOL    bEnable;  
    AX_U32     nBgColor;  
} AX_BGCOLOR_T;
```

【成员】

成员名称	描述
bEnable	背景色的使能标志位
nBgColor	背景色的颜色属性，RGB888，范围[0, 0xffff]

【注意】

无

AX_COLORKEY_T

【说明】

定义 Color Key 功能，只有 TDP 的 region layer0 支持此功能。

【定义】

```
typedef struct {  
    AX_BOOL bEnable;  
    AX_BOOL bInv;  
    AX_U32 nKeyLow;  
    AX_U32 nKeyHigh;  
} AX_COLORKEY_T;
```

【成员】

成员名称	描述
bEnable	Color key 使能位
nInv	AX_TRUE:阈值范围内的透明 AX_FALSE:阈值范围内外透明
mKeyLow	Color key 颜色阈值的下限，RGB888 格式，表示为：RRGGBB
nKeyHigh	Color key 颜色阈值的上限，RGB888 格式，表示为：RRGGBB

【注意】

无

AX_BITCOLOR_T

【说明】

定义 Bitmap 格式的二进制位图颜色及反色功能。

【定义】

```
typedef struct {  
    AX_U32    nColor;  
    AX_BOOL   bColorInv;  
    AX_U32    nColorInv;  
    AX_U32    nColorInvThr;  
} AX_BITCOLOR_T;
```

【成员】

成员名称	描述
nColor	二进制位图当前颜色，RGB888 格式，表示为：0xRRGGBB
bColorInvEn	反色使能位
nColorInv	二进制位图反色后的颜色，RGB888 格式，表示为：0xRRGGBB
nColorInvThr	反色阈值，RGB888 格式，表示为：0xRRGGBB； 若位图中和背景色差值在阈值内的区域，此区域颜色会变反色

【注意】

无

AX_OVERLAY_T

【说明】

OVERLAY 结构体定义。

【定义】

```
typedef struct {  
    AX_BOOL          bEnable;  
    AX_U16           nWidth;  
    AX_U16           nHeight;  
    AX_U32           nStride;  
    AX_IMG_FORMAT_E eFormat;  
    AX_U64           u64PhyAddr[2];  
    AX_U8            nAlpha;  
    AX_POINT_T       tOffset;  
    AX_COLORKEY_T    tColorKey;  
    AX_BITCOLOR_T    tBitColor;  
} AX_OVERLAY_T;
```

【成员】

成员名称	描述
bEnable	Overlay 的使能开关
nWidth、nHeight、nStride	宽、高、跨度
eFormat	格式
u64PhyAddr[2]	物理地址信息
nAlpha	透明度, [0-255]
tOffset	叠加的偏移值
tColorKey	ColorKey 参数
tBitColor	BitColor 参数

【注意】

无

AX_CSC_MATRIX_TYPE_E

【说明】

图像格式转换。

【定义】

```
typedef enum {  
    AX_CSC_MATRIX_BT601 = 0,      /* full range */  
    AX_CSC_MATRIX_BT709,        /* full range */  
    AX_CSC_MATRIX_BT2020,        /* full range */  
    AX_CSC_MATRIX_BT601_TV,     /* limit range */  
    AX_CSC_MATRIX_BT709_TV,     /* limit range */  
    AX_CSC_MATRIX_USER,  
    AX_CSC_MATRIX_TYPE_BUTT  
} AX_CSC_MATRIX_TYPE_E;
```

【注意】

无

AX_CSC_MATRIX_T

【说明】

图像格式转换结构体。

【定义】

```
typedef struct {  
    AX_CSC_MATRIX_TYPE_E nType;  
    AX_U16 nMatrix[9];  
    AX_U16 nInOffset[3];  
    AX_U16 nOutOffset[3];  
} AX_CSC_MATRIX_T;
```

【成员】

成员名称	描述
nType	转换矩阵类型
nMatrix	用户自定义的 3x3 转换矩阵参数，行列式。 小数定点化处理：直接乘 256。
nInOffset	用户自定义的偏移量，先做此偏移再做矩阵转换。yuv 格式的话 分别对应 y u v 三个分量的偏移：Offset0-y Offset1-u Offset2-v
nOutOffset	用户自定义的偏移量，矩阵转换后再做此偏移。yuv 格式的话分 别对应 y u v 三个分量的偏移：Offset0-y Offset1-u Offset2-v

【注意】

- nMatrix 3x3 行列式取值为 **【-1024,1023】**，1.0 对应取值 256。负数可以强制类型转换为 AX_U16，
- nInOffset 和 nOutOffset 取值范围为 **【-128, 127】**，负数可以强制类型转换为 AX_U16。

AX_MAX_COLOR_COMPONENT

【说明】

定义颜色最大分量的个数

【定义】

```
#define AX_MAX_COLOR_COMPONENT 3
```

AX_VSCAN_FORMAT_E

【说明】

视频扫描模式

【定义】

```
typedef enum {  
    AX_VSCAN_FORMAT_RASTER = 0,  
    AX_VSCAN_FORMAT_BUTT  
} AX_VSCAN_FORMAT_E;
```

【成员】

成员名称	描述
AX_VSCAN_FORMAT_RASTER	视频光栅扫描模式

【注意】

无

AX_GDC_MODE_E

【说明】

GDC 模式

【定义】

```
typedef enum _AX_GDC_MODE_E_ {  
    AX_GDC_MODE_CORE0      = 0,  
    AX_GDC_MODE_MAX  
} AX_GDC_MODE_E;
```

【成员】

成员名称	描述
AX_GDC_MODE_CORE0	GDC 核 0

【注意】

无

AX_VDSP_MODE_E

【说明】

VDSP 模式

【定义】

```
typedef enum _AX_VDPS_MODE_E_ {  
    AX_VDSP_MODE_CORE0      = 0,  
    AX_VDSP_MODE_CORE1      = 1,  
    AX_VDSP_MODE_AUTO       = 2,  
    AX_VDSP_MODE_MAX  
} AX_VDSP_MODE_E;
```

【成员】

成员名称	描述
AX_VDSP_MODE_CORE0	VDSP 核 0
AX_VDSP_MODE_CORE1	VDSP 核 1
AX_VDSP_MODE_AUTO	自动选择 VDSP 核

【注意】

AX_WARP_ENGINE_E

【说明】

WARP 引擎

【定义】

```
typedef enum _AX_WARP_ENGINE_E_ {  
    AX_WARP_ENGINE_GDC      = 0,  
    AX_WARP_ENGINE_VDSP     = 1,  
    AX_WARP_ENGINE_MAX  
} AX_WARP_ENGINE_E;
```

【成员】

成员名称	描述
AX_WARP_ENGINE_GDC	使用 GDC
AX_WARP_ENGINE_VDSP	使用 VDSP

【注意】

- 不支持 VDSP

AX_WARP_MODE_U

【说明】

WARP 模式

【定义】

```
typedef union _AX_WARP_MODE_U_ {  
    AX_GDC_MODE_E eGdcMode;  
    AX_VDSP_MODE_E eVdspMode;  
} AX_WARP_MODE_U;
```

【成员】

成员名称	描述
eGdcMode	GDC 模式参数
eVdspMode	VDSP 模式参数

【注意】

无