

Hall the state of the state of

MPP 媒体处理软件 开发参考

Will Think the the training of the state of

THE REPORT OF THE PARTY OF THE

RAND linder

た 版本号: 1.2 8布日期・2019 11:1

1/-1/1



Will Little White High State of the State of

### 版本历史

A	LLWIMER ST		版本历史	文档密级	
	版本号	日期	制/修订人	内容描述	]
	1.0	2017.08.04	eric_wang	初版,基于 V5 芯片。	]
	1.1	2019.03.31	eric_wang	修订 V316 相关内容	
	1.2	2019.11.18	eric_wang	增加 V833 相关内容	

And the state of t Will Think the think of the state of the sta 操海洋提供推拔推拔 操海上提開推構推構推開 Will Little White High State of the State of

> THE BEAT HER BEAT HER LEGISLATION OF THE PARTY OF THE PAR 版权所有 © 珠海全志科技股份有限公司。保留一切权利



#### 录 目

ALLWIME	aleri Bilideri	Blinderi	文档密级、秘密
THE TOTAL PROPERTY OF THE PARTY		录	R.VV
A STATE OF THE STA	_		A A A A A A A A A A A A A A A A A A A
1 前言	A CONTRACTOR OF THE CONTRACTOR	Hilling.	1
-150	既述	· 🎢 · · · · · · · · · · · · · · · · · ·	1
	产品版本 ॐ`	·*., · · · · · · · · · · · · · · · · · · ·	.*
	读者对象		1
1.4			1
-	1.4.1 符号约定		1
2 系统技	空制		3
	既述		3
	功能描述		3
	2.2.1 状态		
11-	2.2.2 系统绑定		
. X(x)			4
√3 <del>*</del> €^	13K/		
4>-4	2.2.5 通过 mpp 的 proc 节点实时查	45-4	
-100	API 参考	-100	7
	2.3.1 AW MPI_SYS_Init		
	2.3.2 AW_MPI_SYS_Exit		9
	2.3.4 AW_MPI_SYS_GetConf		9
	2.3.5 AW MPI SYS Bind	M ?	
	2.3.6 AW MPI SYS UnBind	<b>V</b>	
	2.3.7 AW_MPI_SYS_GetBindbyDe		
	2.3.8 AW_MPI_SYS_GetVersion		
,	2.3.9 AW MPI SYS GetCurPts.		<u>1</u> 3 <sup>(1)</sup>
	2.3.10 AW MPI SYS InitPtsBase		
	2.3.11 AW MPI SYS SyncPts .		
N. K.	2.3.12 AW_MPI_SYS_MmzAlloc_0	Cached	15
	2.3.13 AW_MPI_SYS_MmzFree		
	2.3.14 AW MPI_SYS_MmzFlushC		
	2.3.15 AW_MPI_SYS_GetVirMem		
	2.3.16 AW_MPI_SYS_HANDLE_Z		
	2.3.17 AW_MPI_SYS_HANDLE_S		
	2.3.18 AW_MPI_SYS_HANDLE_IS		
	2.3.19 AW_MPI_SYS_HANDLE_S		
	数据类型		
•	2.4.1 视频公共类型		
	2.4.1.1 VIDEO_FRAME_S 2.4.1.2 VIDEO_FRAME_IN	FO S	
	2.4.1.2 VIDEO_FRAME_IN 2.4.1.3 BITMAP_S		
AND THE PERSON OF THE PERSON O	2.4.1.3 BITMAT 3		22
A LINE WAY			Hall St.
- TEN -	版权所有 © 珠海全志科	支股份有限公司。保留一切权利	ii
× <sub>y</sub> ۱۰	*yi*	-yi	×yl*

그 사내 중 (표 🛆	X North
文档密级公	ANN ANN

X AND THE VIEW OF THE PARTY OF	2.4.2.1 MPPCallbackInfo		
2.5	错误码		23
3 视频	输入	7-Je	24
3.1	概述 ※	· · · · · · · · · · · · ·	24
3.2	功能框图		24
3.3	VIPP Buffer 管理和使用		25
	3.3.1 ViPP 非绑定情况下		25
	3.3.2 ViPP 绑定情况下		26
3.4	API 和状态图		26
3.5	API 接口		27
	$3.5.1\ AW\_MPI\_VI\_CreateVipp\ .\ .\ .$	· ·	
	3.5.2 AW_MPI_VI_DestoryVipp		29
117	3.5.3 AW_MPI_VI_SetVippAttr		
A STATE OF THE STA	3.5.4 AW MPI VI GetVippAttr		
N. K.	3.5.5 AW MPI VI SetVIFreq		,&X 32
H. H	3.5.6 AW_MPI_VI_EnableVipp		
THE TOTAL PROPERTY OF THE PARTY	3.5.7 AW MPI_VI_DisableVipp	,	34
×,1	3.5.8 AW_MPI_VI_SetVippFlip 3.5.9 AW_MPI_VI_GetVippFlip		35
	3.5.9 AW_MPI_VI_GetVippFlip		
	3.5.10 AW_MPI_VI_SetVippMirror.		36
	3.5.11 AW_MPI_VI_GetVippMirror		
	$3.5.12~AW\_MPI\_VI\_CreateVirChn$ .		
	3.5.13 AW_MPI_VI_DestoryVirChn		
	3.5.14 AW_MPI_VI_SetVirChnAttr .		40
	3.5.15 AW_MPI_VI_GetVirChnAttr. 3.5.16 AW_MPI_VI_EnableVirChn. 3.5.17 AW_MPI_VI_DisableVirChn. 3.5.18 AW_MPI_VI_GetFrame 3.5.19 AW_MPI_VI_ReleaseFrame. 3.5.20 AW_MPI_VI_SetVIFreq		41
,	3.5.16 AW_MPI_VI_EnableVirChn .		
W. T.	3.5.17 AW_MPI_VI_DisableVirChn.		
	3.5.18 AW_MPI_VI_GetFrame		43
W. W	3.5.19 AW_MPLVI_ReleaseFrame.		44
	3.5.20 AW_MP1_VI_SetVIFreq		45
**************************************	3.5.21 AW MPI_VI_RegisterCallback		321
3.6	数据结构		
	3.6.1 VI_ATTR_S		
	3.6.2 RGN_ATTR_S		
	3.6.4 BITMAP_S		
	3.6.5 osd_fmt		
	3.6.6 VIDEO_FRAME_INFO_S		
3 7			61
5.7	index in the second sec	inden	iinde
4 视频	错误码	AID.	63
4.1	概述		63
a Military	A WAR	A MILE XXX	O MARKET NAT
		1. The state of th	
(英)	版权所有 © 珠海全志科技股份	有限公司。保留一切权利	iii
7'	2		7-



ALLWII	WER	Bline	Bline	文档密级:秘密
4.2	411	文档目的	A CONTRACTOR OF THE PARTY OF TH	63
XXXXX	4.1.2	VO 简介		63
Light.	4.1.3	VO 简介		63
4.2	图层			64
**	4.2.1	图层操作说明	₹DE-1	
	4.2.2	显示输出设备操作说明		64
	4.2.3	图层 size 与 crop		65
	4.2.4	图层 crop 和 screen_win		65
	4.2.5	alpha		65
4.3	输出设	备介绍		65
4.4	模块状	态转换		65
4.5	API接	🗖		66
	4.5.1	AW_MPI_VO_Enable		
WIND WAR	4.5.2	AW_MPI_VO_Disable		67
	4.5.3	AW_MPI_VO_SetPubAttr		67
HALL TO THE PARTY OF THE PARTY	4.5.4	AW_MPI_VO_GetPubAttr	×	<u> </u>
H. H. L. H.	4.5.5	AW_MPI_VO_GetHdmiHwMode		70
K. W.	4.5.6	AW_MPI_VO_EnableVideoLayer AW_MPI_VO_DisableVideoLayer		71
	4.5.7	AW_MPI_VO_DisableVideoLayer		
		AW_MPI_VO_AddOutsideVideoLaye		
		AW_MPI_VO_RemoveOutsideVideoI		
		O AW_MPI_VO_OpenVideoLayer AW MPI VO CloseVideoLayer		
		2 AW_MPI_VO_SetVideoLayerAttr		
		B AW_MPI_VO_GetVideoLayerAttr .		
		AW_MPI_VO_SetVideoLayerPriorit		
		5 AW_MPI_VO_GetVideoLayerPriorit		
11V		6 AW_MPI_VO_SetVideoLayerAlpha		
A A A A A A A A A A A A A A A A A A A	4.5.1	7 AW_MPI_VO_GetVideoLayerAlpha		83
A PROPERTY.	4.5.18	B AW MPI VO EnableChn		84
	4.5.19	B AW_MPI_VO_EnableChn		
THE REAL PROPERTY OF THE PERSON OF THE PERSO	4.5.20	) AW_MPI_VO_RegisterCallback		86
	4.5.2	AW_MPI_VO_SetChnDispBufNum .		88
	4.5.22	2 AW_MPI_VO_GetChnDispBufNum		89
	4.5.23	$ m 8~AW\_MPI\_VO\_GetDisplaySize$		90
		l AW_MPI_VO_StartChn		
		5 AW_MPI_VO_StopChn		
		S AW_MPI_VO_PauseChn		
	4.5.2	7 AW_MPI_VO_ResumeChn		93
	4.5.28	3 AW_MPI_VO_Seek		
~ 11.7	<b>4</b> .5.29	AW_MPI_VO_Seek		
X. Market	4.5.30	O AW_MPI_VO ShowChn		96
LEXXY	4.5.3	AW MPI VO HideChn	XY	97

ALLWIMER		SetChnPts	Blinder	文档密级、秘密
4.5.3 4.5.3 4.5.3 4.6 数据组 4.6.1	32 AW MPI VO	SetChnPts	N. V.	98
4.5.3	33 AW MPI VO	SendFrame		99
4.5.3	34 AW MPI VO I	Debug_StoreFrame		100
4.6 数据组				101
4.6.1	VO_CHN_ATTF	 R_S		.*
		_S		
4.6.3	OVIDEO_LA	YER_ATTR_S		103
4.6.4	VO_VIDEO_LA	YER_ALPHA_S		104
	<del>_</del>	E_INFO_S		
		E_S		
		E		
4.6.8	3 VIDEO_FORM	AT_E		107
4.6.9	O COMPRESS_M	ODE_E	· · · · · · · · · · · · · · · · · · ·	
4.7 错误	冯	ODE_E		
5.1 概述				B 4 111
5.1.1	ISE 简介			111
5.1.2	· · · · · · · · · · · · · · · · · · ·	W. The		
5.1.3	B ISE 组件 buffer	管理和使用	.a.M.G	112
5.1.4	I ISE 组件典型应序	用场景	1. 1	113
5.2 状态	转换与 API 接口 .			114
5.2.1	状态图			114
5.2.2	2 API 接口	管理和使用		115
	5.2.2.1 AW MJ	PI_ISE_CreateGroup		115
Will I Hall Beet Hall Be To Thinderi	5.2.2.2 AW_MI	PI_ISE_DestroyGrou	p	116
dinder.	5.2.2.3 AW_M	PI_ISE_SetGrpAttr	· · · · · · · · · · · · · · · · · · ·	
	5.2.2.4 AW_MI	PI ISE GetGrpAttr	4/12	
A TOP TO THE PERSON OF THE PER	5.2.2.5 AW_M	PI_ISE_CreatePort		
and the state of t	5.2.2.6 AW_MI	PI_ISE_DestroyPort PI_ISE_GetPortAttr	£	
1 July 1	5.2.2.7 AW_M			
**************************************	5.2.2.8 AW_M	PI_ISE_SetPortAttr		121
	_	PI_ISE_Start		
	_	IPI_ISE_Stop		
		IPI_ISE_GetData .		
		IPI_ISE_ReleaseData		
		IPI_ISE_SendPic IPI_ISE_RegisterCal		
		IFI_ISE_RegisterCar IPI_ISE_SetISEFreq		
53 数据组	_	· · · · ;// · · · · · · · ·		
5.4 错误	·3.1.5 ·3.	sinden.	ninder.	
11V	.,	9.1V	WIN.	WIN.
6 视频编码	X4(1)	· Jahrager · · · · · · · · · · · · · · · · · · ·	NA TONING	144
a William	A STATE OF THE STA	, vič	KX <sup>X</sup>	AND THE PARTY OF T
			-	
- 18 1	版	双所有 © 珠海全志科技股份有限公司	司。保留一切权利	- (A)
,	,	7'		•

**144** 



				文档密级:秘密
6.1 根		N. V	A LIV	文档密级、秘密
6.2 년	九 九能描述	***************************************		144
6.2 %	2.1 缩放功能			145
4/2 6	.2.1 %		Start Fr	145
				145
				146
				146
				147
				147
				147
6	.2.9 输入数据压缩	模式省带宽		148
6.3 粉	犬态转换与 API 接□	1		148
.6	.3.1 状态图	iirdeni		
6	.3.2 API 和状态 .			
A 6	.3.3 API 参考	**************************************		
A PARTY AND A PART	6.3.3.1 AW	MPI VENC Cre	ateChn	
THE WAY	6.3.3.2 AW	MPI VENC Des	stroyChn	152
-1637	6.3.3.3 AW	 _MPI_VENC_Res	etChn	153
×yı	6.3.3.4 AW	MPI_VENC_Sta	rtRecvPic	
	6.3.3.5 AW	_MPI_VENC_Sta	rtRecvPicEx	155
	6.3.3.6 AW	MPI_VENC_Sto	pRecvPic	156
	6.3.3.7 AW	_MPI_VENC_Que	ery	157
	6.3.3.8 AW	_MPI_VENC_Reg	jisterCallback	157
				158
	6.3.3.10 AV			159
	6.3.3.11 AV	V_MPI_VENC_Ge	etStream	160
	6.3.3.12 AV			
A STATE OF THE STA	6.3.3.13 AV		VA *	
A STATE OF THE STA	6.3.3.14 AV			
操機工程制機及可能	6.3.3.15 AV	V_MPI_VENC_Ge	etHandle	
-1961 - 1	6.3.3,16 AV	V_MPI_VENC_Se	tRoiCig	165
₹K,	6.3%3.1 / AV			166
	0.5.5.10 Av			166
			<del>-</del>	167
				169
				170
				171
	63321 AV			172
	6.3.3.21 AV			172
	6.3.3.26 AV			
ARIV TO	6.3.3.27 AV	113	117	
A A A A A A A A A A A A A A A A A A A	6.3.3.28 AV	J16, — — —	J'K'	175
操機上提供推進			A STATE OF THE STA	A. William
-160	-16h	版权所有 © 珠海全志科技		vi
K.V.	T.K.T.	WILLIAM CONTRACTOR		***·



X KINGS	6.3.3.29 AW_MPI_VENC_SetTimeLapse	°. 176
	6.3.3.30 AW_MPI_VENC_GetTimeLapse	. 177
17-17 T	6.3.3.31 AW_MPI_VENC_SetColor2Grey	. 178
A THE PARTY OF THE	6.3.3.32 AW_MPI_VENC_GetColor2Grey	. 178
K,	6.3.3.33 AW_MPI_VENC_SetCrop	. 179
	6.3.3.34 AW_MPI_VENC_GetCrop	
	6.3.3.35 AW_MPI_VENC_SetIntraRefresh	. 181
	6.3.3.36 AW MPI VENC GetIntraRefresh	
	6.3.3.37 AW_MPI_VENC_SetSmartP	. 183
	6.3.3.38 AW_MPI_VENC_GetSmartP	. 183
	6.3.3.39 AW_MPI_VENC_SetBrightness	
	6.3.3.40 AW_MPI_VENC_GetBrightness	. 185
	6.3.3.41 AW_MPI_VENC_SetVEFreq	. 186
接接工具機構排放	6.3.3.42 AW_MPI_VENC_Set3DNR	187
A TOP OF THE PERSON NAMED IN COLUMN TO THE PERSON NAMED IN COLUMN	6.3.3.43 AW_MPI_VENC_Get3DNR	187
A STATE OF THE STA	6.3.3.44 AW MPI VENC GetCacheState	. 188
The state of the s	6.3.3.45 AW MPI VENC SetRefParam	. 189
	6.3.3.44 AW_MPI_VENC_GetCacheState	. 190
XI	6.3.3.47 AW MPI VENC SetHorizonFlip	. 190
	6.3.3.48 AW_MPI_VENC_GetHorizonFlip	. 191
	6.3.3.49 AW MPI VENC SetAdaptiveIntraInP	
	6.3.3.50 AW_MPI_VENC_EnableNullSkip	
	6.3.3.51 AW MPI VENC EnablePSkip	
	6.3.3.52 AW MPI VENC SaveBsFile	
	6.3.3.53 AW MPI VENC SetProcSet	
6.4	4 数据结构说明	. 196
	6.4.1 H264E_NALU_TYPE_E	. 196
	6.4.2 JPEGE PACK TYPE E	¥. 197
XXX	6.4.3 H265E_NALU_TYPE_E	. 197
EL ANTA	6.4.4 VENC_DATA_TYPE_U	. 198
7-1	6.4.5 VENC_PACK_S	. 199
-1.69 ,	6.4.6 VENC_STREAM_S	. 200
	6.4.7 VENC_ATTR_H264_S	. 201
	6.4.8 VENC_ATTR_H265_S	. 203
	6.4.9 VENC_ATTR_MJPEG_S	. 204
	6.4.10 VENC_ATTR_JPEG_S	. 205
	6.4.11 VENC_ATTR_S	. 207
	6.4.12 VENC_ATTR_H264_CBR_S	. 208
	6.4.13 VENC_ATTR_H264_VBR_S	. 209
	6.4:14 VENC_ATTR_H264_FIXQP_S	. 210
c.	6.4.15 VENC ATTR MIPEG CBR S	. 211
	6.4.16 VENC_ATTR_MJPEG_FIXQP_S	212
JAN YAN	6.4.17 VENC_ATTR_H265_CBR_S	. 212
THE STATE OF THE S		
	版权所有 ⑥ 珠海全志科技股份有限公司。保留一切权利	vii
张,	₩,	



(	ALLWIMER		Minderi	·····	, 3	文档密级: 秘密
_	6418	VENC ATTR	H265 VBR S	N. IV		. 213
12	6.4.19	VENC ATTR	_H265_FIXQP_S	A A A A A A A A A A A A A A A A A A A		214
A HELPHAN		_ V_V	_H265_ABR_S .	_ V_V		
-163x					. 4/5	
Kit.						
			_			
			_			
			<del>-</del>			
	6.4.26	VENC_GOP_	ATTR_S			221
	6.4.27	VENC_CHN_	ATTR_S			222
	6.4.28	VENC_CHN_	STAT_S			222
	6.4.29	VENC_EXIF	NFO_S		,	223
	6.4.30	VENC_JPEG	_THUMB_BUFFI	ER_S in <sup>der</sup> .		225
. 4	[X]	7/ 1/			7. X	1
EL HIST		./X. V				
		. /_		. J		
**************************************	6.4.35	VENC_FRAM	IE_RATE_S . 🧩	·' · · · · · · · · · · · · ·		229
	6.4.36	VENC_STRE	AM_BUF_INFO_	S		229
	6.4.37	VENC_PARA	M_INTRA_REFR	ESH_S		230
	6.4.38	VENC_PARA	M_REF_S		• • • • • • • • •	231
	6.4.39	VencHeader	Data	<b>M</b> .s		231
	6.4.42	VancSavaRS	File			<u>2</u> 33
	6.4.43	VeProcSet	inte indenti		· · · · · · · · · · · · · · · · · · ·	235
	6.5 错误码	verroeset .	AIV.	AIV.		
	A HIND	×4 <sup>1</sup>		A TOP TO SERVICE AND A SERVICE	<i>X</i> 4	W. 200
- Land	7 视频解码	<b>小</b> 拉尔之		A WALL TO SERVICE AND A SERVIC	Children of the Control of the Contr	237
	7.1 概述 .			· / · · · · · · · · · · · · · · · · · ·		237
类	7.2 功能描述	<b>公</b> 下.			%////	237
,						
	7.3.1					
			<del>_</del>			
			_			
			_	eam		
				eam		
	/^ / ·		C_Resume	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\		246
<b>护棚</b>	χ',			Harry Commencer of the	Harris Control of the	20
		- Table	版权所有 © 珠海全志科技股份	分有限公司。保留一切权利	- Table	viii
\$T		*\T	K.		***	





Au	WIMER	liinderi	ilinderi	文档密级: 秘密
	112	AMA MOL VIDEO C. 1	W/N	DIV.
XA		AW_MPI_VDEC_Seek	S	. 247
STATE A			12.45	248
		AW MPL VDEC_RegisterCallback	_/ ~~	249
		AW_MPI_VDEC_SetStreamEof 2 AW MPI VDEC ResetChn	41X-1	
		B AW MPI VDEC_ResetChnParam		
		AW_MFI_VDEC_SetChnFaram		
		5 AW MPI VDEC SendStream		
		S AW_MPI_VDEC_SendStream		
		AW_MFI_VDEC_Gettiliage		
		3 AW MPI VDEC SetRotate		
		AW_MPI_VDEC_SetRotate		
		) AW MPI VDEC ReopenVideoEngine.	. 0/1.	
		AW MPI VDEC SetVEFreq		
	<del>*</del>	2 AW MPI VDEC SetVideoStreamInfo	384	
THE WAY		3 AW MPI VDEC ForceFramePackage		260
7		构说明。	Little Control	260
, ,		HIDE MODE E		260
		VDEC STREAM S	37	263
		VDEC DECODE ERROR S		
		VDEC CHN STAT S		
		VDEC_CHN_PARAM_S		
7	,.s.s .6 错误码			267
, ,		7		207
8 M	IUX 模块	<u> </u>	À	269
8.	.1 概述		· · illug <sub>el</sub> · · · · · · · · · ·	269
8.	.2 功能描		(1)	269
X4 (1) X	8.2.1	muxGroup 和 muxChannel	§	. 270
中国农民人	8.2.2	状态图		×° 271
	8.2.3	API 和状态		271
8.	.3 API 参	5/k,	XX.	272
		AW_MPI_MUX_CreateGrp . ·		
		$AW\_MPI\_MUX\_DestroyGrp \ . \ . \ . \ . \ .$		
		$AW\_MPI\_MUX\_StartGrp\ .\ .\ .\ .\ .\ .$		
		AW_MPI_MUX_StopGrp		
		AW_MPI_MUX_GetGrpAttr		
		AW_MPI_MUX_SetGrpAttr		
		$AW\_MPI\_MUX\_SetH264SpsPpsInfo\ .\ .$		
		AW_MPI_MUX_SetH265SpsPpsInfo		
	8.3.9	AW_MPI_MUX_CreateChn	· Aline	278
X	V	O AW_MPI_MUX_DestroyChn	5,	V50
V.F.X.YA	8.3.11	AW_MPI_MUX_GetChnAttr	'	280



ALLWIN	MER CHILD	文档密级:秘密
KIND V	8.3.12 AW_MPI_MUX_SetChnAttr	280
AXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	8.3.13 AW MPI MUX SwitchFd	281
NING.	8.3.14 AW_MPI_MUX_RegisterCallback	282
	8.3.15 AW_MPI_MUX_SetMuxCacheDuration	<u>.</u> 283
	8.3.16 AW_MPI_MUX_SetSwitchFileDurationPolicy	283
	8.3.17 AW_MPI_MUX_GetSwitchFileDurationPolicy	
8.4		
	8.4.1 MUX GRP ATTR S	
	8.4.2 MUX_CHN_ATTR_S	
	8.4.3 CdxFdT	
	8.4.4 RecordFileDurationPolicy	
8.5		
	iinderi iinderi	obnii, de
A11/2	错误码	290
· XXX	概述	290
9.2	功能描述与使用	
The same of the sa	9.2.1 状态图	291
	9.2.2 API 和状态	291
9.3	API 参考	292
	API 参考	292
	9.3.2 AW_MPI_DEMUX_DestroyChn	293
	9.3.3 AW_MPI_DEMUX_RegisterCallback	
	9.3.4 AW_MPI_DEMUX_SetChnAttr	
	9.3.5 AW_MPI_DEMUX_GetChnAttr	
	9.3.6 AW_MPI_DEMUX_GetMediaInfo	
	9.3.7 AW_MPI_DEMUX_Start/	297
	9.3.8 AW_MPI_DEMUX_Stop	298
11/2	9.3.9 AW_MPI_DEMUX_Pause	
XATON	9.3.10 AW_MPI_DEMUX_ResetChn	
ARKY,	9.3.11 AW_MPI_DEMUX_Seek	301
	9.3.12 AW_MPI_DEMUX_getDmxOutPutBuf	301
	9.3.8 AW_MPI_DEMUX_Stop	(h) 302
9.4	数据类型	303
	9.4.1 STREAMTYPE_E	
	9.4.2 SOURCETYPE_E	
	9.4.3 CEDARX_MEDIA_TYPE	
	9.4.4 DEMUX_DISABLE_TRACKINFO	
	9.4.5 DEMUX_CHN_ATTR_S	
	9.4.6 DEMUX_VIDEO_STREAM_INFO_S	
	9.4.7 DEMUX_AUDIO_STREAM_INFO_S	308
	9.4.8 DEMUX_MEDIA_INFO_S	309
THE LO	9.4.9 DemuxCompOutputBuffer	
49.5	错误码	



ALLWIMER	<b>Allidet</b>	- Blinder	文档密级、秘密
10 音频		THE IT	313
10.1 概述			
10.2 功能	苗述	<u>,</u>	
10.2.	1 AI 设备状态图		315
10.2.	2 AO 设备状态图	· · · · · · · · · · · · · · · · · ·	
10.2.	3 AI 通道状态图		316
10.2.	4 AO 通道状态图		317
10.2.	5 AEnc 通道状态图		318
10.2.	6 ADec 通道状态图		319
10.2.	7 AIO 设备与通道		319
10.2.	8 音频回声消除		320
10.2.	9 音频降噪		320
10.3 音频	妾口调用流程介绍		320
	1 AI 通道使用流程		
10.3.	2 AO 通道使用流程	· · · · · · <u></u>	
	3 AEnc 通道调用流程		
10.3.4	4 ADec 通道调用流程		323
10.4 API	接口 <sub>來</sub> ®¹	(A)	324
10.4.	1 音频输入		324
	10.4.1.1 AW_MPI_AI_SetPub	Attr	324
	10.4.1.2 AW_MPI_AI_GetPub		
	10.4.1.3 AW_MPI_AI_Enable		
	10.4.1.4 AW_MPI_AI_Disable		
	10.4.1.5 AW_MPI_AI_Created		
	10.4.1.6 AW_MPI_AI_Destroy		
A STANTON OF THE PROPERTY OF T	10.4.1.7 AW_MPI_AI_ResetC 10.4.1.8 AW_MPI_AI_PauseC	hn	329
Alline	10.4.1.8 AW_MPI_AI_PauseC	hn	
ALL TO THE PARTY OF THE PARTY O	10.4.1.9 AW_MPI_AI_Resume	eChn	
A TANK	10.4.1.10 AW_MPI_AI_Enable	eChn	
All Control of the Co	10.4.1.11 AW_MPI_AI_Disable	leChn	
•	10.4.1.12 AW_MPI_AI_GetFr	ame	
	10.4.1.13 AW_MPI_AI_Release		*
	10.4.1.14 AW_MPI_AI_SetCh		
	10.4.1.15 AW_MPI_AI_GetCh		
	10.4.1.16 AW_MPI_AI_Enable	<del>-</del>	
	10.4.1.17 AW_MPI_AI_Disable		
	10.4.1.18 AW_MPI_AI_SetVq		
	10.4.1.19 AW_MPI_AI_GetVq 10.4.1.20 AW_MPI_AI_Enable		
is the state of th			
allinder.	10.4.1.21 AW_MPI_AI_Disable	ekModo	
WILL.	10.4.1.23 AW MPI_AI_SetTr	ackMode	
A THE THE PARTY OF	10.4.1.24 AW MPI_AI_Get11	J 187	5 187
14EV.		OT TOOL 1/2/4,	· · · · · · · · · · · · · · · · · · ·



	- 4	~ ~
A CONTRACTOR OF THE PARTY OF TH	10.4.1.25 AW_MPI_AI_SaveFile	. 344
15 XX	10.4.1.26 AW MPI AI QueryFileStatus	. 345
	10.4.1.27 AW MPI AI SetVqeVolume	. 346
	10.4.1.28 AW_MPI_AI_GetVqeVolume	. 347
	10.4.1.29 AW_MPI_AI_RegisterCallback	. 347
	10.4.1.30 AW_MPI_AI_SetVolume	. 348
	10.4.1.31 AW_MPI_AI_GetVolume	. 349
	10.4.1.32 AW_MPI_AI_SetMute	. 350
	10.4.1.33 AW_MPI_AI_GetMute	. 351
10.4.2	2 音频输出	. 352
	10.4.2.1 AW_MPI_AO_SetPubAttr	. 352
A STAN STAN STAN STAN STAN STAN STAN STA	10.4.2.2 AW_MPI_AO_GetPubAttr	. 353
llinder	10.4.2.3 AW_MPI_AO_ClrPubAttr	. 354
ARTIVE STATES	10.4.2.4 AW_MRI_AO_Enable	354
A TON	10.4.2.5 AW_MPI_AO_Disable	. 355
A HILL Y	10.4.2.6 AW_MPI_AO_EnableChn	. 356
	10.4.2.7 AW_MPI_AO_DisableChn	. 357
	10.4.2.8 AW_MPI_AO_StartChn	. 358
	10.4.2.9 AW_MPI_AO_StopChn	. 358
	10.4.2.10 AW_MPI_AO_RegisterCallback	. 359
	10.4.2.11 AW_MPI_AO_SendFrame	. 360
	10.4.2.12 AW_MPI_AO_EnableReSmp	
	10.4.2.13 AW_MPI_AO_DisableReSmp	
	10.4.2.14 AW_MPI_AO_PauseChn	
	10.4.2.15 AW_MPI_AO_ResumeChn	
teril	10.4.2.16 AW_MPI_AO_Seek	. 364
- Blind	10.4.2.17 AW_MPI_AO_ClearChnBuf ,	. 365
Elv.	10.4.2.18 AW MPI AO QueryChnStat	<sup>\\\\</sup> 366
	10.4.2.19 AW_MPI_AO_SetTrackMode	. 367
	10.4.2.20 AW_MPI_AO_GetTrackMode	. 368
	10.4.2.21 AW_MPI_AO_SetVolume	. 369
	10.4.2.22 AW_MPI_AO_GetVolume	. 369
	10.4.2.23 AW_MPI_AO_SetMute	. 370
	10.4.2.24 AW_MPI_AO_GetMute	
	10.4.2.25 AW_MPI_AO_SetVqeAttr	
	10.4.2.26 AW_MPI_AO_GetVqeAttr	
	10.4.2.27 AW_MPI_AO_EnableVqe	
	10.4.2.28 AW_MPI_AO_DisableVqe	
•	10.4.2.29 AW_MPI_AO_SetStreamEof	. 376
aindeni	10.4.2.30 AW_MPI_AO_SaveFile	. 376
AND AD A	10.4.2.31 AW_MPI_AO_QueryFileStatus	.2377
10.4.3	3 音频编码	270
N N 1	THE A Z I A WALLAND A DISTRICT CONDUCTOR DAY	<i>J</i> - / U

文档密级、秘密

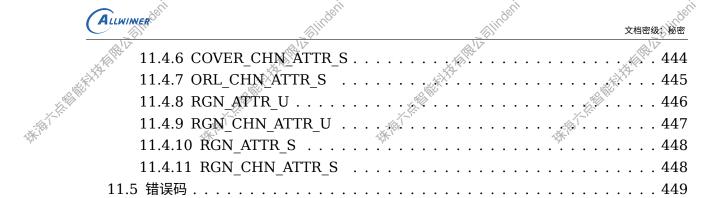


ALLWIMER	Blinderi	a dinderi	文档密级、秘密
ALLWIMERS ALLWIMERS	10.4.3.2 AW MPI_AENC_De	octrov.Chn	
12 XX	10.4.3.3 AW MPI_AENC_Se		
A STATE OF THE STA	10.4.3.4 AW_MPI_AENC_Ge	. 1/2/20	. V
de la	10.4.3.5 AW_MPI_AENC_Re	- / Ye	-/ V=
4Kito	10.4.3.6 AW_MPI_AENC_St	√ \	-X/ \
	10.4.3.7 AW_MPI_AENC_St		
	10.4.3.8 AW_MPI_AENC_Re		
	10.4.3.9 AW MPI AENC Qu		
	10.4.3.10 AW_MPI_AENC_F	<del>-</del>	
	10.4.3.11 AW_MPI_AENC_S	SetChnAttr	386
	10.4.3.12 AW_MPI_AENC_C	GetChnAttr	387
is a second of the second of t	10.4.3.13 AW_MPI_AENC_0		
10.4.4	1 音频解码	·····	389
10.4.4	10.4.4.1 AW_MPI_ADEC_Cr	reateChn	
A STATE OF THE STA	10.4.4.2 AW_MPI_ADEC_De	\`X/	\ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \
A CONTRACTOR OF THE PARTY OF TH	10.4.4.3 AW_MPI_ADEC_Re	W/V V	
T. T	10.4.4.4 AW_MPI_ADEC_Re		
**************************************	10.4.4.5 AW_MPI_ADEC_Se	ndStream	392
	10.4.4.6 AW_MPI_ADEC_Cl		393
	10.4.4.7 AW_MPI_ADEC_Ge		
	10.4.4.8 AW_MPI_ADEC_Re		
	10.4.4.9 AW_MPI_ADEC_Se		
	10.4.4.10 AW_MPI_ADEC_S		
	10.4.4.11 AW_MPI_ADEC_S		
	10.4.4.12 AW_MPI_ADEC_S		
nderi'	10.4.4.13 AW_MPI_ADEC_C	GetChnAttr	399
A STATE OF THE STA	10.4.4.14 AW_MPI_ADEC_F	ause , , , , ,	
10 F *h+P4	10.4.4.15 AW MPI_ADEC_S		
10.5 数据约			401
10.5.1	L 音频输入输出		401
- TEX 1-1	10.5.1.2 AI_CHN_PARAM_S		401
***	10.5.1.3 AUDIO_FRAME_S		*
	10.5.1.4 AEC FRAME S		
	10.5.1.5 AUDIO_AGC_CON		
	10.5.1.6 AI_AEC_CONFIG_S		
	10.5.1.7 AUDIO_ANR_CON		
	10.5.1.8 AUDIO_HPF_CON	<del>-</del>	
	10.5.1.9 AI_RNR_CONFIG_	<del>_</del>	
ing			
- Blino	10.5.1.10 AUDIO_EQ_CONI 10.5.1.11 AI_VQE_CONFIG	_S	
CONTRACTOR OF THE PARTY OF THE	10.5.1.12 AO VQE_CONFIC	G_S	
A A A A A A A A A A A A A A A A A A A	10.5.1.13 AUDIO_STREAM	V.K.	X K
High Control of the C	A THE STATE OF THE	A STATE OF THE STA	A THE STATE OF THE
	版权所有 © 珠海全志科技		xiii
₹K,	άκ,	灰,	ψ́κ'





ALLWIMER	alinderi .	Minderni	文档密级、秘密
10.5.1.14 10.5.1.14 10.5.1.14 10.5.1.14	4 AOCHN_STATE_S		. 414
10.5.1.1	5 AUDIO_FADE_S	V2L	415
10.5.1.1	6 AUDIO_SAMPLE_RATE_E		416
10.5.1.1	7 AUDIO_BIT_WIDTH_E	-/ Y-	
10.5.1.1	8 AIO_MODE_E	1051	
	9 AIO SOUND MODE E .		
	O AUDIO_HPF_FREQ_E		
	1 AQE_WORKSTATE_E		
	2 AUDIO_TRACK_MODE_E		
	3 AUDIO_FADE_RATE_E .		
	4 G726_BPS_E		
à 10.5.1.2	5 ADPCM_TYPE_E		424
10.5.2 音频编码			425
10.5.2.1	AENC_CHN_ATTR_S		
10.5.3 音频解码	,	, A.F	. 426
10.5.3.1	ADEC_CHN_ATTR_S	,	426
10.5.4 音频编解	码器类型与数据格式要求		427
10.6 错误码			428
10.6.1 音频输入	.错误码		428
10.6.2 音频输出	错误码		428
10.6.3 音频编码	错误码		429
10.6.4 音频解码	错误码		429
11 Region 模块	ALL		431
11.1 概述			
11.2 功能描述	T_RGN_Create		431
11.2.1 状态			431
11.3 API 参考			431
11.3.1 AW_MP	I_RGN_Create		432
11.3.2 AW_MP	I_RGN_Destroy	×	433
11.3.3 AW_MP	I_RGN_GetAttr		
	I_RGN_SetAttr		
11.5.5 AW_MP	I_RGN_SetBitMap		
—	I_RGN_AttachToChn		
<del>-</del>	I_RGN_DetachFromChn		
—	I_RGN_SetDisplayAttr		
	I_RGN_GetDisplayAttr		
<del>-</del>	YPE_E		
11.4.2 KGN_A	REA_TYPE_E AY_ATTR <sub>g</sub> S <sup>.</sup>		4.41
11.4.3 OVERL	AY_INVERT_COLOR_S	· · · · · · · · · · · · · · · · · · ·	<del>44</del> 1
11.4.4 OVERL	AY_CHN_ATTR_S		112
II.4.5 OVERLE		Ž,	.,,





版权所有 © 珠海全志科技股份有限公司。保留一切权利



文档密级: 秘密

William Control of the Control of th			AAAA	抽			. 4	A TON
HANDER X	2-1 ima.		C.K.					
	2-2 img.				17-15-Y			5
K K	3-1 img.	K. K.		**			KK.	25
								26
	3-3 img.							27
	5-1 img.							112
	5-2 img.							113
	5-3 img.							113
	5-4 img.							114
	5-6 img.		المرزن			· Jinden		118
	6-1 img.		A.V			\$1\Z		144
	6-2 img.		TAKE			24		
A MANUTE AND A STATE AND A	6-3 img.							×× 148
	7-1 img.							237
W. T.	7-2 img.	· · · · · · · · · · · · · · · · · · ·						238
Α,	8-1 img.					NE		269
	8-2 img.					W		270
	8-3 img.							271
	9-1 img.							290
	9-2 img.							291
	10-1 img							315
	10-2 img			1				316
	10-3 img			sti /		g <sup>1</sup> /2 jijideri		316
	10-4 Img		Allin .			- Allin		347
	1/20 C :		A STATE OF THE STA			\$1/V		210
A STATE OF THE STA	10-6 img 10-7 img		CAXXXX		· · · · · · · · · · · · · · · · · · ·			319
HAT WELL	10-7 IIIIg	45					45	320
### Little Heart		-1/2/-		J.			William Control of the Control of th	
ŔŁ,		45K,		**	*		\$\mathcal{L}'	

William The State of the State

Will Fifth the transfer of the state of the



前言

# 1.1 概述

MPP 媒体处理软件开发参考

1.2 产品版本

With the state of the state of

1.3 读者对象

本文档(本指南)主要适用于以下工程师;

1技术支持工程师

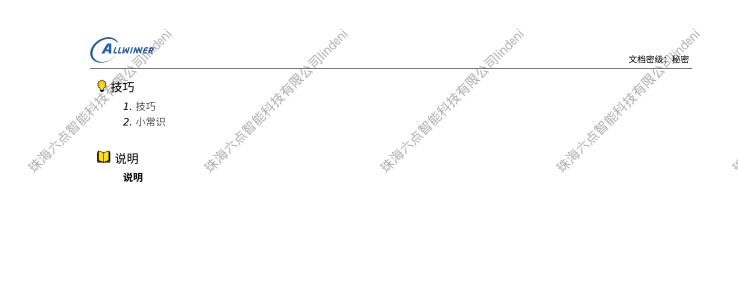
1 软件开发工程师

1.4 约定

1.4.1 符号约定

本文中可能出现的符号如下:

版权所有 © 珠海全志科技股份有限公司。保留一切权利









# 2

# 系统控制

# 2.1 概述

MPP 系统控制模块,根据芯片特性,完成硬件各个部件的复位、基本初始化工作,同时负责完成 MPP(Media Process Platform 媒体处理平台)系统各个业务模块的初始化、去初始化以及管理 MPP 系统各个业务模块的工作状态、提供当前 MPP 系统的版本信息等功能。

应用程序启动 MPP 业务前,必须完成 MPP 系统初始化工作。同理,应用程序退出 MPP 业务后,也要完成 MPP 系统去初始化工作,释放资源。

# 2.2 功能描述

- (1) 初始化 MPP 组件的运行环境,完成音频输入输出、视频输入输出等硬件设备的初始化配置。
- (2) 提供绑定组件的接口。
- (3) 提供媒体内存分配、释放、查询的接口。

### 2.2.1 状态

本组件没有内部线程,所以没有状态转换。

### 2.2.2 系统绑定

MPP 提供系统绑定接口(AW\_MPI\_SYS\_Bind),即通过数据接收者绑定数据源来建立两者之间的关联关系(只允许数据接收者绑定数据源)。绑定后,数据源生成的数据将自动发送给接收者。绑定关系是相互的,接收者处理完数据,如果传输数据的 Buffer 来自数据源,需归还 Buffer 给数据源。一个组件可以和多个组件建立绑定关系,绑定关系精确到组件端口。

目前 MPP 支持的绑定关系如下表 1-1 所示:

数据源	数据接收者
VI	VO
VENC	LEX XX



数据源	数据接收者
ISE	VO CONTRACT
ISE	VO
VENC	Ř <u>í</u> p.
VENC	MUX
AI	AO
AENC	
AENC	MUX
DEMUX	VDEC
ADEC	
VDEC	VO AO AI AO VO
ADEC	AO AI AO AO AO AO AO AO AO
ÃO	AI AIR
CLOCK	AO (SP)
VO	
DEMUX	Light Control of the
VDEC	
ISE	VO
VENC	
11	

#### 表 1-1 MPP 支持的绑定关系

附:

下列智能算法组件不支持绑定:

libaiMOD(运动目标检测),

libVLPR(车牌识别),

libeveface(人脸检测)。

### 2.2.3 组件端口数据传递模式

MPP 组件有两个端口 (inport/outport),inport 端口用于接收数据,在组件内部线程处理后生成新的数据,添加到输出队列的数据链表中进行管理,等待用户从 outport 端口主动拿数据或通过 outport 自动送到所绑定的下个组件。

组件端口数据传递分为 tunnel 模式和 non-tunnle 模式。自动传递数据到下个组件称为 tunnel 模式,手动管理、传递数据方式称为 non-tunnel 模式。Tunnel 模式及 non-tunnel 模式工作数据传递方式见下面的 1-1 和 1-2 图示。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

4

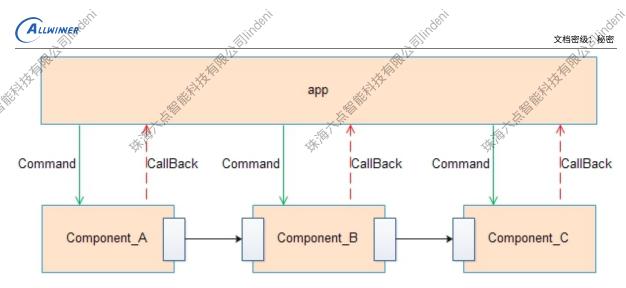


图 2-1: img

#### 图 1-1 MPP 组件 tunnel 模式

上图显示了组件间 tunnel 模式传递数据的工作原理。应用只需通过几个简单的 command 来创建、启动、停止、销毁组件,启动命令控制组件内部线程运行起来后,会源源不断地产生数据,并在内部的数据链表中进行统一管理,接下来把生成的数据数据自动发送到下个组件,下个组件内部线程利用输入端口中送来的数据生产出一笔数据,添加到其数据链表中进行管理,接下来将已经使用过的输入端中的数据还给前一个组件,使前一个组件释放该数据占用的 buffer 空间。

例如,当 ai 组件和 aenc 组件绑定时,即意味着 ai 的 outport 和 aenc 的 inport 进行绑定,那么当 ai 通道中存在 pcm 数据时,会自动将数据通过其 outport 端口送到 aenc 的 inport 端口,aenc 组件内部线程进行编码,生成压缩的音频数据后,进行管理,待送到 mux 组件或等待用户取数据,这取决于 aenc 组件输出端口的数据传递模式。

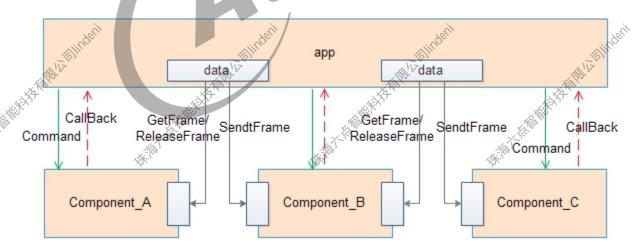


图 2-2: img

#### 图 1-2 MPP 组件 non-tunnel 模式

上图显示了 non-tunnel 模式的组件间数据传递方式的工作原理。应用创建、启动组件后,需通过 SendFrame()/SendStream() 等接口,往组件的 inport 输入端口送数据,然后应用调用 Get-Stream()/GetFrame() 等接口去取生产出的数据 (分为阻塞方式和超时等待方式),待组件内部



William The State of the State

线程利用 inport 端的数据生产出数据后,添加到输出数据队列中进行管理,此时应用的取数据函数调用方可退出 (阻塞方式),应用拿到生成的这笔数据进行处理,接下来仍需要利用这笔数据还帧给组件,主动告诉组件应用已经使用完这笔数据,可以释放其占用的 buffer 空间。

\*Notice\*: 使用 non-tunnel 模式时,应用如果往组件 inport 端口送数据不及时,会导致组件内部输入缓冲区的 underflow;如果不及时取走数据和还帧,会导致组件内部输出缓冲区的 overflow,因为内部线程一直源源不断地生产出新的数据,输出缓冲区队列逐渐变满直至爆仓,除非应用不往组件 inport 端口送数据,那么就不会生产出数据,输出缓冲区也不会爆仓。输入缓冲区的 underflow 和输出缓冲区的 overflow 都会导致组件无法处理数据,造成丢帧等后果。

各组件输入端口和输出端口绑定、非绑定支持如下表 1-2 中所示。

组件类型	输入端	输出端
VI	——allinde	tunnel non-tunnel
ISE	tunnel, non-tunnel	tunnel non-tunnel
AI	<del>1</del> <del>-</del> -	tunnel, non-tunnel
VENC	tunnel, non-tunnel	tunnel, non-tunnel
AENC	tunnel, non-tunnel	tunnel, non-tunnel
<b>V</b> DEC	tunnel, non-tunnel	tunnel, non-tunnel
ADEC	tunnel, non-tunnel	tunnel, non-tunnel
VO	tunnel, non-tunnel	4-11
AO	tunnel, non-tunnel	tunnel, non-tunnel
MUX	tunnel	
DEMUX	AIL	tunnel, non-tunnel

#### 表 1-2 MPP 支持的绑定关系

一般的规律是,source 型组件,其输入端接硬件设备用于获取 raw-data,输出端口支持 tunnel 和 non-tunnel 两种模式;sink 型组件,其输入端支持 tunnel 和 non-tunnel 两种模式,输出端直接接 render 类型硬件设备,进行数据的呈现,如声音的播放、图像的显示;filter 型组件,输入和输出端口都支持 tunnel 和 non-tunnel 两种模式。

### 2.2.4 媒体内存分配

用于多媒体处理的物理连续的内存分配,使用 ION 方式。系统控制模块封装了 ION 接口,提供给 APP 使用。V459 的硬件 IP 如果支持 IOMMU 模式,就不再需要物理连续的内存,IOMMU 内存的操作也通过 ION 驱动接口。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

6



# 2.2.5 通过 mpp 的 proc 节点实时查看硬件信息

mpp 可以为若干硬件驱动生成 proc 节点,挂载在 debugfs 文件系统下,供使用者在系统运行过程中实时查看驱动的运行信息。

配置内核 make kernel\_menuconfig,激活 SUNXI MPP SUNXI\_MPP\_AIO 就生成 proc 节点了。

系统启动后,在终端执行命令: mount -t debugfs none /sys/kernel/debug,挂载 debugfs 文件系统。然后通过 cat 指令查看 mpp 节点信息:

cat /sys/kernel/debug/mpp/vi

cat /sys/kernel/debug/mpp/ve

cat/sys/kernel/debug/mpp/vo

mpp 还可以进一步配置是否打开统计功能,例如 venc 组件:AW\_MPI VENC SetProcSet

# 2.3 API 参考

系统控制实现 MPP(Media Process Platform)系统初始化、系统绑定解绑、获取 MPP 版本号等功能。

# 2.3.1 AW MPI\_SYS\_Init

#### 【描述】

初始化 MPP 系统。包括音频输入输出、视频输入输出、视频编码、视频叠加区域、视频侦测分析 等都会被初始化。

#### 【语法】

ERRORTYPE AW MPI SYS Init();

#### 【参数】

参数名称 描述 输入/输出 ————————————————————— 无

【返回值】



返回值	描述	XA
0	成功	ALINE TO SERVICE AND ADDRESS OF THE PERSON O
≢ 0	失败,	参见错误码。

#### 【注意】

1 必须先调用 AW MPI SYS SetConf 配置 MPP 系统后才能初始化,否则初始化会失败。

l 如果多次初始化,仍会返回成功,但实际上系统不会对 MPP 的运行状态有任何影响。

#### 【举例】

无。

# 2.3.2 AW\_MPI\_SYS\_Exit

#### 【描述】

去初始化 MPP 系统。包括音频输入输出、视频输入输出、视频编码、视频叠加区域、视频侦测分析通道等都会被销毁或者禁用。

#### 【语法】

ERRORTYPE AW MPI SYS Exit();

#### 【参数】

, o.(1,		
参数名称	描述	输入/输出
无		× Kinger

#### 【返回值】

	7
返回值	描述
0	成功
非 0	失败,参见错误码。

#### 【注意】

l 去初始化时,如果有阻塞在 MPI 上的用户进程,则去初始化会失败。如果所有阻塞在 MPI 上的调用都返回,则可以成功去初始化。

1可以反复去初始化,不返回失败。



1 由于系统去初始化不会销毁音频的编解码通道,因此这些通道的销毁需要用户主动进行。如果创建这些通道的进程退出,则通道随之被销毁。

【举例】

无。

### 2.3.3 AW\_MPI\_SYS\_SetConf

【描述】

配置系统控制参数。

【语法】

ERRORTYPE AW\_MPI\_SYS\_SetConf(const MPP\_SYS\_CONF\_S\* pstSysConf);

#### 【参数】

参数名称 描述

pstSysConf 系统控制参数指针。静态属性(指只能在系统未初始化、未启用设备或通道时,才能设置的属性)

#### 【返回值】

返回值	描述	
0 Leri	成功	
非0000	失败,	参见错误码。
115		1.

# 【注意】

1. 只有在 MPP 整个系统处于未初始化状态,才可调用此函数配置 MPP 系统,否则会配置失败。

【举例】

无。

# $2.3.4~AW\_MPI\_SYS\_GetConf$

【描述】

获得系统控制参数。



#### 【语法】

ERRORTYPE AW\_MPI\_SYS\_GetConf(MPP\_SYS\_CONF\_S\* pstSysConf);

#### 【参数】

参数名称	描述		输入/输出
pstSysConf	系统控制参数指针。	静态属性。	输出

#### 【返回值】

			_		
返回值	描述		is		is.
O Allinden	成功	参见错误码。	Minder		uger
非 0	失败,	参见错误码。	7	IR IV	
	 	能获取配置。	16		
_		MIL	No		
Bind	/		•.		
in	/		in		100

#### 【注意】

1必须先调用 AW\_MPI\_SYS\_SetConf 成功后才能获取配置。

【举例】

无。

# 2.3.5 AW\_MPI\_SYS\_Bind

【描述】

绑定数据源通道端口和数据接收者通道端口。

【语法】

ERRORTYPE AW\_MPI\_SYS\_Bind(MPP\_CHN\_S\* pstSrcChn, MPP\_CHN\_S\* pstDestChn);

#### 【参数】

参数名称	描述	输入/输出
pstSrcChn	源通道指针	输入
pstDestChn	目的通道指针	输入

【返回值】



描述 返回值 0 成功 非 0 失败,参见错误码。

【注意】

无。

【举例】

无。

2.3.6 AW\_MPI\_SYS\_UnBind

【描述】

数据源到数据接收者解绑定接口。

【语法】

Se Republish to the second of ERRORTYPE AW\_MPI\_SYS\_UnBind(MPP\_CHN\_S\* pstSrcChn, MPP CHN S\* pst-DestChn);

【参数】

参数名称 输入/输出 描述 pstSrcChn 源通道指针。 pstDestChn 目的通道指针。 输入

【返回值】

返回值 描述 0 成功 非 0 失败,参见错误码。

【注意】

无。

版权所有 © 珠海全志科技股份有限公司。保留一切权利



# 2.3.7 AW\_MPI\_SYS\_GetBindbyDest

#### 【描述】

获取此通道上绑定的源通道的信息。

#### 【语法】

ERRORTYPE AW MPI SYS GetBindbyDest(MPP CHN S\*pstDestChn, MPP CHN S\* pstSrcChn);

#### 【参数】

pstDestChn 源通道指针 输入 返回值 描述 0 成功			
返回值 描述 0 成功	参数名称	描述	输入/输出
返回值 描述 0 成功	pstDestChn		ARL V
返回值 描述 0 成功	A CONTRACTOR OF THE PARTY OF TH		XX PORT
返回值 描述 0 成功	,		
	 返回值	-1501	
	0	成功 失败,参见错	NA

### 【返回值】

返回值	描述
0	成功
非 0	失败,参见错误码。

#### 【注意】

1 如果该通道绑定了 2 个以上的源通道(例如 Muxer 模块同时接收音频、视频编码通道的编码数 据),只返回最先绑定的源通道信息。

# 2.3.8 AW\_MPI\_SYS\_GetVersion

#### 【描述】

获取 MPP 的版本号。

#### 【语法】

ERRORTYPE AW\_MPI\_SYS\_GetVersion(MPP\_VERSION\_S\* pstVersion);



输入/输出 参数名称 描述

版本号描述指针。动态属性(指在任何时刻都可以设置的属性) pstVersion 输入

【返回值】

返回值	描述	
0	成功	
非 0	失败,	参见错误码。

【注意】

【举例】

- CS 2.3.9 AW\_MPI\_SYS\_GetCurPts

【描述】

获取 MPP 的当前

[TOC]

时间戳。

【语法】

ERRORTYPE AW\_MPI\_SYS\_GetCurPts(uint64\_t\* pu64CurPts);

【参数】

参数名称	描述	输入/输出
pu64CurPts	当前时间戳指针	输出

【返回值】

返回值、描述 成功 非 0 失败,

版权所有 © 珠海全志科技股份有限公司。保留-



【举例】

无。

# $2.3.10~AW\_MPI\_SYS\_InitPtsBase$

【描述】

初始化 MRP 的时间戳基准。

【语法】

ERRORTYPE AW\_MPI\_SYS\_InitPtsBase(uint64\_t u64PtsBase);

【参数】

参数名称 描述 输入/输 u64PtsBase 时间戳基准。单位:微秒。 输入

【返回值】

	4		
返回值	描述		,
O deni	成功		<
# 0	失败,	参见错误码。	
1/4		1770	

【注意】

1 初始化时间戳基准会将当前系统的时间戳强制置成 u64PtsBase,与系统原有时间戳没有任何约 束。因此,建议在媒体业务没有启动时(例如操作系统刚启动),调用这个接口。如果媒体业务 已经启动,建议调用 AW MPI SYS SyncPts 进行时间戳微调。

【举例】

无。

# 2.3.11 AW\_MPI\_SYS\_SyncPts



同步 MPP 的时间戳。

#### 【语法】

ERRORTYPE AW\_MPI\_SYS\_SyncPts(uint64\_t u64PtsBase);

#### 【参数】

参数名称	描述	输入/输出
u64PtsBase	时间戳基准。	输入

#### 【返回值】

 返回值 描述

 0 成功

 非 0 失败,参见错误码。

#### 【注意】

l 对当前系统时间戳(微秒级)进行微调,微调后不会出现时间戳回退现象。在多片之间做同步时,由于单板的时钟源误差可能比较大,建议一秒钟进行一次时间戳微调。

#### 【举例】

无。

# 2.3.12 AW\_MPI\_SYS\_MmzAlloc\_Cached

#### 【描述】

在用户态分配 MMZ 内存。内部从 ION 分配 iommu 内存。

#### 【语法】

ERRORTYPE AW\_MPI\_SYS\_MmzAlloc\_Cached(unsigned int\* pu32PhyAddr, void\*\* ppVirtAddr, unsigned int u32Len);

#### 【参数】

参数名称	描述	输入/输出
	iommu 物理地址指针	——————— 输出
ppVirtAddr	指向虚拟地址指针的指针	输出
u32Len	内存块大小	输入



【返回值】

	LeX 1
返回值	描述
0	成功。
非 0	失败,参见错误码。

【注意】

无。

【举例】

无。

2.3.13 AW\_MPI\_SYS\_MmzFree

【描述】

在用户态释放 MMZ 内存。

【语法】

et u? ERRORTYPE AW\_MPI\_SYS\_MmzFree(unsigned int u32PhyAddr, void\* pVirtAddr);

【参数】

参数名称	描述	输入/输出
u32PhyAddr	iommu 物理地址	输入
pVirtAddr	虚拟地址指针	输入

【返回值】

	XI.	
返回值	描述	
0	成功	
非 0	失败,参见错误码。	)

【注意】

无。



# 2.3.14 AW\_MPI\_SYS\_MmzFlushCache

### 【描述】

刷新 cache 里的内容到内存并且使 cache 里的内容无效。

#### 【语法】

ERRORTYPE AW\_MPI\_SYS\_MmzFlushCache(unsigned int u32PhyAddr, void\* pVitAddr, unsigned int u32Size);

#### 【参数】

参数名称	描述。	输入/输出
u32PhyAddr	待操作数据的起始物理地址。	输入
pVitAddr	待操作数据的起始虚拟地址指针。	输入
u32Size	待操作数据的大小。	输入
A TEN	A THE STATE OF THE	ER
	返回值 描述	

#### 【返回值】

返回值	描述	.118
0	成功	
非 0	失败,	参见错误码。

#### 【注意】

无。

【举例】

无。

# $2.3.15\ AW\_MPI\_SYS\_GetVirMemInfo$

#### 【描述】

根据虚拟地址获取对应的内存信息,包括物理地址及 cached 属性。

#### 【语法】

ERRORTYPE AW\_MPI\_SYS\_GetVirMemInfo(const void\* pVitAddr, SYS\_VIRMEM\_INFO\_S\* pstMemInfo);

# 【参数】



参数名称	描述	输入/输出
pVitAddr	虚地址。	输入
pstMemInfo	虚内存的信息	输出

【返回值】

返回值	描述	
0	成功	
非 0	失败,	参见错误码。

2.3.16 AW\_MPI\_SYS\_HANDLE\_ZERO

2.3.18 AV

2.3.18 AW MPI\_SYS\_HANDLE\_ISSET

2.3.19 AW\_MPI\_SYS\_HANDLE\_Select

#### 【描述】

把 mpp 组件实例模拟为文件句柄 fd,查询是否有数据可用。

#### 【语法】

ERRORTYPE AW MPI SYS HANDLE ZERO(handle set \*pHandleSet);

ERRORTYPE AW MPI SYS HANDLE SET(int handle, handle set \*pHandleSet);

ERRORTYPE AW MPI SYS HANDLE ISSET(int handle, handle set \*pHandleSet);

ERRORTYPE AW MPI SYS HANDLE Select(handle set \*pRdFds, int nMilliSecs);



参数名称	描述	输入/输出
pHandleSet	类似于 fd_s	at。 输入

【返回值】

返回值	描述	
0	成功	
非 0	失败,	参见错误码。

【注意】

无。

【举例】

无。

# 2.4 数据类型

# 2.4.1 视频公共类型

# 2.4.1.1 VIDEO\_FRAME\_S

【说明】

定义视频原始图像帧结构。

【定义】

typedef struct VIDEO\_FRAME\_S

{

unsigned int mWidth;

unsigned int mHeight;

VIDEO FIELD E mField;

PIXEL\_FORMAT\_E mPixelFormat;

VIDEO\_FORMAT\_E mVideoFormat;

Sin A State of the state of the



## COMPRESS MODE E mCompressMode;

unsigned int mPhyAddr[3];// Y, U, V; Y, UV; Y, VU

void\* mpVirAddr[3];

unsigned int mStride[3];

unsigned int mHeaderPhyAddr[3];

void\* mpHeaderVirAddr[3];

unsigned int mHeaderStride[3];

short mOffsetTop; /\* top offset of show area \*/

short mOffsetBottom; /\* bottom offset of show area \*/

short mOffsetLeft; /\* left offset of show area \*/

short mOffsetRight; /\* right offset of show area \*/

uint64 t mpts; //unit:us

unsigned int mTimeRef;

unsigned int mPrivateData;

VIDEO SUPPLEMENT S mSupplement;

int mEnvLV; //environment luminance value

unsigned int mWhoSetFlag;

uint64 t mFlagPts;

unsigned int mFrmFlag;

} VIDEO FRAME S;

## 【成员】

成员名称 描述 mWidth 装填图像的 buffer 的宽度。 装填图像的 buffer 的高度。 mHeight mField 帧场模式,目前只支持 VIDEO FIELD FRAME。 mPixelFormat 视频图像像素格式。 视频图像格式。只支持 VIDEO FORMAT LINEAR。未使用。 mVideoFormat mCompressMode 视频压缩模式。未使用。 mPhyAddr[3] 视频帧的 yuv 分量的物理地址。 mpVirAddr[3] 视频帧的 yuv 分量的虚拟地址。

area\*/

操稿

ik lā linderi

A THE PARTY OF THE



成员名称 描述 mStride[3] 视频帧的 yuv 分量的一行的跨距,单位为字节。 mHeaderPhyAddr[3] 未使用。 mpHeaderVirAddr[3] 未使用。 mHeaderStride[3] 未使用。 mOffsetTop 图像顶部剪裁宽度,单位为像素。是图像帧第一行像素的 Y 坐标。 图像底部剪裁宽度,单位为像素。是图像帧最后一行像素的 Y 坐标加 1。 mOffsetBottommOffsetLeft 图像左侧剪裁宽度,单位为像素。是图像帧左侧像素的 X 坐标。 图像右侧剪裁宽度,单位为像素。是图像帧右侧像素的 X 坐标加 1。 mOffsetRight mpts 视频帧 pts。单位微秒。

mTimeRef未使用。 mPrivateData 未使用。 mSupplement 未使用。《

MINERRA mEnvLV 采集图像帧时的环境亮度值。

## 【注意事项】

【相关数据类型及接口】

## VIDEO\_FRAME\_INFO\_S

## 【说明】

定义视频图像帧信息结构体。

## 【定义》

typedef struct VIDEO FRAME INFO S

VIDEO FRAME S VFrame;

unsigned int mId; //id identify frame uniquely

} VIDEO FRAME INFO S;

## 【成员】

成员名称 描述 **VFrame** 视频图像帧。 装填图像帧的 buffer 的 id。

【注意事项】

【相关数据类型及接口】

## **2.4.1.3 BITMAP\_S**

【说明】

定义位图图像信息结构。

【定义】

typedef struct BITMAP S

PIXEL FORMAT E mPixelFormat; /\* Bitmap's pixel format \*/

unsigned int mWidth; /\* Bitmap's width \*/

unsigned int mHeight; /\* Bitmap's height \*/

void\* mpData; /\* Address of Bitmap's data \*/

} BITMAP\_S;

【成员】

描述 成员名称

位图像素格式,支持 MM\_PIXEL\_FORMAT\_RGB\_8888 和 MM\_PIXEL\_FORMAT\_RGB\_ mPixelFormat

mWidth

位图宽度。

mHeight

位图高度。

mpData

位图数据起始虚地址。

【注意事项】

【相关数据类型及接口】

## 2.4.2 组件公共类型

## ${\bf 2.4.2.1 \ MPP Callback Info}$



通道的 callback 回调注册信息。

## 【定义】

typedef ERRORTYPE (MPPCallbackFuncType)(void cookie, MPP\_CHN\_S pChn, MPP\_EVENT\_TYPE event, void pEventData);

typedef struct MPPCallbackInfo {

void cookie; //EyeseeRecorder

MPPCallbackFuncType callback; //MPPCallbackWrapper

} MPPCallbackInfo;

【成员】

- In the state of		D参数
成员名称	描述	<u>"</u>
cookie	回调函数的 app 数据结构	 ]参数
callback	回调函数类型的指针	
	LLWIN	Jinden Jakur
定义	描述	A COLOR
DD EVE NII		<u>+</u>

【注意事项】

【相关数据类型及接口】

## 2.5 错误码

	dille	diffe	Alific
81	错误代码	宏定义	描述
	0xA0028006	ERR_SYS_NULL_PTR	空指针错误
	0xA0028010	ERR_SYS_NOTREADY	系统控制属性未配置
	0xA0028009	ERR_SYS_NOT_PERM	操作不允许
	0xA002800C	ERR_SYS_NOMEM ***	分配内存失败,如系统内存不足
	0xA0028003	ERR_SYS_ILLEGAL_PARAM	参数设置无效
	0xA0028012	ERR_SYS_BUSY	系统忙
	0xA0028008	ERR_SYS_NOT_SUPPORT	不支持的操作或类型。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

23



## 3.1 概述

视频输入模块实现的功能:用于接收并解析不同协议(Parallel、MIPI、Sub-lvds、Hispi、 Bt601/656/1120、Digital camera)传输过来的图像,通过 ISP 和 VIPP 模块处理后输出。

## 主要功能如下:

l 支持 BT.656、BT1120、BT601、Digital Camera 时序

1 单 ISP: ISP0: 2592x1936-30Fps。

## 3.2 功能框图

## 通路为:

 $CSI0 > ISP0 > VIPP[0 \sim 3]$ 

CSI[0~1]: 表示物理 Camera Signal Input Pasrse Device 的接口; CSI0/1 可以选择连接任 意一个 ISP

ISPO: 表示物理 ISP; ISP 可以选择连接任意多个 VIPP

VIPP[0~3]:表示物理 Scale + OSD + Mask 通道。每个 VIPP 配合一个 DMA 输出一路 Video 给到 DDR

Virvi[0~3]:表示每个物理通道虚拟 4 个虚通道输出。默认情况下推荐使用一个物理通道和-虚通道来采集视频数据。虚拟通道的图像属性是物理通道的复制品。

## CSI 输入设备

支持不同的时序输入,对输入的时序进行配置和解析

CSI、ISP、VIPP

数据处理不经过 DDR



## 3.3 VIPP Buffer 管理和使用

## 3.3.1 ViPP 非绑定情况下

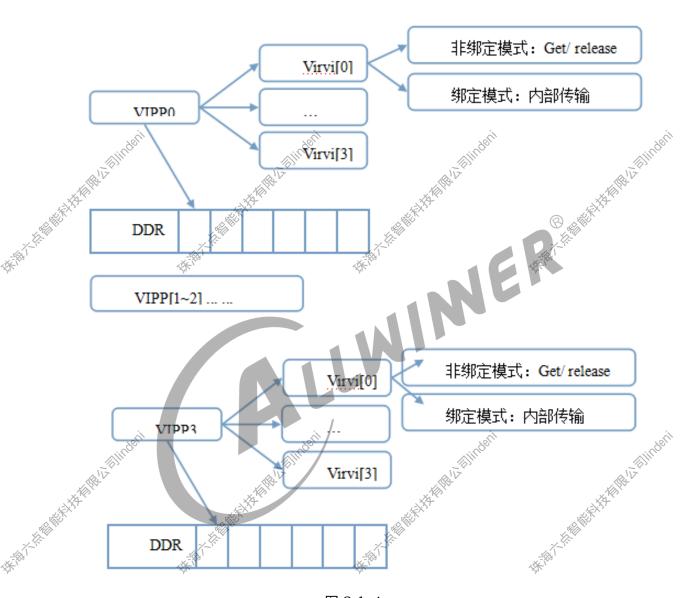


图 3-1: img

l 每个 VIPP 物理通道对应一段 Buff 空间,Buff 空间数目由 MPI 函数设定,由 Kernel Driver 层统一管理、分配、使用,默认为 5 个 Buff:ABCDE。

l 同一个 VIPP 设备下所有的虚通道共用同一个 VIPP buff。

l 用户通过 AW\_MPI\_VI\_GetFrame 获取 buff 数据,AW\_MPI\_VI\_ReleaseFrame 释放 buff 数据。必须成对使用。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

2



## 3.3.2 ViPP 绑定情况下

1 绑定情况下的 buff 和非绑定情况下的 buff 分配是一致的。

l 绑定情况下的 Buff 数据在组件直接内部传递。AW\_MPI\_VI\_GetFrame 与 AW\_MPI\_VI\_ReleaseFrame 函数不可使用。

注意: 同一个虚拟通道,同一时间只能使用绑定,或者非绑定其中一种方式获取 YUV 数据,不支持两种方式同时存在。

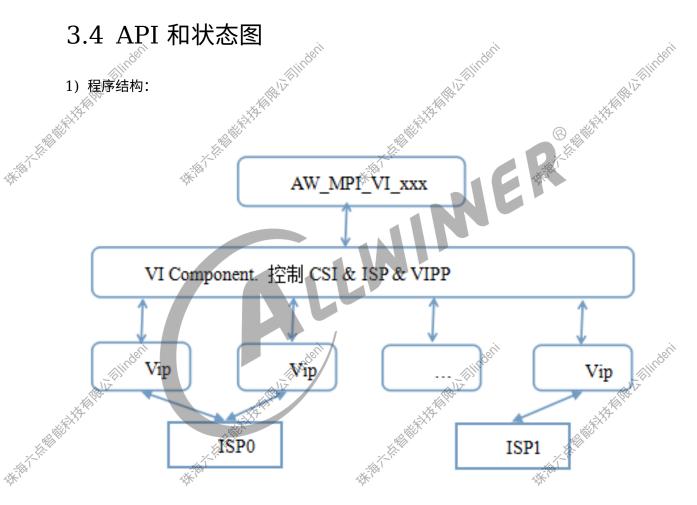


图 3-2: img

## 2) 状态转换图:

版权所有 © 珠海全志科技股份有限公司。保留一切权利

说明。

COMP StateLoaded: 组件完成初始化。

COMP\_StateIdle: 组件准备就绪。

COMP StateExecuting: 组件运行状态。

COMP StatePause: 组件暂停(挂起)状态。

COMP StateInvalid: 组件非法状态。

## 3.5 API 接口

AW MPI VI CreateVipp 创建 VIPP 物理设备

AW MPI VI DestoryVipp 销毁 VIPP 物理设备

AW MPI VI SetVippAttr 设置 VIPP 物理设备属性

AW MPI VI GetVippAttr 获取 VIPP 物理设备属性

AW\_MPI\_VI\_SetVippMirro 设置 VIPP 水平镜像

AW MPI VI SetVippMirror 设置 VIPP 或 sensor 水平镜像

AW MPI VI GetVippMirror 获取 VIPP 或 sensor 水平镜像

AW MPI VI SetVippFlip 设置 VIPP 或 sensor 垂直镜像

AW MPI VI GetVippFlip 获取 VIPP 或 sensor 垂直镜像

AW MPI VI EnableVipp 启动 VIPP 物理设备

版权所有 © 珠海全志科技股份有限公司。保留一切权利

27



AW\_MPI\_VI\_DisableVipp 停止 VIPP 物理设备

AW MPI VI SetRegion 设置显示区域

AW MPI VI DeleteRegion 更新显示区域

AW MPI VI UpdateOverlayBitmap 更新顶层覆盖的位图数据

AW MPI VI UpdateRegionChnAttr 更新显示区域的通道属性

AW MPI VI SetVippMirror 设置 vipp 的镜像(水平翻转)

AW MPI VI SetVippFlip 设置 vipp 的垂直翻转

AW\_MPI\_VI\_CreateVirChn 基于某个VIPP,创建虚通道

AW MPI VI DestoryVirChn 销毁虚通道

AW MPI VI GetVirChnAttr 设置虚通道属性

AW MPI VI SetVirChnAttr 获取虚通道属性

AW MPI VI EnableVirChn 启动虚通道

AW MPI VI DisableVirChn 停止虚通道

/\* 绑定情况下以下 API 不可用,数据在组件直接内部传递

AW MPI VI GetFrame 获取视频帧

AW MPI VI ReleaseFrame 释放视频帧

AW MPI VI Debug StoreFrame 捕获一帧数据到指定文

AW MPI VI SetVIFreq 设置 ISP 频率

AW MPI VI RegisterCallback 设置回调函数

## 3.5.1 AW\_MPI\_VI\_CreateVipp

【目的】

创建一个 VIPP 设备。

【语法】

AW S32 AW MPI VI CreateVipp(VI DEV ViDev);

\*/ Carlotte and the state of th



参数 描述

ViDev 需要创建的 VIPP 设备号

【返回值】

返回值描述SUCCESS成功错误码参考 mm\_common\_vi.h 中的错误码描述。

【需求】

头文件: mpi\_vi.h

库文件: libmpp\_vi.so

【注意】

VI 所需系统资源(数据源类型、接口、数据位宽、时序、场、输入/输出格式 PIN、CLK)配置完成,系统启动后才会形成/dev/videoX 节点,此处针对节点进行初始化操作。

【举例】

无。

## 3.5.2 AW\_MPI\_VI\_DestoryVipp

【目的】

销毁 VIPP 物理设备

【语法】

AW S32 AW MPI VI DestoryVipp(VI DEV ViDev);

【参数】

参数 描述 ViDev 需要销毁的的 VIPP 设备号

【返回值】



返回值 描述 SUCCESS 成功 错误码 参考 mm common vi.h 中的错误码描述

【需求】

l 头文件: mpi\_vi.h

l 库文件: libmpp vi.so

【注意】

l 该函数会关闭/dev/videoX 设备节点,并销毁 AW\_MPI\_VI\_CreateVipp 函数申请的所有的资 

【举例】

1无。

## 3.5.3 AW\_MPI\_VI\_SetVippAttr

【目的】

设置 VIPP 物理设备属性

【语法】

AW\_S32 AW\_MPI\_VI\_SetVippAttr(VI\_DEV ViDev, VI\_ATTR\_S \*pstAttr);

【参数】

参数	描述
ViDev	需要设置属性的 VIPP 设备号
VI_DEV_ATTR_S	属性结构体(静态属性)

## 【返回值】

返回值	描述
SUCCESS	成功
错误码	参考 mm_common_vi.h 中的错误码描述。

【需求】



```
l 头文件: mpi vi.h
```

Ì库文件: libmpp vi.so

## 【注意】

VIPP 设备创建成功后,需要设置 format 格式、buf 数量、nbufs 个数、memtype、 nplanes、type。参见 VI\_ATTR\_S 结构体描述。

## 【举例】

int ret;

VI ATTR S stAttr;

AW MPI VI GetVippAttr(0, &stAttr);

stAttr.format.width = 1920;

stAttr.format.height = 1080;

Set of the second of the secon stAttr.format.pixelformat = V4L2 PIX FMT NV21M;

stAttr.format.field = V4L2 FIELD NONE;

stAttr.fps = 30;

stAttr.memtype = V4L2 MEMORY MMAP;

stAttr.nbufs = 10;

stAttr.nplanes = 2;

stAttr.type = V4L2 BUF TYPE VIDEO CAPTURE MPLANE;

ret = AW\_MPI\_VI\_SetVippAttr(0, &stAttr);

if (SUCCESS != ret) {

return -1;

}

## 3.5.4 AW\_MPI\_VI\_GetVippAttr

【目的】

获取 VIPP 物理设备属性

【语法】

版权所有 © 珠海全志科技股份有限公司。保留一切权利



AW\_S32 AW\_MPI\_VI\_GetVippAttr(VI\_DEV ViDev, VI\_ATTR\_S \*pstAttr);

## 【参数】

参数	描述
ViDev	需要获取属性的 VIPP 设备号
VI_DEV_ATTR_S	属性(可以动态获取)

## 【返回值】

返回值	描述
SUCCESS 错误码	成功。 参考 mm_common_vi.h 中的错误码描述。
······································	成功。 参考 mm_common_vi.h 中的错误码描述。
vi.so buf 数量、r	bufs 个数、memtype、nplanes、type 等。

【需求】

头文件: mpi\_vi.h

库文件: libmpp vi.so

【注意】

获取 format 格式、buf 数量、nbufs 个数、memtype、nplanes、type 等。

【举例】

参见上一条的例子。

3.5.5 AW\_MPI\_VI\_SetVIFreq

【目的】

设置 VIPP 物理设备的运行频率

【语法】

AW S32 AW MPI VI SetVIFreq(VI DEV ViDev, int nFreq);

【参数】

参数 描述

ViDev 需要获取属性的 VIPP 设备号 nFreq 频率值(单位: Hz)静态属性



## 【返回值】

描述	A STATE OF THE STA
成功 参考 mm_commo	· n_vi.h 中的错误码描述。
	<b>以</b>

## 【需求】

头文件: mpi vi.h

库文件: libmpp vi.so

# 3.5.6 AW\_MPI\_VI\_EnableVipp 【目的】 自动 VIPP 物理设备 【语法】 W\_S32 AW MPY

【参数】

参数	描述
ViDev	使能 VIPP 设备

## 【返回值】

返回值	描述
SUCCESS	成功
错误码	参考 mm_common_vi.h 中的错误码描述。



头文件: mpi vi.h

库文件: libmpp\_vi.so

【注意】

该函数作用是申请 buffer,将所有申请到的 buffer 放入队列,然后开启数据流,创建数据捕获线程。

【举例】

无。

## 3.5.7 AW\_MPI\_VI\_DisableVipp

【目的】

停止 VIPP 物理设备

【语法】

AW\_S32 AW\_MPI\_VI\_DisableVipp(VI\_DEV ViDev);

【参数】

Vipp(VI\_DEV ViDev);

参数

描述

ViDev 禁止的 VIPP 设备号

【返回值】

返回值 描述 SUCCESS 成功

错误码 参考 mm common vi.h 中的错误码描述。

【需求】

头文件: mpi\_vi.h

库文件: libmpp vi.so

【注意】

该函数会停止数据流,释放掉所有申请到的 buffer。

【举例】



## 3.5.8 AW\_MPI\_VI\_SetVippFlip

【目的】

设置 vipp 翻转。

【语法】

AW S32 AW MPI VI SetVippFlip(VI DEV ViDev, int Value);

参数 描述

ViDev VIPP 设备号

- Republish the state of the st Value 翻转标志(0:正常;1:翻转)动态属性

【返回值】

描述 返回值

SUCCESS 成功

参考 mm common\_vi.h 中的错误码描述。 错误码

【需求】

头文件: mpi\_vi.h

库文件: libmpp vi.so

【注意】

该函数与 AW\_MPI\_ISP\_SetFlip 函数作用一致。

【举例】

无。

3.5.9 AW\_MPI\_VI\_GetVippFlip

【目的】

获取 vipp 的翻转标志。



## 【语法】

AW\_S32 AW\_MPI\_VI\_GetVippFlip(VI\_DEV ViDev, int \*Value);

【参数】

参数 描述

ViDev VIPP 设备号

翻转标志动态属性 Value

## 【返回值】

· ix -返回值 描述。 成功 **SUCCESS** 参考 mm\_common\_vi.h 中的错误码描述。

【需求】

l 头文件: mpi vi.h

l 库文件: libmpp vi.so

【注意】

l 获取到的 Value 值(0:未翻转; 1: 翻转),这里说的翻转指的是垂直方向的翻转。该函数与 AW\_MPI\_ISP\_GetFlip 函数的作用是一样的。

l 无。

## 3.5.10 AW\_MPI\_VI\_SetVippMirror

【目的】

设置 vipp 的图像水平镜像。

【语法】

AW\_S32 AW\_MPI\_VI\_SetVippMirror(VI\_DEV ViDev, int Value);



参数 描述

ViDev VIPP 设备号

Value 镜像标志(0:不翻转,1:翻转)动态属性

【返回值】

返回值 描述
SUCCESS 成功
错误码 参考 mm\_common\_vi.h 中的错误码描述。

【需求】

头文件: mpi vi.h

库文件: libmpp vi.so

【注意】

设置的 Value 值(0:不镜像;1:镜像),这里说的镜像指的是水平方向的翻转。该函数与 AW\_MPI\_ISP\_SetMirror 函数的作用是一样的。

【举例】

无。

## $3.5.11~AW\_MPI\_VI\_GetVippMirror$

【目的】

获取 vipp 的水平镜像标志。

【语法】

AW S32 AW MPI VI GetVippMirror(VI DEV ViDev, int \*Value);

【参数】

参数 描述
ViDev VIPP 设备号
Value 镜像标志动态属性

【返回值】



返回值	描述	
SUCCESS	成功	A CONTRACTOR OF THE PROPERTY O
错误码	参考 mi	m_common_vi.h 中的错误码描述。

## 【需求】

头文件: mpi vi.h

库文件: libmpp vi.so

## 【注意】

获取到的 Value 值(0: 未镜像; 1: 镜像),这里说的镜像指的是水平方向的翻转。该函数与 AW MPI ISP GetMirror 函数的作用是一样的。

## 3.5.12 AW\_MPI\_VI\_CreateVirChn 【目的】 基于某个 VIPP,创建虚拟通道。 【语法】

AW S32 AW MPI VI CreateVirChn(VI DEV ViDev, VI CHN ViCh, void \*pAttr);

## 【参数】

参数	描述
AW_DEV ViDev	已经创建的 VIPP 设备通道取值范围: [0,3]
VI_CHN ViCh	需要创建的 VIR 虚拟通道取值范围: [0,3]
void *pAttr	NULL 静态属性

## 【返回值】

返回值	描述
SUCCESS	成功。
错误码	参考 mm_common_vi.h 中的错误码描述。

版权所有 © 珠海全志科技股份有限公司。保留一切权利



## 【需求】

l 头文件: mpi vi.h

l 库文件: libmpp\_vi.so

## 【注意】

l 保证 VIPP 创建后在进行 AW\_MPI\_VI\_CreateVirChn 操作。

## 【举例】

```
/declaration/
```

int ret = 0

AW\_CHN ViCh;

/\* init VI device\*/

ret = AW MPI VI InitCh(Videv, ViCh, NULL);

if (SUCCESS != ret)

{

return -1;

}

## Dest 3.5.13 AW\_MPI\_VI\_DestoryVirChn

## 【目的】

销毁虚拟通道

## 【语法】

AW\_S32 AW\_MPI\_VI\_DestoryVirChn(VI\_DEV ViDev, VI\_CHN ViCh);

## 【参数】

参数	描述
AW_DEV ViDev	需要销毁的 VIPP 设备通道取值范围: [0, 3]
VI_CHN ViCh	需要销毁的 VIR 虚拟通道取值范围: [0,3]

版权所有 © 珠海全志科技股份有限公司。保留一切权利



## 【返回值】

		L-X
返回值	描述	the state of the s
SUCCESS	成功	
错误码	参考 m	m_common_vi.h 中的错误码描述。

## 【需求】

l 头文件: mpi vi.h

l 库文件: libmpp vi.so

3.5.14 AW\_MPI\_VI\_SetVirChnAttr

【目的】

设置虚拟通道属性

【语法】

W\_S32 AW

AW\_S32 AW\_MPI\_VI\_SetVirChnAttr(VI\_DEV ViDev, VI\_CHN ViCh, void \*pAttr);

## 【参数】

1	参数	描述
-/	AW_DEV ViDev	VIPP 设备通道取值范围: [0,3]
	VI_CHN ViCh	VIR 虚拟通道取值范围: [0,3]
	void *pAttr	Default = NULL 静态属性

## 【返回值】

返回值	描述
SUCCESS	成功。影
错误码	参考 mm_common_vi.h 中的错误码描述。



## 【需求】

l 头文件: mpi\_vi.h

l 库文件: libmpp\_vi.so

【注意】

l 无。

【举例】

1无。

3.5.15 AW\_MPI\_VI\_GetVirChnAttr

【目的】

获取虚拟通道属性

【语法】

AW\_S32 AW\_MPI\_VI\_GetVirChnAttr(VI\_DEV ViDev, VI\_CHN ViCh, void \*pAttr);

【参数】

参数

AW DEV ViDev VIPP 设备通道取值范围: [0,3] VI CHN ViCh VIR 虚拟通道取值范围: [0, 3]

void \*pAttr Default=NULL

【返回值】

返回值 描述

**SUCCESS** 成功

错误码 参考 mm\_common\_vi.h 中的错误码描述。

【需求】

l 头文件: mpi vi.h

l 库文件: libmpp\_vi.so

【注意】

版权所有 © 珠海全志科技股份有限公司。保留一切权利

【举例】

1无。

## $3.5.16~AW\_MPI\_VI\_EnableVirChn$

【目的】

启动虚拟通道

【语法】

Charles the state of the state AW\_\$32 AW\_MPI\_VI\_EnableVirChn(VI\_DEV ViDev, VI\_CHN ViCh);

【参数】

参数 描述

要使能的 VIPP 设备通道 ViDev 要使能的 VIR 虚拟通道 ViCh

【返回值】

返回值 描述

SUCCESS 成功

错误码 参考 mm\_common\_vi.h 中的错误码描述。

【需求】

l 头文件: mpi\_vi.h

l 库文件: libmpp\_vi.so

【注意】

l无。

【举例】

1无。

O Republication of the second of the second



## 3.5.17 AW\_MPI\_VI\_DisableVirChn

【目的】

停止虚拟通道

【语法】

AW S32 AW MPI VI DisableVirChn(VI DEV ViDev, VI CHN ViCh);

【参数】

参数 描述

ViDev 要禁止的 VIPP 设备通道 ViCh 要禁止的 VIR 虚拟通道

【返回值】

返回值 描述

SUCCESS 成功

错误码 参考 mm\_common\_vi.h 中的错误码描述。

【需求】

l 头文件: mpi\_vi.h

l 库文件: libmpp\_vi.so

【注意】

1无。

【举例】

1无。

## 3.5.18 AW\_MPI\_VI\_GetFrame

【目的】

获取 VI 设备一帧图像,属性包括 width、height、field、pixelformat、timestamp index、VirAddr、mem\_phy、size等。

【语法】



AW\_S32 AW\_MPI\_VI\_GetFrame(VI\_DEV ViDev, VI\_CHN ViCh, VIDEO\_FRAME\_INFO\_S \*pstFrameInfo, AW\_S32 s32MilliSec);

## 【参数】

	4 DE-1
参数	描述
ViDev	VIPP 设备通道
ViCh	VIR 虚通道
pstFrameInfo	帧信息
s32MilliSec	Timeout 超时时间设置动态属性

	Mindeni		Mindeni		EFFE HOLD TO THE STATE OF THE S
返回值	描述		RIV.		A TOP TO THE PERSON OF THE PER
SUCCESS	成功		A. C.		XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
错误码	参考 mm_	common_vi.l	h 中的错误码	B描述。 <sup>®</sup>	ç
.h _vi.so			N		
c 设置的时间	值并且还没有	<b>す获取到帧数</b> 据	居时函数就会	返回。	deri

## 【需求】

l 头文件: mpi vi.h

l 库文件: libmpp vi.so

## 【注意】

l 超过 s32MilliSec 设置的时间值并且还没有获取到帧数据时函数就会返回。

【举例】

J£。

## 3.5.19 AW\_MPI\_VI\_ReleaseFrame

## 【目的】

释放 VI 设备图像内存资源。

## 【语法】

AW\_S32 AW\_MPI\_VI\_ReleaseFrame(VI\_DEV ViDev, VI\_CHN ViCh, VIDEO\_FRAME\_INFO\_S \*pstFrameInfo);



参数	 描述
ViDev	VIPP 设备通道
ViCh	VIR 虚通道
VI_FRAME_BUF_INFO_S *pstFrameInfo	帧信息

## 【返回值】

返回值	描述	
SUCCESS 错误码	成功 参考 mm common vi h 中的错误码描述	
错误码 .h _vi.so	参考 mm_common_vi.h 中的错误码描述。  ***********************************	
	SetVIFreq Malindani	

## 【需求】

1头文件: mpi\_vi.h

l 库文件: libmpp\_vi.so

【注意】

1无。

【举例】

1 无。

## 3.5.20 AW\_MPI\_VI\_SetVIFreq

【目的】

设置 ISP 频率。

【语法】

AW\_S32 AW\_MPI\_VI\_SetVIFreq(VI\_DEV ViDev, int nFreq);

【参数】

参数 描述 VIPP 设备通道 ViDev

ISP 频率,单位:MHz



	A 117	\$1\Z
返回值	描述	A PART OF THE PART
SUCCESS	成功	
错误码	参考 m	m_common_vi.h 中的错误码描述。

【需求】

l 头文件: mpi\_vi.h

l 库文件: libmpp vi.so

【注意】

※Ĩ无。

## 3.5.21 AW\_MPI\_VI\_RegisterCallback 【目的】 注册 vi 组件实例的回调函数,接收 vi 组件等

ERRORTYPE AW\_MPI\_VI\_RegisterCallback(VI\_DEV ViDev, MPPCallbackInfo \*pCall> back);

【参数】

参数	描述
ViDev	VIPP 设备通道
MPPCallbackInfo *pC	Callback 回调函数数据结构

## 【返回值】

返回值	描述
SUCCESS	成功
错误码	参考 mm_common_vi.h 中的错误码描述。

需求】



l 头文件: mpi\_vi.h

l 库文件: libmpp\_vi.so

【注意】

1 无。

【举例】

l 无。

## 3.6 数据结构

## 3.6.1 VI ATTR S

【说明】

定义 Vi 输入设备接口属性

【定义】

typedef struct awVI\_ATTR\_S {

enum v4l2\_buf\_type type;

enum v4l2 memory memtype;

struct v4l2\_pix\_format\_mplane format;

unsigned int nbufs;

unsigned int nplanes;

unsigned int fps;

unsigned int capturemode;

unsigned int use current win;

unsigned int wdr mode;

unsigned int drop\_frame\_num; // drop frames number after enable vipp device(default 0).

int antishake enable; // 1:enable; 0:disable

ANTISHAKE\_ATTR\_S antishake\_attr;

deri

版权所有 © 珠海全志科技股份有限公司。保留一切权利



## } VI\_ATTR\_S;

## 【成员】

	~/ 45	-/ V.	~/ \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\	
成员名称	描述	茶卷	W. Tall	类形1
type	默认值:V	4L2_BUF_TYPE_VIDE	O_CAPTURE_MPLANE。	采集数
	据方式,不	下能修改		
memtype	默认值:V	4L2_MEMORY_MMAF	P。采集数据内存使用方式,	不建议
	修改			
format	struct v4	l2_pix_format_mplane	{u32 width;u32 he	ight;
	u32 pix	xelformat; /* 默认 V4L2	_PIX_FMT_NV21M */1	ı32
	field;u	32 colorspace; struct v	/4l2_plane_pix_format	
Ja To Relation of the later of	plane_fm	t[VIDEO_MAX_PLANE	[S];	u8 <sub>inderi</sub>
	reserved	[11];} <b>attribute</b> ((pacl	ked));pixelformat 的参数数	数据格
	式: field	的取值		
A STATE OF THE STA	V4L2_PIX	K_FMT_NV12V4L2_PIX	K_FMT_NV21V4L2_PIX_I	FMT_NV21MV4L2_PI
,	V4L2_PIX	K_FMT_SBGGR12V4L2	PIX_FMT_SGBRG12V4	L2_PIX_FMT_SGRBC
	V4L2_FII	ELD_NONE V4L2_FIEI	LD_TOP V4L2_FIELD_BO	OTTOM
Ś	<sup>⋙</sup> V4L2_FII	ELD_INTERLACED V4	L2_FIELD_SEQ_TB	Æ,
	V4L2_FII	ELD_SEQ_BT V4L2_FI	ELD_ALTERNATE	
	V4L2_FII	ELD_INTERLACED_TB	V4L2_FIELD_INTERLAC	CED_BT
nbufs	默认值:5	YUV/RAW 内存节点缓冲	个数。	
nplanes	plane 个数	数,属于返回值,不设置。		
fps	默认值: 2	5。设置 Sensor 的帧率		
capturemode	默认值: V	4L2_MODE_VIDEO V4	1L2_MODE_VIDEO	
	V4L2_MC	ODE_IMAGE V4L2_MO	DE_PREVIEW	à
use_current_wi	n 0:表示不	管之前有没有设置过分辨	率,都重新找当前设置分辨率	<b>率最近的</b>
- ////		⟨⟨S⟩ = ⟨ , , , , , , , , , , , , , , , , , ,	(	

分辨率 1:表示使用之前设置过的分辨率注意:0 相当于从新设置 sensor 输出的分辨率,1 相当于 sensor 在有输出的情况下,使用后端

VIPP 做视频缩小处理,出不同规格的分辨率。 默认值: 00: normal1: DOL 2: Sensor Commanding

drop\_frame\_num 丢帧计数

antishake\_enable Antishake 功能使能 antishake\_attr Antishake 功能属性

## 【注意事项】

wdr mode

无。

## 【相关数据类型及接口】

enum v4l2\_buf\_type {

V4L2\_BUF\_TYPE\_VIDEO\_CAPTURE = 1,



```
V4L2 BUF TYPE VIDEO OUTPUT = 2,
  V4L2 BUF TYPE VIDEO OVERLAY = 3,
 V4L2_BUF_TYPE_VBI_CAPTURE = 4,
 V4L2 BUF TYPE VBI OUTPUT = 5,
 V4L2 BUF TYPE SLICED VBI CAPTURE = 6,
 V4L2 BUF TYPE SLICED VBI OUTPUT = 7,
 V4L2 BUF TYPE VIDEO OUTPUT OVERLAY = 8,
                                                                                           The state of the s
 V4L2_BUE_TYPE_VIDEO_CAPTURE_MPLANE = 9,
 V4L2 BUF TYPE VIDEO OUTPUT MPLANE = 10,
/* Deprecated, do not use */
 V4L2 BUF TYPE PRIVATE = 0x80,
};
enum v4l2 memory {
 V4L2 MEMORY MMAP = 1,
 V4L2 MEMORY USERPTR = 2,
 V4L2 MEMORY OVERLAY = 3,
 V4L2 MEMORY DMABUF = 4
enum v4l2 field {
 V4L2 FIELD ANY = 0, /* driver can choose from none, top, bottom interlaced
 depending on whatever it thinks is approximate ... */
 V4L2 FIELD NONE = 1, /* this device has no fields ... */
 V4L2 FIELD TOP = 2, /* top field only */
 V4L2 FIELD BOTTOM = 3, /* bottom field only */
 V4L2_FIELD_INTERLACED = 4, /* both fields interlaced */
 V4L2_FIELD_SEQ_TB = 5, /* both fields sequential into one buffer, top-bottom order
```

版权所有 © 珠海全志科技股份有限公司。保留一切权利

4



V4L2 FIELD SEQ BT = 6,/\* same as above + bottom-top order \*/

V4L2 FIELD ALTERNATE = 7, /\* both fields alternating into separate buffers \*/

V4L2\_FIELD\_INTERLACED\_TB = 8, /\* both fields interlaced, top field first and the top field is

transmitted first \*/

V4L2\_FIELD\_INTERLACED\_BT = 9, /\* both fields interlaced, top field first and the bottom field is

transmitted first \*/

**}**;

## 3.6.2 RGN ATTR S

【说明】

定义 region 的属性。

【定义】

typedef struct RGN ATTR S

{

Report of the state of the stat RGN\_TYPE\_E enType; /\* region type \*/

RGN\_ATTR\_U unAttr; /\* region attribute \*/

} RGN ATTR S;

## 【成员】

成员名称	描述
enType	region 的类型。取值有以下几种
	OVERLAY_RGNCOVER_RGNCOVEREX_RGNOVERLAYEX_RGN
unAttr	region 属性(共用体类型)。typedef union RGN_ATTR_U{
	OVERLAY_ATTR_S stOverlay; OVERLAYEX_ATTR_S stOverlayEx;}
	RGN_ATTR_U;

【注意事项】



## 【相关数据类型及接口】

```
typedef struct OVERLAY_ATTR_S
```

/\* bitmap pixel format, now only support ARGB1555 or ARGB4444 \*/

PIXEL FORMAT E mPixelFmt;

/\* background color, pixel format depends on "enPixelFmt" \*/ unsigned int mBgColor;

/\* region size,W:[4,4096],align:2,H:[4,4096],align:2 \*/

SIZE S mSize;

}OVERLAY\_ATTR\_S;

## 3.6.3 RGN\_CHN\_ATTR\_S

【说明】

定义 region 的属性。

【定义】

typedef struct RGN\_CHN\_ATTR\_S

BOOL bShow;

RGN\_TYPE\_E enType; /\* region type \*/

RGN\_CHN\_ATTR\_U unChnAttr; /\* region attribute \*/

} RGN\_CHN\_ATTR\_S;

## 【成员】

成员名称	描述		
bShow	bool 类型,表示是否隐藏。		
enType	参见上面一条的描述	deri	



	ALLWIMER		Allindeni		Minden	文档密级、秘密
	成员名称	描述		XXXIII		A STATE OF THE PARTY OF THE PAR
W. T.	unChnAttr	OVERLA stCoverC	on 属性(共用体 Y_CHN_ATTR_S Chn; COVEREX_ YEX_CHN_ATTI	stOverlayChr CHN_ATTR_S	; COVER_CHI stCoverExChi	N_ATTR_S n;
	【注意事项】					<u> </u>
	【相关数据类型	及接口】	, sti		, sci	0
	typedef struct	OVERLAY	Y_CHN_ATTR_S		A TE THINGS	in the state of th
	/* X:[0,4096],a	align:4,Y:10	0,4096],align:4	*/		(S) Hall Hall
	POINT_S stPo	int		* The state of the		
	/* background	l an foregr	round transpare	nce when pixe	l format is AR	GB1555
	* the pixel for	mat is AR(	GB1555,when tl	ne alpha bit is	1 this alpha is	value!
	* range:[0,128	3]	11	N.		

\*/

unsigned int mFgAlpha;

/\* background an foreground transparence when pixel format is ARGB1555

\* the pixel format is ARGB1555, when the alpha bit is 0 this alpha is value

\* range:[0,128]

unsigned int mBgAlpha;

unsigned int mLayer; /\* OVERLAY region layer range:[0,7]\*/

OVERLAY\_QP\_INFO\_S stQpInfo;

OVERLAY INVERT COLOR S stInvertColor;

}OVERLAY\_CHN\_ATTR\_S;



## 3.6.4 BITMAP\_S

## 【说明】

位图数据描述结构体

## 【定义】

{

```
typedef struct BITMAP S
```

PIXEL\_FORMAT\_E mPixelFormat; /\* Bitmap's pixel format \*/

unsigned int mWidth; /\* Bitmap's width \*/

unsigned int mHeight; /\* Bitmap's height \*/

void\* mpData; /\* Address of Bitmap's data \*/

} BITMAP S;

## 【成员】

, ru. m ri _L min mon or	1 1		
int mWidth; /* Bitma	ap's width */	MA Vallinderi	a the state of the
int mHeight; /* Bitn	nap's height */	A CONTRACTOR OF THE PARTY OF TH	A RELIVE
Data; /* Address of E	Bitmap's data */	0	, Š <sub>X</sub> ,
P_S;		NE	
成员名称描述	术		
mPixelFormat 位图	图的像素格式,参见该项的	【相关数据类型及接口】	
mWidth 位图	图的宽		
mHeight 位置	图的高		
mpData 位图	<b>图的像素数据</b>	ä	à

## 【注意事项】

## 【相关数据类型及接口】

```
typedef enum PIXEL FORMAT E
```

{

MM PIXEL FORMAT RGB 1BPP = 0,

MM PIXEL FORMAT RGB 2BPP,

MM PIXEL FORMAT RGB 4BPP,

MM PIXEL FORMAT RGB 8BPP,

MM\_PIXEL\_FORMAT\_RGB\_444,



MM\_PIXEL\_FORMAT\_RGB 4444,

MM PIXEL FORMAT RGB 555,

MM\_PIXEL\_FORMAT\_RGB\_565,

MM PIXEL FORMAT RGB 1555,

/\* 9 reserved \*/

MM PIXEL FORMAT RGB 888,

MM PIXEL FORMAT RGB 8888,

MM PIXEL FORMAT RGB PLANAR 888,

MM\_PIXEL\_FORMAT\_RGB\_BAYER\_8BPP,

MM\_PIXEL\_FORMAT\_RGB\_BAYER\_10BPP,

MM PIXEL FORMAT RGB BAYER 12BPP,

MM PIXEL FORMAT RGB BAYER 14BPP,

MM PIXEL FORMAT RGB BAYER, /\* 16 bpp \*/

MM PIXEL FORMAT YUV A422,

MM\_PIXEL\_FORMAT\_YUV\_A444,

MM PIXEL FORMAT YUV PLANAR 422,

MM\_PIXEL\_FORMAT\_YUV\_PLANAR\_420, //YU12

MM PIXEL FORMAT YUV PLANAR 444,

MM PIXEL FORMAT YUV SEMIPLANAR 422, //NV16

MM\_PIXEL\_FORMAT\_YUV\_SEMIPLANAR\_420, //NV12

MM PIXEL FORMAT YUV SEMIPLANAR 444,

MM\_PIXEL\_FORMAT\_UYVY\_PACKAGE\_422,

MM PIXEL FORMAT YUYV PACKAGE 422,

MM\_PIXEL\_FORMAT\_VYUY\_PACKAGE\_422,

MM\_PIXEL\_FORMAT\_YCbCr\_PLANAR,

MM\_PIXEL\_FORMAT\_SINGLE,

MM\_PIXEL\_FORMAT\_YVU\_PLANAR\_420, //YV12

A Republished the second of th

MR Vallinderi

rderi di Lalinde

反权所有 © 珠海全志科技股份有限公司。保留一切权利

54



```
MM_PIXEL_FORMAT_YVU_SEMIPLANAR_422, //NV61
MM_PIXEL_FORMAT_YVU_SEMIPLANAR_420,#/NV21
MM_PIXEL_FORMAT_YUV_AW_AFBC, //by andy
MM PIXEL FORMAT YUV AW LBC 2 0X,
MM_PIXEL_FORMAT_YUV_AW_LBC_2_5X,
MM PIXEL FORMAT YUV AW LBC 1 0X,
MM PIXEL FORMAT BUTT
int clipcount; /* number of clips */
int chromakey;
nt global_alpha;
pt inv_w_rar*
} PIXEL_FORMAT_E;
int inv_h_rgn[8];
int inv th;
int reverse close[64]; /osd reverse close, 1:close, 0:open/
int rgb cover[8];
int glb_alpha[64];
void *bitmap[64];
 struct v4l2_rect region[64]; /* overlay or cover win */
```



### 【成员】

5	^ <del></del>	- Art A		- Art	
,	成员名称	描述	A PARTY OF THE PAR	H. H	
	clipcount	OSD& 视频遮挡数目	- 10 T		
	chromakey	只支持以下图片格式:	SK.	SK.	
		V4L2_PIX_FMT_RC	GB555V4L2_PIX_FM	T_RGB444V4L2_PIX_	FMT_RGB32
	global_alpha	全局 Alpa 值			
	inv_w_rgn	Inverse region wid	th		
	inv_h_rgn	Inverse region heigh	ght		
	inv_th	threshold for inver	rse luminance		
	reverse_close	Enable or disable i	nverse 1:disable;0:e	nable	
	rgb_cover	Rgb value used for	cover	<sub>28</sub> ri	deni
	glb_alpha	allino.		ino	<b>Allino</b>
	*bitmap[64]	180	2. Mask: 必须设置为 ]	Null	SIL
-	region[64]	图片位置大小	XX	- XA	
5				(C) THE PROPERTY OF THE PROPER	
	【注意事项】				
		ж,	张,		
-	无。				
	【相关数据类型及	·接口】			
	们以外加入主人		4 1AI I		
t	ypedef enum V	'IDEO_FIELD_E			
	r				
-	{		,		
7	VIDEO EIEID	TOP = 0v1 /* even fice	14 */	<u> </u>	

### 【注意事项】

```
【相关数据类型及接口】
```

```
typedef enum VIDEO FIELD E
{
```

```
VIDEO FIELD TOP = 0x1, /* even field */
```

```
VIDEO FIELD BOTTOM = 0x2, /* odd field */
```

```
WIDEO_FIELD_INTERLACED = 0x3, /* two interlaced fields */
```

```
VIDEO FIELD FRAME = 0x4, /* frame */
```

VIDEO FIELD BUTT

```
} VIDEO FIELD E;
```

typedef enum PIXEL FORMAT E

{

MM PIXEL FORMAT RGB 1BPP = 0,

MM\_PIXEL\_FORMAT\_RGB\_2BPP

MM PIXEL FORMAT RGB 4BPP,



MM PIXEL FORMAT RGB 8BPP,

MM PIXEL FORMAT RGB 444,

MM\_PIXEL\_FORMAT\_RGB\_4444,

MM PIXEL FORMAT RGB 555,

MM PIXEL FORMAT RGB 565,

MM PIXEL FORMAT RGB 1555,

/\* 9 reserved \*/

MM\_PIXEL\_FORMAT\_RGB\_888,

MM PIXEL FORMAT RGB 8888,

MM\_PIXEL\_FORMAT\_RGB\_PLANAR\_888,

MM PIXEL FORMAT RGB BAYER 8BPP,

MM PIXEL FORMAT RGB BAYER 10BPP,

MM PIXEL FORMAT RGB BAYER 12BPP,

MM PIXEL FORMAT RGB BAYER 14BPP,

MM\_PIXEL\_FORMAT\_RGB\_BAYER, /\* 16 bpp \*/

MM PIXEL FORMAT YUV A422,

MM\_PIXEL\_FORMAT\_YUV\_A444

MM PIXEL FORMAT YUV PLANAR 422,

MM PIXEL FORMAT YUV PLANAR 420, //YU12

MM\_PIXEL\_FORMAT\_YUV\_PLANAR\_444,

MM PIXEL FORMAT YUV SEMIPLANAR 422, //NV16

MM PIXEL FORMAT YUV SEMIPLANAR 420, //NV12

MM PIXEL FORMAT YUV SEMIPLANAR 444,

MM\_PIXEL\_FORMAT\_UYVY\_PACKAGE\_422,

MM\_PIXEL\_FORMAT\_YUYV\_PACKAGE\_422,

MM\_PIXEL\_FORMAT\_VYUY\_PACKAGE\_422,

MM\_PIXEL\_FORMAT\_YCbCr\_PLANAR,

A September 1 Sept

rible la linder

版权所有 © 珠海全志科技股份有限公司。保留一切权利

57



```
MM PIXEL FORMAT SINGLE,
MM_PIXEL_FORMAT_XVU_PLANAR_420, //YV12
MM_PIXEL_FORMAT YVU SEMIPLANAR 422, //NV61
MM PIXEL FORMAT YVU SEMIPLANAR 420, //NV21
MM PIXEL FORMAT BUTT
} PIXEL FORMAT E;
typedef enum VIDEO FORMAT E
{
VIDEO FORMAT LINEAR = 0x0, /* nature video line */
WIDEO FORMAT TILE = 0x1, /* tile cell: 256pixel x 16line, default tile mode */
VIDEO FORMAT TILE64 = 0x2, /* tile cell: 64pixel x 16line */
                                    NIME
VIDEO FORMAT BUTT
} VIDEO FORMAT E;
typedef enum COMPRESS MODE E
{
COMPRESS MODE NONE = 0x0, /* no compress */
COMPRESS_MODE_SEG = 0x1, /* compress unit is 256 bytes as a segment, default
seg mode */
COMPRESS MODE SEG128 = 0x2, /* compress unit is 128 bytes as a segment */
COMPRESS_MODE_LINE = 0x3, /* compress unit is the whole line */
COMPRESS MODE FRAME = 0x4, /* compress unit is the whole frame */
COMPRESS MODE BUTT
} COMPRESS MODE E;
typedef struct VIDEO FRAME S
{
unsigned int mWidth;
unsigned int mHeight;
```



```
VIDEO FIELD E mField;
PIXEL FORMAT E mPixelFormat;
VIDEO_FORMAT_ mVideoFormat;
COMPRESS MODE E mCompressMode;
unsigned int mPhyAddr[3];// Y, U, V; Y, UV; Y, VU
void* mpVirAddr[3];
unsigned int mStride[3];
                                                                                                                                                                                             C Republic to the state of the 
short mOffsetTop; /* top offset of show area */
short mOffsetBottom; /* bottom offset of show area */
short mOffsetLeft; /* left offset of show area */
short mOffsetRight; /* right offset of show area */
uint64 t mpts; //unit:us
unsigned int mExposureTime; /* every frame exp time *
unsigned int mFramecnt; /* rename mPrivateData to Framecnt_exp_start */
int mEnvLV; /* environment luminance value */
unsigned int mWhoSetFlag; /* reserve(8bit)|COMP TYPE(8bit)|DEV NUM(8bit)|CHN NUM(8bit)
uint64 t mFlagPts; /* when generate this flag, unit(us) */
unsigned int mFrmFlag;
} VIDEO_FRAME_S;
```

# 3.6.6 VIDEO\_FRAME\_INFO\_S

【说明】

VI 视频输出属性

【定义】

typedef struct VIDEO\_FRAME\_INFO\_S

{

RATE OF THE PARTY OF THE PARTY



```
VIDEO_FRAME_S VFrame;
```

unsigned int mId;

} VIDEO\_FRAME\_INFO\_S;

【成员】

描述 成员名称

VFrame Buf 数据信息结构属性

mId Buf 唯一 ID 号

### 【注意事项】

【相关数据类型及接口】

typedef struct VIDEO\_FRAME S

{

unsigned int mWidth;

unsigned int mHeight;

VIDEO\_FIELD\_E mField;

State of the state PIXEL FORMAT E mPixelFormat;

VIDEO FORMAT E mVideoFormat;

COMPRESS\_MODE\_E mCompressMode;

unsigned int mPhyAddr[3]; // Y, U, V 或 Y, UV 或 Y, VU

void\* mpVirAddr[3]; //虚拟地址,对应于物理地址

unsigned int mStride[3];

unsigned int mHeaderPhyAddr[3];

void\* mpHeaderVirAddr[3];

unsigned int mHeaderStride[3];

short mOffsetTop; /\* top offset of show area \*/

short mOffsetBottom; /\* bottom offset of show area

版权所有 © 珠海全志科技股份有限公司。保留-



short mOffsetLeft; /\* left offset of show area \*/

short mOffsetRight; /\* right offset of show area \*

uint64\_t mpts; //unit:us

unsigned int mExposureTime; /\* every frame exp time \*/

unsigned int mFramecnt; /\* rename mPrivateData to Framecnt\_exp\_start \*/

int mEnvLV; //environment luminance value

unsigned int mWhoSetFlag;

uint64_t mFla	gPts;	_%	ii derii
unsigned int r	gPts; nFrmFlag;	HA RING TO THE SECOND TO THE SECOND TO THE SECOND THE S	Rath Hard Hard Linderin
	N. C.	A STATE OF THE STA	A THE PARTY OF THE
} VIDEO_FRA	ME_S;		Q AND THE
X .		H.	
2 7 4#10	T-76-161		
3.7 错误	11-5		
		1 N	
错误码	宏定义		描述
0xA0108002	ERR_VI_INVALID_CHNID		无效的 VI 通道号
0xA0108003	ERR_VI_INVALID_PARA		无效的参数
0xA0108006	ERR_VI_INVALID_NULL_PTR		空指针
0xA0108007	ERR_VI_FAILED_NOTCONFIG		模块未配置
0xA0108008	ERR_VI_NOT_SUPPORT	Minde	模块未配置 设备不支持 不允许
0xA0108009	ERR_VI_NOT_PERM	RIV	不允许
0xA0108001	ERR_VI_INVALID_DEVID	A A A A A A A A A A A A A A A A A A A	无效的 VI 设备号
0xA010800C	ERR_VI_NOMEM	ALL MARKET	无可用的内存
0xA010800E	ERR_VI_BUF_EMPTY	J. W.	数据缓冲区为空
0xA010800F	ERR_VI_BUF_FULL		数据缓冲区为满
0xA0108010	ERR_VI_SYS_NOTREADY		系统还未准备好
0xA0108012	ERR_VI_BUSY		VI 设备正忙
0xA0108041	ERR_VI_FAILED_NOTENABLE	3	设备未使能
0xA0108042	ERR_VI_FAILED_NOTDISABLI	E	设备未禁止(处于使能状态)
0xA0108040	ERR_VI_CFG_TIMEOUT		配置超时
0xA0108043	ERR_VI_NORM_UNMATCH		不匹配
0xA0108044	ERR_VI_INVALID_PHYCHNID		无效的物理通道
0xA0108045	ERR_VI_FAILED_NOTBIND	_%	设备未绑定 设备已经绑定 VI 设备不存在
0xA0108046	ERR_VI_FAILED_BINDED	- Bline	设备已经绑定
0xA0108047	ERR_VI_UNEXIST	ALL LAND	VI 设备不存在
0xA0108048	ERR_VI_EXIST	SEXE TO THE SE	VI 设备已经存在
No.	A TOTAL CONTRACTOR OF THE PARTY	A VINCO	A PARTY



	ALLWIMER		Mindeni	a dilinde		文档密级、秘密
	错误码	宏定义	in the second		描述	A STATE OF THE STA
4	0xA0108014	ERR_VI_SAI	MESTATE	A STATE OF THE STA	状态相同	(常见于状态转换)
	0xA0108015	ERR_VI_INV	/ALIDSTATE	1/2/2	无效的状态	\$
KK.K.	0xA0108016	ERR_VI_INC	CORRECT_STA	TE_TRANSITION	不正确的	犬态转换
	0xA0108017	ERR VI INC	CORRECT STA	ATE OPERATION	不正确的》	<b>犬</b> 态操作

And the state of t

版权所有 © 珠海全志科技股份有限公司。保留一切权利

Will Little White Heart Heart Little Committee in the Com



# 4.1 概述

### 4.1.1 文档目的

介绍 VO 模块的使用方式,以供开发人员可以快速根据本文档进行基于 VO 模块的开发。

## 4.1.2 VO 简介

- 支持 linux 标准的 framebuffer 接口
- 支持 lcd(hv/lvds/cpu/dsi) 输出
- 支持多图层叠加混合处理
- VO 模块主要处理与视频输出显示相关的功能,主要功能如下: 持 linux 标准的 framebuffer 接口 持 lcd(hv/lvds/cpu/dsi) 输出 持多图层叠加混合处理 特多种显示效果<sup>从工严</sup> ′ • 支持多种显示效果处理(alpha, colorkey, 图像增强,亮度/对比度/饱和度/色度调整)
- 支持智能背光调节
- 支持多种图像数据格式输入 (argb,yuv)
- 支持图像缩放处理
- 支持截屏
- 支持图像转换

## 4.1.3 术语解释

说明文中涉及的专业术语。



# 4.2 图层

## 4.2.1 图层操作说明

显示中最重要的资源是图层,v459 VO 中支持 1 路显示设备,第 0 路显示设备支持 3 个显示通道,通道 0,1 为视频图层通道,通道 2 为 UI 图层通道。每个显示通道包含 4 个图层。通道 0,1 的图层都支持缩放,通道 2 的图层不支持缩放。图层由 disp、channel、layer\_id 三个索引唯一确定(disp:0/1,channel:0/1/2/3,layer id:0/1/2/3)。

需要注意的是 channel 0.1 通道下对应 layer\_id 为  $0_{3 \, \text{Hr}, \, \text{$rac{1}{8}$}}$   $0_{3 \, \text{H$ 

正常情况下,使用 0 路显示设备(本系统默认使用第 0 路显示设备)就可以满足用户需求。用户可以选择 disp: 0 的某个通道对应的图层来播放视频,或者选择另外一个图层显示 UI,不同图层之间可以设置优先级、alpha 等参数,进行叠加显示。

对于内核来说,12 个 Layer 可以看作从  $0\sim11$  线性排布的(由 channel\*4+layer\_id 计算得到,默认 disp 为 0),其中第  $0_{7}$  个是视频图层,第 8 11 个是 UI 图层,视频图层的  $0_{3,4}$  7 图层的属性需要分别保持一致。

V459 只有 channel0, 1, 2 三个通道。

- 设置图层参数并使能:接口为 AW\_MPI\_VO\_SetVideoLayerAttr、AW\_MPI\_VO\_EnableVideoLayer。
- 释放图层:接口 AW MPI VO Disable Video Layer,参数为需要释放的图层号。
- 打开/关闭图层:接口为 AW MPI VO OpenVideoLayer/AW MPI VO CloseVideoLayer。

## 4.2.2 显示输出设备操作说明

V459 VO 支持 LCD 显示输出设备。开启显示输出设备有几种方式,第一种是在 sys\_config.fex 中配置 [disp] 的初始化参数,显示模块在加载时将会根据配置来初始化选择的显示输出设备;第二种是在 kernel 启动后,调用 VO 模块的 API 接口去开启或关闭指定的输出设备,以下是操作的说明:

● 切换到某个具体的显示输出设备:接口是 AW\_MPI\_VO\_SetPubAttr,参数是一个 VO\_PUB\_ATTR\_S 类型的结构体,其中第二个参数 enIntfType 参数用来指定显示设备。

版权所有 © 珠海全志科技股份有限公司。保留一切权利



# 4.2.3 图层 size 与 crop

图层 Frame Buffer 有两个与 size 有关的参数,分别是 size 与 crop。Size 表示 buffer 的完整尺寸,crop 则表示 buffer 中需要显示裁减区。如下图所示,完整的图像以 size 标识,而矩形框住的部分为裁减区,以 crop 标识,在屏幕上只能看到 crop 标识的部分,其余部分是隐藏的,不能在屏幕上显示出来。

## 4.2.4 图层 crop 和 screen\_win

Screen\_win 为 crop 部分 buffer 在屏幕上显示的位置。如果不需要进行缩放的话,crop 和 screen\_win 的 width,height 是相等的,如果需要缩放,需要用 scaler\_mode 的图层来显示,crop 和 screen\_win 的 width,height 可以不等。

## 4.2.5 alpha

Alpha 模式有三种:

- Gloabal alpha: 全局 alpha, 也叫面 alpha, 即整个图层共用一个 alpha, 统一的透明度
- Pixel alpha: 点 alpha,即每个像素都有自己单独的 alpha,可以实现部分区域全透,部分区域半透,部分区域不透的效果
- Global\_pixel alpha:可以是说以上两种效果的叠加,在实现 pixel alpha 的效果的同时,还可以做到淡入淡出的效果。

1

# 4.3 输出设备介绍

- VO 支持屏、HDMI 以及 cvbs 等输出
- 屏的接口有很多类型,该平台支持 RGB/CPU/LVDS/DSI 接口。

# 4.4 模块状态转换

各种状态的定义如下所示:

◆ COMP\_StateIdle: 有资源,不传输数据。



• COMP StateExecuting: 有资源,传输数据,处理数据。

▶ COMP\_StatePause: 有资源,传输数据,不处理数据。

• COMP StateLoaded: 无资源。

• COMP\_StateInvalid: 非法状态。

# 4.5 API 接口

# 4.5.1 AW\_MPI\_VO\_Enable

## 【目的】

使能指定的 VO 设备,构造 VODevInfo 结构体并加入全局链表。

### 【语法】

ERRORTYPE AW\_MPI\_VO\_Enable(VO\_DEV VoDev);

### 【参数】

<b>Allinderi</b>	- Alinderi - Alinderi
备,构造 VODevInfo 结构体并加入:	全局链表。
音、构造 VODevInfo 结构体并加入	全局链表。
V_MPI_VO_Enable(VO_DEV VoDe	v);
	NE
参数 描述	输入输出
VoDev 需要使能的 VO 设备编号	

### 【返回值】

返回值	描述	_ Blinds
SUCCESS	成功	
错误码。	参考 mm_c	omm_vo.h 中的错误码描述。

### 【需求】

• 头文件: mpi\_vo.h

库文件: libmpp\_vo.so

### 【注意】

• 在进行 VO 相关操作之前应该首先调用此接口,保证相应的 VO 设备被使能。

### 【举例】

无。



# 4.5.2 AW\_MPI\_VO\_Disable

### 【目的】

禁用指定的 VO 设备,同时释放相关的资源。

### 【语法】

ERRORTYPE AW\_MPI\_VO\_Disable(VO\_DEV VoDev);

### 【参数】

参数	描述	输入输出
VoDev	需要禁用的 VO 设备编号	输入。

### 【返回值】

返回值	描述		
SUCCESS	成功	\$K,	
错误码	参考 n	nm_comm_vo.h 中的错误	吴码描述。

### 【需求】

• 头文件: mpi vo.h

• 库文件: libmpp\_vo.so

### 【注意】

◆ 该操作会将指定 VO 设备结构体从全局链表中移除,该函数需要与 AW\_MPI\_VO\_Enable 配对使用,在整个 VO 模块使用完毕之后调用该函数进行相关资源的释放。

### 【举例】

无。

# 4.5.3 AW\_MPI\_VO\_SetPubAttr

### 【目的】

设置 VO 显示设备的背景色,指定 VO 设备的类型(HDMI、LCD 等),设置显示尺寸(720p、1080p 等)、频率。

版权所有 © 珠海全志科技股份有限公司。保留一切权利



### 【语法】

ERRORTYPE AW MPI\_VO\_SetPubAttr(VO\_DEV VoDev, const VO\_PUB\_ATTR\_S \*pstPubAttr);

### 【参数】

 参数	描述	输入输出
VoDev	需要设置的 VO 设备编号	输入
pstPubAttr	VO 设备属性结构体	输入(动态)

### 【返回值】

	ideri		allinden'i	本。  《
返回值	描述	II.	This.	WIV.
SUCCESS	成功			
错误码	参考 mm_	_comm_vo.h 中的	错误码描述	<b>述。</b> ② 编 <sup>数</sup>
			NE	
o.h o_vo.so		WIL		
J.v0.50	A Levil		e de la companya de l	

### 【需求】

• 头文件: mpi\_vo.h

• 库文件: libmpp vo.so

### 【注意】

● 需要初始化 pstPubAttr 的 enIntfType(显示设备)与 enIntfSync(分辨率以及刷新频率) 成员。

enIntfType: VO\_INTF\_LCD、VO\_INTF\_TV, VO\_INTF\_HDMI、VO\_INTF\_VGA。

enIntfSync:

VO\_OUTPUT\_PAL、VO\_OUTPUT\_NTSC

VO OUTPUT 720P50, VO OUTPUT 720P 60

VO\_OUTPUT\_1080P24、VO\_OUTPUT\_1080P\_25、VO\_OUTPUT\_1080P\_30、 VO\_OUTPUT\_1080P\_50、VO\_OUTPUT\_1080P\_60 等。

### 【举例】

/\* 显示设备切换为HDMI设备,输出格式是1080P/60fps \*/ VO\_PUB\_ATTR\_S stPubAttr;



```
AW_MPI_V0_GetPubAttr(0, &stPubAttr);
stPubAttr.enIntfType = V0_INTF_HDMI;
stPubAttr.enIntfSync = V0_OUTPUT_1080P60;
AW_MPI_V0_SetPubAttr(0, &stPubAttr);
```

# 4.5.4 AW\_MPI\_VO\_GetPubAttr

### 【目的】

获取 VO 设备的背景色、显示设备类型、显示数据格式等信息。

### 【语法】

ERRORTYPE AW\_MPI\_VO\_GetPubAttr(VO\_DEV VoDev, const VO\_PUB\_ATTR\_S \*pstPubAttr);

### 【参数】

参数	描述		输入输出
VoDev	需要获取信息的 VO	设备编号	输入
pstPubAttr	VO 设备属性结构体		输出(动态)

### 【返回值】

### 【需求】

• 头文件: mpi vo.h

• 库文件: libmpp vo.so

### 【注意】

无。

### 【举例】

参见上面一条。



# 4.5.5 AW\_MPI\_VO\_GetHdmiHwMode

### 【目的】

获取接入的 Hdmi 设备显示分辨率信息

### 【语法】

ERRORTYPE AW\_MPI\_VO\_GetHdmiHwMode(VO\_DEV VoDev, VO\_INTF\_SYNC\_E \*mode);

### 【参数】

返回值	描述	输入输出	ક્લ
VoDev	VO 设备	输入	
mode	设备信息	输出(动态属性)	

### 【返回值】

返回值	描述	ANG
SUCCESS	成功	119
错误码	参考 mm_comm_vo	o.h 中的错误码描述。

### 【需求】

• 头文件: mpi\_vo.h

• 库文件: libmpp vo.so

### 【注意】

该函数在没有 HDMI 设备接入的时候会返回 ERR\_VO\_DEV\_NOT\_CONFIG 错误码,如果 Hdmi 设备的显示分辨率不被支持则会返回 ERR VO NOT SUPPORT 错误码。

### 【举例】

```
/* declaration */
int ret = 0;

V0_INTF_SYNC_E mode;

/* enable one layer, chn = 0, lyl = 1 */
ret = AW_MPI_V0_GetHdmiHwMode(0, amode);

If (SUCCESS != ret) {
```



# 4.5.6 AW\_MPI\_VO\_EnableVideoLayer

### 【目的】

申请并使能一个图层,将该图层信息结构体加入到全局链表里面。

### 【语法】

ERRORTYPE AW\_MPI\_VO\_EnableVideoLayer(VO\_LAYER VoLayer);

### 【参数】

返回值	描述	输入输出
VoLayer	图层索引	输入

### 【返回值】

返回值 描述 SUCCESS 成功 错误码 参考 mm comm vo.h 中的错误码描述。

#### 【需求】

• 头文件: mpi vo.h

• 库文件: libmpp\_vo.so

### 【注意】

• VoLayer 要依靠一个宏定义来得到,该宏定义为 #define HLAY(chn, lyl) (chn\*4+lyl),其中 chn 是 VO 设备通道号(channel),lyl 是 layer 的编号(layer\_id)。chn(0-3),lyl(0-3)。

### 【举例】

/\* declaration \*/
int ret = 0;



```
(*) enable one layer, chn = (*) lyl = 1 */
ret = AW_MPI_VO_EnableVideoLayer(HALY(0, 1));
If (SUCCESS != ret) {
return -1;
```

# $4.5.7~AW\_MPI\_VO\_DisableVideoLayer$

## 【目的】

释放指定的图层,从全局链表里面移除该图层信息结构体。

ERRORTYPE AW\_MPI\_VO\_DisableVideoLayer(VO\_LAYER VoLayer); 输入 VoLayer 图层索引

### 【返回值】

返回值 描述 SUCCESS《成功 参考 mm\_comm\_vo.h 中的错误码描述。

### 【需求】

• 头文件: mpi vo.h

• 库文件: libmpp\_vo.so

### 【注意】



```
/* declaration */
int ret = 0;
/* enable one layer, chn = 0, lyl = 1 */
ret = AW_MPI_VO_DisableVideoLayer(HALY(0, 1));
If (SUCCESS != ret) {
  return -1;
}
```

# 4.5.8 AW\_MPI\_VO\_Add@utsideVideoLayer

【目的】

添加一个外部图层(专门用于 GUI)。

【语法】

ERRORTYPE AW MPI VO AddOutsideVideoLayer(VO LAYER VoLayer);

【参数】

返回值 描述 输入输出 VoLayer 图层索引 输入

【返回值】

返回值 描述
SUCCESS 成功
错误码 参考 mm\_comm\_vo.h 中的错误码描述。

### 【需求】

• 头文件: mpi vo.h

• 库文件: libmpp\_vo.so

【注意》

● 该函数专门用于添加 GUI 的界面图层(因为 GUI 图层的申请不经过 MPP\_VO 这个模块, 所以如果用 AW\_MPI\_VO EnableVideoLayer 的话可能会出现因图层重复被申请而导致申请



失败的情况,AW\_MPI\_VO\_AddOutsideVideoLayer 函数没有图层申请这一步骤),特别注意:如果不是用于 GUI 图层的话请使用函数 AW\_MPI\_VO\_EnableVideoLayer 来完成图层的新建。VoLayer 需要 HLAY 宏定义来配合生成。

### 【举例】

```
/* declaration */
int ret = 0;

/* add one UI layer, chn = 1, lyl = 0 */
ret = AW_MPI_VO_AddOutsideVideoLayer(HALY(1, 0));

If (SUCCESS != ret) {
    return -1;
}
```

# 4.5.9 AW\_MPI\_VO\_RemoveOutsideVideoLayer

### 【目的】

移除外部图层(专门用于 GUI)。

### 【语法】

 $ERRORTYPE\ AW\_MPI\_VO\_RemoveOutsideVideoLayer(VO\_LAYER\ VoLayer);$ 

### 【参数】

返回值	描述	输入输出
VoLayer	图层索引	输入

### 【返回值】

返回值	描述
SUCCESS	成功
错误码	参考 mm_comm_vo.h 中的错误码描述。

#### 【需求

• 头文件: mpi vo.h



库文件: libmpp vo.so

【注意】

● 该函数专门用于移除 GUI 的界面图层(因为 GUI 图层的释放不经过 MPP VO 这个模块, 如果用 AW MPI VO DisableVideoLayer 的话可能会出现别的程序正在使用该图层,而 该图层的资源被释放的情况),特别注意:如果不是用于 GUI 图层的话请尽量使用函数 AW\_MPI\_VO\_DisableVideoLayer 来完成图层的移除。VoLayer 需要 HLAY 宏定义来配合 生成。

【举例】

INE RELIGIBLE TO THE REPORT OF THE PARTY OF 4.5.10 AW\_MPI\_VO\_OpenVideoLayer

【目的】

打开一个指定的图层。

【语法】

ERRORTYPE AW MPI VO OpenVideoLayer(VO LAYER VoLayer);

【参数】

返回值 描述 输入输出 VoLayer 图层索引

【返回值】

返回值 描述 **SUCCESS** 成功 错误码 参考 mm comm vo.h 中的错误码描述。

【需求】

头文件: mpi vo.h

库文件: libmpp vo.so



### 【注意】

● 在调用完 AW\_MPI\_VO\_EnableVideoLayer 函数之后图层默认是被关闭的,如果需要在屏幕 上显示出该图层,需要先调用该函数打开图层。

### 【举例】

```
/* declaration */
 int ret = 0;
 /* open one layer, chn = 1, lyl = 0 */
 ret = AW_MPI_V0_OpenVideoLayer(HALY(1, 0));
 If (SUCCESS != ret) {
4.5.11 AW_MPI_VO_CloseVideoLayer
【目的】

关闭一个指字和一
```

### 【语法】

ERRORTYPE AW\_MPI\_VO CloseVideoLayer(VO\_LAYER VoLayer);

### 【参数】

返回值	描述	输入输出
VoLayer	图层索引	输入

### 【返回值】

返回值	描述
SUCCESS	成功
错误码	参考 mm_comm_vo.h 中的错误码描述。



头文件: mpi vo.h

库文件: libmpp vo.so

【注意】

• 如果先调用函数 AW MPI VO DisableVideoLayer,再调用该函数可能会造成花屏现象,花 屏持续的时间取决于两个函数的调用间隔,间隔时间越长,花屏时间就越长。最好还是先调用 该函数关闭图层,然后再 Disable。

【举例】

4.5.12 AW\_MPI\_VO\_SetVideoLayerAttr

【目的】

设置指定图层的属性。

【语法】

The state of the s AW MPI VO SetVideoLayerAttr(VO LAYER ERRORTYPE VoLayer, const VO\_VIDEO\_LAYER\_ATTR\_S \*pstLayerAttr);

### 【参数】

	9	-00
返回值	描述	输入输出
VoLayer	图层索引	输入
pstLayerAttr	图层属性结构体	输入(动态)

### 【返回值】

返回值	描述
SUCCESS	成功
错误码	参考 mm_comm_vo.h 中的错误码描述。

### 【需求】

文件: mpi vo.h

库文件: libmpp vo.so



### 【注意】

• pstLayerAttr 的结构体类型是 VO\_VIDEO\_LAYER\_ATTR\_S,里面包含显示区域、图片大小、图层像素格式等元素,但是本函数只对 VO\_VIDEO\_LAYER\_ATTR\_S 的 stDispRect 成员生效,也就是说本函数只是用于设置图像显示的 X、Y 坐标以及宽和高。

### 【举例】

```
/* declaration */
int ret = 0;

VO_VIDEO_LAYER_ATTR_S stLayerAttr;

/* set layer attribute, chn = 1, lyl = 0 */
stLayerAttr.stDispRect.X = 0;

stLayerAttr.stDispRect.Y = 0;

stLayerAttr.stDispRect.Width = 1280;

stLayerAttr.stDispRect.Height = 720;

ret = AW_MPI_VO_SetVideoLayerAttr(HALY(1, 0), &stLayerAttr);

If (SUCCESS != ret) {
    return -1;
}
```

# 4.5.13 AW MPI\_VO GetVideoLayerAttr

### 【目的】

获取指定图层的属性。

### 【语法】

 $\label{lem:const} ERRORTYPE \quad AW\_MPI\_VO\_GetVideoLayerAttr(VO\_LAYER \quad VoLayer, \quad const \\ VO\_VIDEO\_LAYER\_ATTR\_S *pstLayerAttr);$ 

### 【参数】

返回值	描述	输入输出
VoLayer	图层索引	输入。
pstLayerAttr	图层属性结构体	输出(动态)

版权所有 © 珠海全志科技股份有限公司。保留一切权利



### 【返回值】

A.FA		A-A
返回值	描述	
SUCCESS 错误码		nm_comm_vo.h 中的错误码描述。

### 【需求】

• 头文件: mpi\_vo.h

• 库文件: libmpp\_vo.so

### 【注意】

无。

### 【举例】

```
/* declaration */
int ret = 0;

VO_VIDEO_LAYER_ATTR_S stLayerAttr;

/* set layer attribute, chn = 1, lyl = 0 */
ret = AW_MPI_VO_GetVideoLayerAttr(HALY(1, 0), &stLayerAttr);

If (SUCCESS != ret) {
    return | 1;
}
```

# 4.5.14 AW\_MPI\_VO\_SetVideoLayerPriority

### 【目的】

设置指定图层的显示优先级。

### 【语法】

ERRORTYPE AW\_MPI\_VO\_SetVideoLayerPriority(VO\_LAYER VoLayer, unsigned int uPriority);

#### 【参数】



返回值	描述	输入输出
VoLayer	图层索引	输入
uPriority	优先级	输入(动态)

### 【返回值】

返回值	描述
SUCCESS	成功
错误码	参考 mm_comm_vo.h 中的错误码描述。

头文件: mpi\_vo.h

• 库文件: libmpp\_vo.so

### 【注意】

A State of the sta • 显示的时候越靠近上层的图层其 uPriority 值越大,所以优先级越高,相应的图层就越靠近顶 层。uPriority 的取值范围是 0~15,使用的时候注意不要超过此范围。

### 【举例】

```
/* declaration */
int ret = 0;
unsigned int uPriority = 10
/* set layer priority, chn = 1, lyl = 0 */
If (SUCCESS != ret) {
return -1;
```

# 4.5.15 AW\_MPI\_VO\_GetVideoLayerPriority

### 【自的】

获取指定图层的显示优先级。



### 【语法】

ERRORTYPE AW MPI\_VO\_GetVideoLayerPriority(VO\_LAYER VoLayer, unsigned int \*puPriority);

### 【参数】

返回值	描述	输入输出
VoLayer	图层索引	输入
uPriority	优先级	输出(动态)

### 【返回值】

返回值 描述
SUCCESS 成功
错误码 参考 mm\_comm\_vo.h 中的错误码描述。
h
vo.so

### 【需求】

- 头文件: mpi\_vo.h
- 库文件: libmpp\_vo.so

### 【注意】

无。inde

### 【举例】

```
/* declaration */
int ret = 0;
unsigned int uPriority;

/* get layer priority, chn = 1, lyl = 0 */
ret = AW_MPI_VO_GetVideoLayerAttr(HALY(1, 0), &uPriority);

If (SUCCESS != ret) {
   return -1;
}
```



# 4.5.16 AW\_MPI\_VO\_SetVideoLayerAlpha

### 【目的】

设置指定图层的 alpha,可以理解为透明度。

### 【语法】

 $ERRORTYPE\ AW\_MPI\_VO\_SetVideoLayerAlpha(VO\_LAYER, VO\_VIDEO\_LAYER\_ALPHA\_S *pAlpha);$ 

### 【参数】

返回值	描述	输入输出。
VoLayer	图层索引	输入
pAlpha	alpha 信息结构体	输入 (动态)

### 【返回值】

返回值	描述		
SUCCESS	成功		
错误码	参考 mm_comm_vo.	h 中的错误	码描述。

### 【需求】

- 头文件: mpi vo.h
- 库文件: libmpp\_vo.so

### 【注意】

• pAlpha 结构体有结构体成员 mAlphaMode,为 0 时是点 alpha 模式,此时每个像素都有自己单独的 alpha,可以实现部分区域全透,部分区域半透,部分区域不透的效果;为 1 时是面 alpha 模式,整个图层共用一个 alpha,统一的透明度。需要注意的是 YUV 格式下并不支持使用 alpha,所以该格式下函数调用会失败,只有 ARGB 等带有 alpha 模式的的图像才可以使用。

### 【举例】

/\* declaration \*/
int ret = 0;



```
V0_VIDEO_LAYER_ALPHA_S stAlpha;

stAlpha.mAlphaMode = 1;

stAlpha.mAlphaValue = 128;

/* set layer' s alpha, chn = 1, lyl = 0 */

ret = AW_MPI_V0_SetVideoLayerAlpha(HALY(1, 0), &stAlpha);

If (SUCCESS != ret) {
    return -1;
}
```

# 4.5.17 AW\_MPI\_VO\_GetVideoLayerAlpha

### 【目的】

获取指定图层的 alpha,包括 alpha 模式以及其 value

### 【语法】

 $ERRORTYPE\ AW\_MPI\_VO\_GetVideoLayerAlpha(VO\_LAYER, VO\_VIDEO\_LAYER\_ALPHA\_S *pAlpha);$ 

### 【参数】

返回值描述	输入输出
VoLayer 图层索引	输入 lindell
pAlpha valpha 信息结构体	输出(动态)

### 【返回值】

返回值	描述	Š
SUCCESS	成功	
错误码	参考 mm_comm_vo.h 中的错误码描述。	

### 【需求】

• 头文件: mpi\_vo.h

• 库文件: libmpp\_vo.so

### 【注意】



无。

### 【举例】

```
declaration */
int ret = 0;
V0_VIDE0_LAYER_ALPHA_S stAlpha;
/* set layer's alpha, chn = 1, lyl = 0 */
ret = AW_MPI_V0_GetVideoLayerAlpha(HALY(1, 0), &stAlpha);
If (SUCCESS != ret) {
return -1
                                                 NERMAN
```

# 4.5.18 AW\_MPI\_VO\_EnableChn

### 【目的】

为指定的 VO 图层创建一个指定编号的 VO 组件通道。

### 【语法】

ERRORTYPE AW\_MPI\_VO\_EnableChn(VO\_LAYER VoLayer, VO\_CHN VoChn);

### 【参数】、

1000		<
返回值	描述	输入输出
VoLayer	图层索引	输入
VoChn	通道号	输入

### 【返回值】

返回值	描述
SUCCESS	成功
错误码	参考 mm_comm_vo.h 中的错误码描述。

### 【需求》。

文件: mpi vo.h

库文件: libmpp\_vo.so



### 【注意】

【举例】

```
/* declaration */
int ret = 0;

/* create channel */

ret = AW_MPI_VO_EnableChn(HLAY(0, 1), 0);

if (SUCCESS == ret) {

printf( "create vo channel succeed!\n" );
} else if (ERR_VO_CHN_NOT_DISABLE == ret) {

printf( "this vo channel has exist, find another one. \n" );
} else {

return -1;
}
```

# 4.5.19 AW\_MPI\_VO\_DisableChn

【目的】

销毁指定 VO 图层指定编号的 VO 通道。

【语法】

ERRORTYPE AW\_MPI\_VO\_DisableChn(VO\_LAYER VoLayer, VO\_CHN VoChn);

【参数】

返回值	描述	输入输出
VoLayer	图层索引	输入
VoChn	通道号	输入

### 【返回值】

返回值描述SUCCESS成功错误码参考 mm\_comm\_vo.h 中的错误码描述。



### 【需求】

• 头文件: mpi\_vo.h

• 库文件: libmpp\_vo.so

【注意】

• 该函数会销毁 AW\_MPI\_VO\_EnableChn 的所有资源。

【举例】

# 4.5.20 AW\_MPI\_VO\_RegisterCallback

【目的】

为创建的 VO 通道组件实例注册一个回调函数。

【语法】

THE WAR HELD THE REAL PROPERTY OF THE PARTY ERRORTYPE AW\_MPI\_VO\_RegisterCallback(VO\_LAYER VoLayer, VO CHN VoChn, MPPCallbackInfo \*pCallback);

### 【参数】

		4///
返回值	描述	输入输出
VoLayer	图层索引	输入
VoChn	VO 通道号	输入
pCallback	回调函数信息结构体	输入(静态)

### 【返回值】

返回值	描述
SUCCESS	成功
错误码	参考 mm_comm_vo.h 中的错误码描述。

头文件: mpi\_vo.h



• 库文件: libmpp vo.so

### 【注意】

- 该函数用于 MPP\_VO 模块向应用程序发送事件通知,如果应用程序不需要接收来自 MPP\_VO 模块的事件,那么也可以不注册该函数(但不建议这么做)。pCallback 的 cookie 成员存储 应用程序自定义的结构体数据,callback 成员则需要指向应用程序的回调函数实体。回调函数的参数为 Func(void *cookie*, *MPP\_CHN\_S* pChn, MPP\_EVENT\_TYPE event, void \*pEventData)。
- MPP VO 模块定义的事件有以下类型:

MPP\_EVENT\_RELEASE\_VIDEO\_BUFFER, //VIDEO\_FRAME\_INFO\_S for recorder/WiChannel: DoVdaThread, ISE, VO, VENC

MPP EVENT VENC TIMEOUT, //uint64 t\*

MPP\_EVENT\_RELEASE\_ISE\_VIDEO\_BUFFERO, //VIDEO\_FRAME\_INFO\_S for recorder/VIChannel::DoVdaThread, ISE, VO, VENC

MPP\_EVENT\_RELEASE\_ISE\_VIDEO\_BUFFER1, //VIDEO\_FRAME\_INFO\_S for recorder/VIChannel::DoVdaThread, ISE, VO, VENC

MPP EVENT RELEASE AUDIO BUFFER, //AUDIO FRAME S

MPP EVENT BSFRAME AVAILABLE, //CDXRecorderBsInfo

MPP EVENT ERROR ENCBUFFER OVERFLOW,

MPP\_EVENT\_NEED\_NEXT\_FD, // int muxerId

MPP EVENT RECORD DONE, // int muxerId

MPP EVENT CAPTURE AUDIO DATA, // unsigned int size;

MPP EVENT NOTIFY EOF = 0x100,

MPP EVENT SET VIDEO SIZE, //SIZE S

MPP EVENT RENDERING START,

### 【举例】

/\* declaration \*/
int ret = 0;

MPPCallbackInfo cbInfo;

struct custom\_st\_name stCustom;



```
cbInfo.cookie = (void *)&stEustom;
cbInfo.callback = (MPPCallbackFuncType)&CallbackFuncName;

/* register it */
ret = AW_MPI_VO_RegisterCallback(HLAY(0, 1), 0, &cbInfo);
if (SUCCESS == ret) {
    return -1;
}
```

# 4.5.21 AW\_MPI\_VO\_SetChnDispBufNum

### 【目的】

设置 VO 组件通道实例的缓存 buf 数量。

### 【语法】

ERRORTYPE AW\_MPI\_VO\_SetChnDispBufNum(VO\_LAYER VoLayer, VO\_CHN VoChn, unsigned int uBufNum);

### 【参数】

返回值	描述	输入输出
VoLayer	图层索引	输入 %
VoChn	VO 通道号	输入。
uBufNum	缓存 buf 的数量	输入 (动态)

### 【返回值】

返回值	描述	ŞT
SUCCESS	成功	
错误码	参考 mm_comm_vo.h 中的错误码描述。	

### 【需求】

• 头文件: mpi\_vo.h

• 库文件: libmpp\_vo.so

### 【注意】



ullet 该函数必须被调用,并且 uBufNum 的数量要大于等于 ullet 同时 uBufNum 的数量不能超过 ullet ullet

【举例】

无。

# $4.5.22\ AW\_MPI\_VO\_GetChnDispBufNum$

【目的】

获取 VO 组件通道实例的缓存 buf 数量。

《语法】

ERRORTYPE AW\_MPI\_VO\_GetChnDispBufNum(VO\_LAYER VoLayer, VO\_CHN VoChn, unsigned int uBufNum);

### 【参数】

返回值	描述	输入输出
VoLayer	图层索引	输入
VoChn	VO 通道号	输入
uBufNum	缓存 buf 的数量	输出(动态)

### 【返回值】

返回值	描述	A THE IV
SUCCESS	成功	
错误码	参考 mm	_comm_vo.h 中的错误码描述。

【需求】

• 头文件: mpi\_vo.h

• 库文件: libmpp vo.so

【注意】

无。

【举例】



无。

# 4.5.23 AW\_MPI\_VO\_GetDisplaySize

【目的】

获取 VO 组件通道实例图层的显示数据(图层的显示宽、高)。

【语法】

ERRORTYPE AW\_MPI\_VO\_GetDisplaySize(VO\_LAYER VoLayer, VO\_CHN VoChn, SIZE\_S \*pDisplaySize);

### 【参数】

- CKA		- CAT		
返回值	描述	输入输出		
VoLayer	图层索引	输入		
VoChn	VO 通道号	输入		
pDisplaySize	size 信息结构体	输出(动态)		

### 【返回值】

返回值 描述 SUCCESS 成功 错误码 参考 mm\_comm\_vo.h 中的错误码描述。

### 【需求】

• 头文件: mpi\_vo.h\_

• 库文件: libmpp\_vo.so

### 【注意】

• pDisplaySize 结构体成员的 Width 代表宽度,Height 代表高度。

#### 【举例】

<del>T</del>



# 4.5.24 AW\_MPI\_VO\_StartChn

### 【目的】

通道开始工作(VO 组件状态设为 StateExecuting),接收视频数据并送去显示。

### 【语法】

ERRORTYPE AW\_MPI\_VO\_StartChn(VO\_LAYER VoLayer, VO\_CHN VoChn);

### 【参数】

返回值	描述	输入输出	
VoLayer	图层索引	输入	
VoChn	VO 通道号	输入	

### 【返回值】

返回值	描述	**************************************	. 1	W. W.
SUCCESS	成功			6
错误码	参考 m	m_comm_vo.h	中的错误	码描述。

### 【需求】

- 头文件: mpi\_vo.h
- 库文件: libmpp vo.so

### 【注意】

• 要在通道使能后,注册完回调函数之后再调用该函数。

### 【举例】

无。

# $4.5.25~AW\_MPI\_VO\_StopChn$

### 【目的】

通道停止(VO 组件状态设为 StateIdle),停止数据传输。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

1



### 【语法】

ERRORTYPE AW\_MPI\_VO\_StopChn(VO\_LAYER VoLayer, VO\_CHN VoChn);

【参数】

描述	输入输出
图层索引	 输入
VO 通道号	输入
	图层索引

### 【返回值】

返回值 描述 参考 mm\_comm\_vo.h.中的错误码描述。

【需求】

• 头文件: mpi\_vo.h

• 库文件: libmpp vo.so

【注意】

• 停止数据传输之后并不会释放相关的资源,可以再次从停止状态恢复。

【举例】

无。

# 4.5.26 AW\_MPI\_VO\_PauseChn

【目的】

通道暂停(设置组件状态为 StatePause),停止显示。

【语法】

ERRORTYPE AW\_MPI\_VO\_PauseChn(VO\_LAYER VoLayer, VO\_CHN VoChn);



返回值 描述 输入输出 图层索引 VoLayer 输入 VoChn VO 通道号 输入

【返回值】

返回值 描述 **SUCCESS** 成功 错误码 参考 mm\_comm\_vo.h 中的错误码描述。

头文件: mpi vo.h

库文件: libmpp vo.so

【注意】

无。

【举例】

无。

Report of the state of the stat 4.5.27 AW\_MPI\_VO\_ResumeChn

【目的】

通道恢复(设置 VO 组件状态为 StateExecuting),开始数据传输。

【语法】

ERRORTYPE AW\_MPI\_VO\_ResumeChn(VO\_LAYER VoLayer, VO\_CHN VoChn);

【参数】

返回值 描述 输入输出 VoLayer 图层索引 输入 VoChn VO 通道号 输入

【返回值】



返回值 描述 SUCCESS 成功 错误码 参考 mm comm vo.h 中的错误码描述。

【需求】

• 头文件: mpi\_vo.h

• 库文件: libmpp\_vo.so

【注意】

White Life History Harden Committee 根据 openMAX 标准的说法,各种状态的定义如下所示

StateIdle:有资源,不传输数据。

StateExecuting: 有资源,传输数据,处理数据。

State Pause: 有资源,传输数据,不处理数据。

三个状态两两可以相互转换。

【举例】

无。

4.5.28 AW\_MPI\_VO\_Seek

【目的】

跳转播放。

【语法】

ERRORTYPE AW MPI VO Seek(VO LAYER VoLayer, VO CHN VoChn);

返回值	描述	输入输出
VoLayer	图层索引	输入
VoChn	VO 通道号	输入



### 【返回值】

返回值	描述	the state of the s
SUCCESS 错误码		m_comm_vo.h 中的错误码描述。

### 【需求】

• 头文件: mpi\_vo.h

• 库文件: libmpp vo.so

### 【注意】

• MPP\_VO 模块的跳转播放不是真正的"跳转",真正的跳转在视频格式解析模块里面(可以跳转到指定的时间播放视频),该模块的跳转函数只是负责在设置跳转之后将跳转之前缓存到的视频帧丢弃,否则会出现跳转之后播放到跳转前视频帧的情况。最好在跳播之后调用该函数。举例:现在播放到视频的第 100ms,此时 MPP\_VO 模块可能缓存了 100ms 之后的几帧视频,然后同时发生了跳转事件,视频直接从 100ms 跳转到了 2000ms 处播放,那么需要调用该函数清除 100ms 之后的几帧缓存视频数据,从而保证跳转播放的连贯性。

### 【举例】

无。

# 4.5.29 AW\_MPI\_VO\_SetStreamEof

### 【目的】

根据参数标记视频流的结束(此时会停止视频的显示),状态转为 State Idle;或者取消视频流结束标记。

### 【语法】

ERRORTYPE AW\_MPI\_VO\_SetStreamEof(VO\_LAYER VoLayer, VO\_CHN VoChn, BOOL bEofFlag);

	返回值	描述	输入输出	
	VoLayer	图层索引	输入	.16
ć	VoChn	VO 通道号	输入	W. XXX
8	bEofFlag	视频流结束标志	输入	H-HI



### 【返回值】

返回值	描述	H. H. C.
SUCCESS 错误码		mm comm vo.h 中的错误码描述。
10 1/V 1-J	シコュ	

【需求】

• 头文件: mpi\_vo.h

• 库文件: libmpp\_vo.so

【注意】

4为1年 bEofFlag 为 1 时表示视频流结束,为 0 时取消视频结束标志。当设置为 1 时 VO 组件的状态 会转为 StateIdle,此时视频流会停止传输。

【举例】

无。

# 4.5.30 AW\_MPI\_VO ShowChn

【目的】

通道图层显示(去隐藏化)

【语法】

ERRORTYPE AW\_MPI\_VO\_ShowChn(VO\_LAYER VoLayer, VO\_CHN VoChn);

【参数】

返回值	描述	输入输出
VoLayer	图层索引	输入
VoChn	VO 通道号	输入

【返回值】



返回值	描述	The state of the s
- XX	лш ~	
SUCCESS	成功	A STATE OF THE STA
错误码	参考:	mm_comm_vo.h 中的错误码描述。
-1300		T1X94

【需求】

• 头文件: mpi\_vo.h

• 库文件: libmpp\_vo.so

【注意】

如果通道本来就是处于 show 状态,则不会有任何的变化, 4.5.31 AW\_MPI\_VO\_HideChn
【目的】
通道图层障壁

【语法】

ERRORTYPE AW\_MPI\_VO\_HideChn(VO\_LAYER VoLayer, VO\_CHN VoChn);

【参数】

返回值	描述	输入输出
VoLayer	图层索引	输入
VoChn	VO 通道号	输入

### 【返回值】

返回值	描述。	nindeni
SUCCESS	成功	W/A
错误码	。 参考 mm_comn	n_vo.h、中的错误码描述。



### 【需求】

• 头文件: mpi\_vo.h;

• 库文件: libmpp\_vo.so

### 【注意】

● 通道隐藏只是在显示屏幕上面变得不可见,并不会销毁通道相关的任何资源,调用 AW\_MPI\_VO\_ShowChn 之后通道对应的图层在显示屏幕上面会重新变为可见状态。

NERRITATION

【举例】

泛。

# 4.5.32 AW\_MPI\_VO\_GetChnPts

### 【目的】

获取 VO 组件通道实例视频播放的时间戳,单位 us

### 【语法】

ERRORTYPE AW\_MPI\_VO\_GetChnPts(VO\_LAYER VoLayer, VO\_CHN VoChn, uint64\_t \*pChnPts);

### 【参数】

返回值	描述	输入输出
VoLayer	图层索引	输入
VoChn	VO 通道号	输入
pChnPts	时间戳(指针类型)	输出(动态)

### 【返回值】

返回值	描述
SUCCESS	成功
错误码	参考 mm_comm_vo.h 中的错误码描述。

### 【季求】



• 头文件: mpi\_vo.h

▶ 库文件: libmpp\_vo.so

【注意】

无。

【举例】

无。

# 4.5.33 AW\_MPI\_VO\_SendFrame

【目的】

发送一帧视频数据送去显示。

【语法】

ERRORTYPE AW\_MPI\_VO\_SendFrame(VO\_LAYER VoLayer, VO\_CHN VoChn, VIDEO\_FRAME\_INDO\_S \*pstFrame, int nMilliSec);

### 【参数】

返回值	描述	输入输出
VoLayer	图层索引	输入
VoChn	VO 通道号	输入 %
pstFrame	视频帧信息结构体	输入(动态)
nMilliSec	等待时间,单位 ms	输入

### 【返回值】

返回值	描述	7.
SUCCESS	成功	
错误码	参考 mm_comm_vo.h 中的错误码描述。	

### 【需求】

• 头文件: mpi\_vo.h

• 库文件: libmpp vo.so

【注意】



- pstFrame 结构体里面包含视频帧 id,视频帧大小、像素格式、物理地址、虚拟地址、时间戳等等。该函数一般用在非绑定的 VO 组件中,此时视频数据是从外部文件中读取到的,必须设置视频帧的 id、宽度、高度、像素格式、物理地址、虚拟地址、时间戳。
- nMilliSec 参数代表等待时间,如果超过该时间,视频还没有被成功发送。那么该帧视频数据 就会被丢弃。

【举例】

无。

## 4.5.34 AW MPI VO Debug StoreFrame

【目的】

保存一帧视频数据,可用于截图。

【语法】

ERRORTYPE AW\_MPI\_VO\_Debug\_StoreFrame(VO\_LAYER VoLayer, VO\_CHN VoChn, unit64\_t framePts);

### 【参数】

返回值	描述	输入输出
VoLayer	图层索引	输入
VoChn	VO 通道号	输入
framePts	时间戳	输入

### 【返回值】

返回值	描述		V.
SUCCESS	成功	4.	у,
错误码	参考 r	nm_comm_vo.h 中的错误码描述。	٥

【需求】

• 头文件: mpi\_vo.h

• 库文件: libmpp vo.so

【注意】



该函数可用于保存一帧视频数据到文件里面,framePts 参数指定了需要保存的视频数据所在的 位置(以时间为坐标,单位 ms)。如果找不到指定时间的帧,那么程序就会选择最接近的帧进 行保存。

【举例】

无。

# 4.6 数据结构

4.6.1 VO\_CHN\_ATTR\_S

### **●PROTOTYPE**

typedef struct VO\_CHN\_ATTR\_S

\*/ New York of the state of the unsigned int mPriority; /\* video out overlay pri \*/

RECT S mRect; /\* rect of video out chn \*

BOOL mbDeflicker; /\* deflicker or not \*/

**}VO CHN ATTR S**;

### •MEMBERS

mPriority: 通道优先级

mRect: 通道视频显示的范围(X、Y坐标,宽高)

mbDeflicker:

### **DESCRIPTION**

VO CHN ATTR S: 用于设置通道的属性

# 4.6.2 VO\_PUB\_ATTR\_S

### **●PROTOTYPE**

typedef struct VO PUB ATTR \$



unsigned int mBgColor; /\* Background color of a device, in RGB format. \*/

VO INTF TYPE E enIntfType; /\* Type of a VO interface \*/

VO\_INTF\_SYNC\_E enIntfSync; /\* Type of a VO interface timing \*/

VO\_SYNC\_INFO\_S stSyncInfo; /\* Information about VO interface timings \*/

} VO PUB ATTR S;

### •MEMBERS

mBgColor:设备背景色,RGB格式

enIntfType: VO 显示设备结口类型

enIntfSync: 显示分辨率以及频率

stSyncInfo: VO 接口的时间信息

/\* VO 显示设备接口类型(部分)\*/

VO INTF CVBS : CVBS 接口

VO INTF YPBPR: YPBPR接口

VO INTF VGA: VGA接口

VO\_INTF\_BT656: BT656接口

VO INTF HDMI: HDMI接口

VO\_INTF\_LCD: LCD 接口

/\* 显示分辨率以及频率(部分)\*/

VO OUTPUT PAL

VO OUTPUT NTSC

VO OUTPUT 1080P24,

VO OUTPUT 1080P25,

VO OUTPUT 1080P30,

VO OUTPUT 720P50,

VO\_OUTPUT\_720P60,

VO\_OUTPUT\_1080P50,

VO\_OUTPUT\_1080P60,

Jani

EN THE PARTY OF TH

反权所有 © 珠海全志科技股份有限公司。保留一切权利



VO OUTPUT 800x600 60, /\* VESA 800 x 600 at 60 Hz (non-interlaced) \*/

VO OUTPUT 1024x768 60, /\* VESA 1024 x 768 at 60 Hz (non-interlaced) \*/

VO OUTPUT 1280x1024 60, /\* VESA 1280 x 1024 at 60 Hz (non-interlaced) \*/

VO OUTPUT 1366x768 60, /\* VESA 1366 x 768 at 60 Hz (non-interlaced) \*/

VO OUTPUT 1440x900 60, /\* VESA 1440 x 900 at 60 Hz (non-interlaced) CVT Compliant \*/

VO OUTPUT 1920x1200 60, /\* VESA 1920 x 1600 at 60 Hz (non-interlaced) CVT (Reduced Blanking)\*/

### DESCRIPTION

NERRINA VO PUB ATTR S:设置显示通道的设备接口类型、分

# 4.6.3 VO VIDEO LAYER ATTR

### **PROTOTYPE**

typedef struct VO VIDEO LAYER ATTR S

RECT S stDispRect; /\* Display resolution \*/

SIZE S stImageSize; /\* Canvas size of the video layer \*/

unsigned int mDispFrmRt; /\* Display frame rate \*/

PIXEL FORMAT E enPixFormat; /\* Pixel format of the video layer \*/

BOOL bDoubleFrame; /\* Whether to double frames \*/

BOOL bClusterMode; /\* Whether to take Cluster way to use memory\*/

} VO VIDEO LAYER ATTR S;

### •MEMBERS

stDispRect: 图层的范围(X、Y坐标,宽高)

stImageSize: 图层画布大小

mDispFrmRt: 显示频率

enPixFormat: 图层像素格式



bDoubleFrame: 视频层倍帧开关。

bClusterMode: 视频层内存聚集使能开关。

### **•DESCRIPTION**

VO VIDEO LAYER ATTR S: 描述显示图层的属性

# 4.6.4 VO\_VIDEO\_LAYER\_ALPHA\_S

### **●PROTOTYPE**

```
alr
typedef struct VO VIDEO LAYER ALPHA S
unsigned char mAlphaMode; /* 0: Pixel Mode, 1: Global Mode */
unsigned char mAlphaValue;
```

**}VO VIDEO LAYER ALPHA S;** 

### •MEMBERS

mAlphaMode: 图层的 alpha 模式, 0: 点 alpha; 1: 面 alpha。参见 alpha 解释

mAlphaValue: 图层 alpha 的值

### **•DESCRIPTION**

VO VIDEO LAYER ALPHA S: 描述图层的 alpha 属性

# 4.6.5 VIDEO\_FRAME\_INFO\_S

### **●PROTOTYPE** ∜

```
typedef struct VIDEO FRAME INFO S
{
VIDEO FRAME S VFrame;
```

unsigned int mId; //id identify frame uniquely

} VIDEO FRAME\_INFO\_S;

### **●MEMBERS**

VFrame: 帧结构体描述

mId:帧ID

### **•DESCRIPTION**

VIDEO FRAME INFO S: 用于描述一帧视频数据

# 4.6.6 VIDEO\_FRAME\_S

### **●PROTOTYPE**

```
C Real Little Heart Little Committee of the Committee of 
typedef struct VIDEO FRAME S
unsigned int mWidth;
unsigned int mHeight;
VIDEO FIELD E mField;
PIXEL FORMAT E mPixelFormat;
VIDEO FORMAT E mVideoFormat;
COMPRESS MODE E mCompressMode;
unsigned int mPhyAddr[3];// Y, U, V; Y, UV; Y, VU
void* mpVirAddr[3];
unsigned int mStride[3];
unsigned int mHeaderPhyAddr[3];
void* mpHeaderVirAddr[3];
unsigned int mHeaderStride[3];
short mOffsetTop; /* top offset of show area */
short mOffsetBottom; /* bottom offset of show area */
short mOffsetLeft; /* left offset of show area */
short mOffsetRight; /* right offset of show area */
uint64_t mpts; //unit:us
```



unsigned int mExposureTime; /\* every frame exp time \*/

unsigned int mFramecat; /\* rename mPrivateData to Framecat exp start \*

int mEnvLV; //environment luminance value

unsigned int mWhoSetFlag;

uint64 t mFlagPts;

unsigned int mFrmFlag;

} VIDEO FRAME S;

### ●MEMBERS

mWidth: 宽

mHeight:高

mField: 视频所在的域

mPixelFormat: 像素格式

mVideoFormat: 视频格式

mCompressMode: 压缩模式

C. Refill the Refill t mPhyAddr:物理地址,按照Y、U、V或者Y、UV或者Y、VU的顺序排列

mpVirAddr: 虚拟地址,对应物理地址

### • DESCRIPTION

VIDEO FRAME S: 描述视频帧的大小,位置等信息。

# 4.6.7 VIDEO FIELD\_E

### •PROTOTYPE

```
typedef enum VIDEO FIELD E
```

{

VIDEO FIELD TOP = 0x1, /\* even field \*/

VIDEO FIELD BOTTOM = 0x2, /\* odd field \*/

VIDEO FIELD\_INTERLACED = 0x3, /\* two interlaced fields \*/



VIDEO\_FIELD\_FRAME = 0x4, /\* frame \*/

VIDEO FIELD BUTT

} VIDEO\_FIELD\_E;

### •MEMBERS

VIDEO FIELD TOP: 偶数行

VIDEO FIELD BOTTOM: 奇数行

VIDEO FIELD INTERLACED:隔行扫描

VIDEO\_FIELD\_FRAME: 帧

### **•DESCRIPTION**

VIDEO\_FIELD\_E: 描述视频帧在一副画面中所在的位置

# 4.6.8 VIDEO\_FORMAT\_E

### •PROTOTYPE

typedef enum VIDEO FORMAT E

{

Fr. VIDEO FORMAT LINEAR = 0x0, /\* nature video line \*/

VIDEO FORMAT TILE = 0x1, /\* tile cell: 256pixel x 16line, default tile mode \*/

VIDEO\_FORMAT\_TILE64 = 0x2, /\* tile cell: 64pixel x 16line \*/

VIDEO FORMAT BUTT

} VIDEO FORMAT E;

### **•**MEMBERS

VIDEO FORMAT LINEAR: 正常的视频行

VIDEO FORMAT TILE:单位: 256 像素 \*16 行

VIDEO FORMAT TILE64:单位:64 像素 \*16 行

### • DESCRIPTION

VIDEO\_FORMAT\_E: 描述视频的格式。



# 4.6.9 COMPRESS\_MODE\_E

### **●PROTOTYPE**

typedef enum COMPRESS MODE E

{

COMPRESS\_MODE\_NONE = 0x0, /\* no compress \*/

COMPRESS\_MODE\_SEG = 0x1, /\* compress unit is 256 bytes as a segment, default seg mode \*/

INER

COMPRESS\_MODE\_SEG128 = 0x2, /\* compress unit is 128 bytes as a segment \*/

COMPRESS MODE LINE = 0x3, /\* compress unit is the whole line \*/

COMPRESS MODE FRAME = 0x4, /\* compress unit is the whole frame \*/

COMPRESS MODE BUTT

} COMPRESS\_MODE\_E;

### **MEMBERS**

COMPRESS MODE NONE: 不压缩

COMPRESS MODE SEG:压缩单位: 256 个字节为一段,默认的段压缩模式

COMPRESS MODE SEG128: 压缩单位: 128 个字节为一段

COMPRESS MODE LINE:压缩单位:一整行压缩

COMPRESS MODE FRAME: 压缩单位: 完整的一帧

### •DESCRIPTION

COMPRESS MODE E: 表明视频帧的压缩方法。

# 4.7 错误码

错误码	宏定义		描述
0xA00F8012	ERR_VO_BUSY		VO 正忙
0xA00F800C	ERR_VO_NO_MEM	adeni	没有足够的内存
0xA00F8006	ERR_VO_NULL_PTR		空指针
0xA00F8010	ERR_VO_SYS_NOTREADY		VO 没有准备好
0xA00F8001	ERR_VO_INVALID_DEVID	XXXXX	无效的 VO 设备 ID
<u></u>	4		

版权所有 © 珠海全志科技股份有限公司。保留一切权利

108



ALLWIMER	Blinderi Blinderi	文档密级、秘密
错误码	宏定义	描述
0xA00F8002	ERR_VO INVALID_CHNID	无效的通道 ID
0xA00F8003	ERR_VO_ILLEGAL_PARAM	非法参数
0xA00F8008	ERR_VO_NOT_SUPPORT	· * 不支持
0xA00F8009	ERR_VO_NOT_PERMIT	没有权限,不允许
0xA00F806C	ERR_VO_INVALID_WBCID	
0xA00F806D	ERR_VO_INVALID_LAYERID	非法的 Layer ID
0xA00F8040	ERR VO DEV NOT CONFIG	VO 设备没有被配置
0xA00F8041	ERR VO DEV NOT ENABLE	VO 设备没有使能
0xA00F8042	ERR VO DEV HAS ENABLED	VO 设备已经使能
0xA00F8043	ERR VO DEV HAS BINDED	VO 设备已经被绑定
0xA00F8044	ERR_VO_DEV_NOT_BINDED	VO 设备没有被绑定
0xA00F8045	ERR_VO_VIDEO NOT_ENABLE	视频没有使能力
0xA00F8046	ERR VO VIDEO NOT DISABLE	视频处于使能状态
0xA00F8047	ERR_VO_VIDEO_NOT_CONFIG	视频没有配置
0xA00F806E	ERR_VO_VIDEO_HAS_BINDED	视频已经绑定
0xA00F806F	ERR_VO_VIDEO_NOT_BINDED	视频没有绑定
0xA00F8048	ERR_VO_CHN_NOT_DISABLE	VO 通道处于使能状态
0xA00F8049	ERR_VO_CHN_NOT_ENABLE	VO 通道没有使能
0xA00F804A	ERR VO CHN NOT CONFIG	VO 通道没有被配置
0xA00F804B	ERR VO CHN NOT ALLOC	VO 通道没有分配
0xA00F806B	ERR_VO_CHN_AREA_OVERLAP	VO 通道区域重叠
0xA00F8014	ERR VO CHN SAMESTATE	同样的状态,错误常见于状态转换
0xA00F8015	ERR_VO_CHN_INVALIDSTATE	非法的状态
0xA00F8016	ERR VO CHN INCORRECT STATE TRANSITION	错误的状态转换
0xA00F8017	ERR_VO_CHN_INCORRECT_STATE_OPERATION	错误的状态行为。
0xA00F804C		无效的样式
0xA00F804D	NQ1."	无效级联位置(例如:组件通道绑
0xA00F804E	ERR VO WAIT TIMEOUT	等待超时
0xA00F804F	ERR_VO_INVALID_VFRAME	非法的视频帧
0xA00F8050	ERR VO INVALID RECT PARA	非法的矩形参数(rect)
0xA00F8051	ERR VO SETBEGIN ALREADY	已经设置为开始
0xA00F8052	ERR VO SETBEGIN NOTYET	还没有设置开始
0xA00F8053	ERR VO SETEND ALREADY	已经设置为结束
0xA00F8054	ERR VO SETEND NOTYET	还没有设置结束
	ERR VO GRP INVALID ID	
	ERR_VO_GRP_NOT_CREATE	
	ERR_VO_GRP_HAS_CREATED	
À	ERR_VO_GRP_NOT_DESTROY	<i>i</i> .
minder.	ERR_VO_GRP_CHN_FULL	alinder.
Hall Blinderi	ERR_VO_GRP_CHN_EMPTY	Walling Seri
X4 (182)	ERR_VO_GRP_CHN_NOT_EMPTY	



- A I V	AND THE PERSON NAMED IN COLUMN TO TH	A IV	A IV	
错误码	宏定义		描述	
A STATE OF THE STA	ERR_VO_GRP_INVALID_SYN_	MODE	WE WELL	, 450
The state of the s	ERR_VO_GRP_INVALID_BASE	PTS		1/2/2
K. Comments	ERR_VO_GRP_NOT_START		***************************************	13/4 /3.
	ERR_VO_GRP_NOT_STOP			
	ERR_VO_GRP_INVALID_FRMI	RATE		
	ERR_VO_GRP_CHN_HAS_REC	;		
	ERR_VO_GRP_CHN_NOT_REC	3		
	ERR_VO_GRP_CHN_NOT_UN	REG		
	ERR_VO_GRP_BASE_NOT_CF	G		
0xA00F8065	ERR_VO_GFX_NOT_DISABLE		图形层处于使能状态	
0xA00F8066	ERR_VO_GFX_NOT_BIND	aleri	图形层没有绑定	
0xA00F8067	ERR_VO_GFX_NOT_UNBIND		图形层没有解绑	
0xA00F8068	ERR_VO_GFX_INVALID_ID	A STATE OF THE STA	非法的图形层 ID	

図形层处于使能状态 図形层没有绑定 図形层没有解绑 非法的图形层 ID

版权所有 © 珠海全志科技股份有限公司。保留一切权利 110

Will Little White High State of the State of



# 5

# 图像拼接

# 5.1 概述

### 5.1.1 ISE 简介

ISE 模块,该模块主要包括三个模式,单目鱼眼模式 (ISEMODE\_ONE\_FISHEYE) 、双目鱼眼模式 (ISEMODE TWO FISHEYE) 、双目拼接模式 (ISEMODE TWO ISH) 。

单目鱼眼模式是对单路鱼眼镜头图像进行校正,支持包括 360 全景左右展开(WARP\_PANO360)、360 全景上下展开(WARP\_180WITH2)、180 度展开(WARP\_PANO180)和 Normal(WARP\_NORMAL)四种校正模式,同时支持对单路普通镜头图像进行畸变校正(WARP\_UNDISTORT);

双目鱼眼模式主要作用是将两路鱼眼镜头图像拼接成一路 360 度全景图像输出;

双目广角拼接模式主要作用是将两路普通镜头的图片拼接成一路广角图像输出。

V459 只支持单目鱼眼模式。

ISE 模块的输入源包括以下二类:

1 用户态读取图像文件向 ISE 组件发送数据;

l 视频输入模块(VI 模块)采集的图像直接发送到 ISE 组件。

# 5.1.2 功能框图

ISE 组件功能框图如图 4-1 所示,ISE 组件接收外部原始图像数据,而不关心图像数据是来自哪个外部模块。由于硬件设计的原因,一个 ISE 组件对应一个 Group,一个 Group 最多可以创建 4 个 Port,其中 Port 0 为 ISE 硬件模块处理后输出的图像,Port 1 ~ Port 3 是对 Port 0 输出 的图像进行缩放而得到的。Group 创建完后必须创建 Port 0,Port 1 ~ Port 3 可以根据实际需要依次创建,Port 1 ~ Port 3 支持无极缩放,图像缩放的范围为 Port 0 宽高的 1/8 ~ 1,各个 Port 2 问相互独立,互不影响。V459 仅有一个 Port,不支持缩放。

ISE 组件输入端的数据流来源有三种,分别是 VI 组件与 ISE 组件通过绑定方式传递 YUV 数据,VI 组件与 ISE 组件通过非绑定方式传递 YUV 数据,以及用户态读取图像文件(Image File)向 ISE 组件传递 YUV 数据。

版权所有 © 珠海全志科技股份有限公司。保留一切权利



ISE 组件输出端的数据流获取有三种,当 ISE 组件输入端(如 VI 组件)通过绑定方式向 ISE 组件传递数据时,输出端(如 VENC 组件或 VO 组件)可以通过绑定方式或者非绑定式获取 ISE 组件输出的图像,当使用绑定方式将 Port 0 与 VO/VENC 组件绑定后,同一个 Group 号的 ISE 组件的 Port 1 ~ Port 3 必须使用绑定方式获取 ISE 输出的图像,同理使用非绑定方式的时候也要遵循上述的规则;用户也可以通过调用 ISE 组件 API 获取 ISE 组件输出端的数据流。

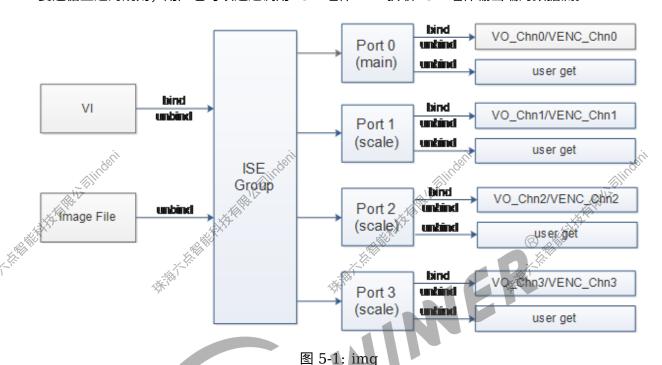


图 4-1 ISE 功能框图

# 5.1.3 ISE 组件 buffer 管理和使用

ISE 组件包含输入和输出 buffer,使用/处理情况如下:

- (1) 对于输入 buffer(待处理 YUV 图像): ISE 组件本身不对外提供输入的 buffer,在非绑定方式下,需要输入端根据输入图像大小开辟 buffer 供 ISE 组件使用,通过调用 AW\_MPI\_ISE\_SendPic()向 ISE 组件传递输入 buffer, ISE 组件内部会对输入的 buffer 进行管理,在使用完输入 buffer 后,ISE 组件会通过事件回调函数 AW\_MPI\_ISE\_RegisterCallback返回事件 MPP\_EVENT\_RELEASE\_ISE\_VIDEO\_BUFFER,通知 ISE 组件输入端数据已经被处理,调用者收到此事件后可以自行对 frame buffer 进行释放等操作。在绑定方式下,由 VI 组件跟 ISE 组件内部自行完成 buffer 的操作交互,不需要调用者再额外处理。
- (2) 对于输出 buffer(处理后 YUV 图像): 该 buffer 由 ISE 组件内部根据输出图像的大小提供,默认为 5 个 buffer。在非绑定方式下,调用者通过调用 AW\_MPI\_ISE\_GetData() 接口获取 ISE 组件输出 buffer,在使用完输出 buffer 后必须及时调用 AW\_MPI\_ISE\_ReleaseData()将 buffer 归还给 ISE 组件。在绑定方式下,由 ISE 组件跟 VENC 组件或 VO 组件内部自行完成 buffer 的操作交互,不需要调用者再额外处理。

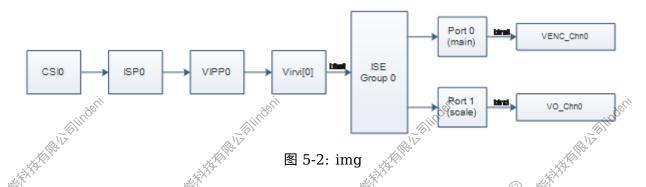
版权所有 © 珠海全志科技股份有限公司。保留一切权利



# 5.1.4 ISE 组件典型应用场景

下面根据典型的应用场景对 ISE 组件的使用进行描述

如图 4-2 所示,该场景是单目鱼眼 360 全景左右展开,创建一个物理设备 VIPP 0 和一个虚通道 Virvi0,将 VI 组件与 ISE 组件输入端进行绑定,ISE 组件输出端分别与编码通道和视频输出通 道进行绑定。



### 图 4-2 单目鱼眼 360 全景左右展开功能框图

如图 4-3 所示,该场景是单目鱼眼 PTZ,创建一个物理设备 VIPP 0 和三个虚通道 Virvi 0 ~ Virvi 2,将 VI 组件与三个 ISE 组件输入端进行绑定,每个 ISE 组件输出端分别与编码通道和视频输出通道进行绑定。

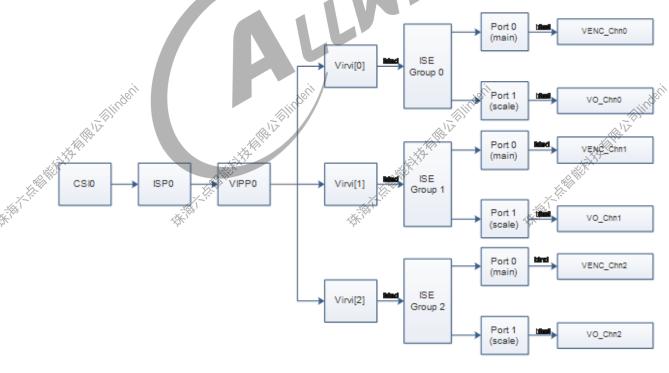


图 5-3: img

图 4-3 单目鱼眼 PTZ 功能框图

版权所有 © 珠海全志科技股份有限公司。保留一切权利

113



如图 4-4 所示, 该场景是双目鱼眼拼接, 创建两个物理设备 VIPP 0 和 VIPP 1, 每个物理设备下 创建一个虚通道 Virvi 0,ISE 组件输出端分别与编码通道和视频输出通道进行绑定。

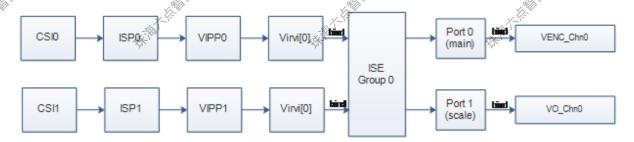


图 5-4: img

### 图 4-4 双目鱼眼拼接功能框图

如图 4-5 所示,该场景是双目广角拼接,创建两个物理设备 VIPP 0 和 VIPP 1,每个物理设备下 创建一个虚通道 Virvi 0,ISE 组件输出端分别与编码通道和视频输出通道进行绑定。

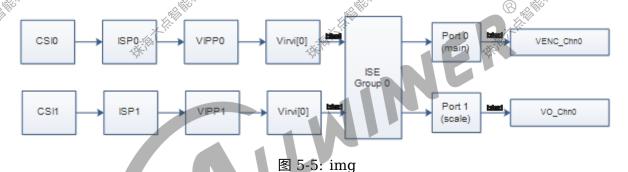


图 4-5 双目广角拼接功能框图

# 5.2 状态转换与 API 接口

## 5.2.1 状态图

操機子機構推構推構推開

每个 ISE 组件内部都有一个线程,其内部按状态机的方式工作,状态转化如图 4-6 所示:

版权所有 © 珠海全志科技股份有限公司。保留一切权利

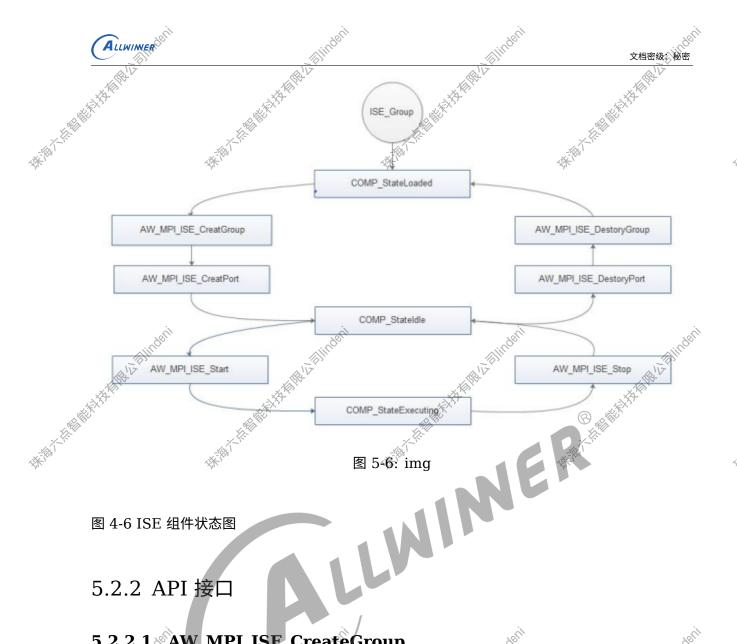


图 4-6 ISE 组件状态图

# 5.2.2 API 接口

# 5.2.2.1 AW\_MPI\_ISE\_CreateGroup

【目的】

创建 ISE 组,选择 ISE 工作模式

【语法】

ERRORTYPE AW MPI ISE CreateGroup(ISE GRP IseGrp, ISE GROUP ATTR S \*pGrpAttr);

### 【参数】

参数	描述	输入/输出
IseGrp pGrpAttr	组 ID 号 组属性,选择 ISE 组件工作模式	输入
	Alv Alv	7,

【返回值】



A11/		A 11
返回值	描述	
0	成功	W. W
非 0	失败,	其值见错误码
	V. (*	

【需求】

头文件: mpi ise.h

库文件: libmpp ise.so

【注意】

目前只支持单目鱼眼、双目鱼眼两种模式下创建多个 Group, 双目拼接模式下只允许创建 PROPERTY AND ASSESSED OF THE PROPERTY ASSESSED. Group

【举例】

无

### AW\_MPI\_ISE\_DestroyGroup **5.2.2.2**

【目的】

销毁 ISE 组

【语法】

ERRORTYPE AW\_MPI\_ISE\_DestroyGroup(ISE\_GRP IseGrp);

【参数】

参数	描述	输入/输出
IseGrp	组 ID 号	输入

【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码

【需求】

文文件: mpi\_ise.h



库文件: libmpp\_ise.so

【注意】

无

【举例】

无

### 5.2.2.3 AW\_MPI\_ISE\_SetGrpAttr

【目的】

设置 ISE 组的属性

【语法】

ERRORTYPE AW\_MPI\_ISE\_SetGrpAttr(ISE\_GRP IseGrp, ISE\_GROUP\_ATTR\_S \*pGr-pAttr);

【参数】

参数	描述	- 1AII	输入/输出
IseGrp	组ID号		———— 输入
pGrpAttr	组属性,说	选择 ISE 组件工作模式	输入

【仮回值】

返回值	描述	A TOPING
0	成功	H. H. L.
非 0	失败,	其值见错误码

【需求】

头文件: mpi\_ise.h

库文件: libmpp\_ise.so

【注意】

无

【举例】



无

# 5.2.2.4 AW\_MPI\_ISE\_GetGrpAttr

【目的】

获取 ISE 组的属性

【语法】

ERRORTYPE AW\_MPI\_ISE\_GetGrpAttr(ISE\_GRP IseGrp, ISE\_GROUP\_ATTR\_S \*pGr-pAttr);

【参数】

参数	描述	Ja Karaman	输入/轴	<b>俞出</b>
IseGrp	组 ID 号	HE THE STATE OF TH	输入	
pGrpAttr	组属性,选	择 ISE 组件工作模式	输出	-16

【返回值】

返回值	直 描述
0	成功
非 0	失败,其值见错误码

【需求】

头文件: mpi ise.h

库文件: libmpp ise.so

【注意】

无

【举例】

无

### 5.2.2.5 AW\_MPI\_ISE\_CreatePort

【目的】



创建 ISE 输出端口

### 【语法】

ERRORTYPE AW\_MPI\_ISE\_CreatePort(ISE\_GRP IseGrp, ISE\_CHN IsePort, ISE\_CHN\_ATTR\_S \*pChnAttr);

### 【参数】

参数	描述	输入/输出
IseGrp	组 ID 号	输入
IsePort	端口 ID 号	输入
pChnAttr	端口属性,设置输出端口属性	输入

【返回值】

端口属	生,设置输出端口属性	输入
M D A	生,设直输出端口属性 描述	输入 Jindon
返回值	描述	(S) Hall Hall Tr.
0 非 0	成功 失败,其值见错误码	
A L	LWIN	skeri kari

### 【需求】

头文件: mpi ise.h

库文件: libmpp\_ise.so

【注意】

每个ISE Group 都有 4 个端口可以输出,每个端口相互独立。

【举例】

无

### 5.2.2.6 AW\_MPI\_ISE\_DestroyPort

### 【目的】

销毁 ISE 输出端口

【语法】

ERRORTYPE AW\_MPI\_ISE\_DestroyPort(ISE\_GRP IseGrp, ISE\_CHN IsePort);



输入/输出 参数 描述 组 ID 号 输入 IseGrp 端口ID号 **IsePort** 输入

【返回值】

返回值 描述 0 成功 非 0 失败,其值见错误码

【需求】

头文件: mpi ise.h

库文件: libmpp ise.so

【注意】

无

【举例】

无

ort. AW\_MPI\_ISE\_GetPortAttr

【目的】

获取 ISE 输出端口属性

【语法】

ERRORTYPE AW\_MPI\_ISE\_GetPortAttr(ISE\_GRP IseGrp, ISE\_CHN IsePort, ISE\_CHN\_ATTR\_S \*pChnAttr);

参数	描述	输入/输出
IseGrp	组 ID 号	输入
IsePort	端口 ID 号	输入
pChnAttr	端口属性,设置输出端口属性	输出



### 【返回值】

λ-X ,		ζ=X'3
	返回值	描述
	0	成功。
	非 0	失败,其值见错误码

【需求】

头文件: mpi ise.h

库文件: libmpp ise.so

【注意】

无

【举例】

无

..CT

5.2.2.8 AW\_MPI\_ISE\_SetPortAttr

【目的】

设置 ISE 输出端口属性

【语法】

ERRORTYPE AW\_MPI\_ISE\_SetPortAttr(ISE\_GRP IseGrp, ISE\_CHN IsePort, ISE\_CHN\_ATTR\_S \*pChnAttr);

【参数】

-V- 1		V_ 1	
参数	描述	- K. 1	输入/输出
IseGrp	组 ID 号		输入
IsePort	端口 ID 号	<u>1</u>	输入
pChnAttr	端口属性,	设置输出端口属性	输入

【返回值】

返回值	描述	
Q allin	成功	
非 0	失败,	其值见错误码

NE RELIGIONAL DE LA CONTRACTOR DE LA CON



### 【需求】

头文件: mpi\_ise.h

库文件: libmpp\_ise.so

【注意】

无

【举例】

无

# AW\_MPI\_ISE\_Start

【目的】

启动 ISE 组件,所有创建的 port 会同时启动工作

【语法】

ERRORTYPE AW\_MPI\_ISE\_Start(ISE\_GRP IseGrp);

【参数】

描述 输入/输出 IseGrp 组 ID 号 输入

【返回值】

返回值 描述 0 成功 非 0 失败,其值见错误码

【需求】

头文件: mpi\_ise.h

库文件: libmpp\_ise.so

【注意】

(**举例**]



### AW MPI\_ISE\_Stop 5.2.2.10

【目的】

停止 ISE 组件,所有创建的 port 会同时停止工作。

【语法】

ERRORTYPE AW MPI ISE Stop(ISE GRP IseGrp);

NE PREMITE HERE 描述 输入/输出 输入 IseGrp 组 ID 号

【返回值】

返回值 描述 0 成功 失败, 非 0 其值见错误码

【需求】

头文件: mpi ise.h

库文件: libmpp\_ise.so

【注意】

无

【举例】

无

# 5.2.2.11 AW\_MPI\_ISE\_GetData

【目的】

在非绑定的情况下,从端口获取 ISE 处理后的数据



### 【语法】

ERRORTYPE AW\_MPI\_ISE\_GetData(ISE\_GRP IseGrp, ISE\_CHN IsePort, VIDEO\_FRAME\_INFO\_S \*pstIseData, AW\_S32 nMilliSec);

### 【参数】

参数	描述		
IseGrp	组 ID 号		
IsePort	端口 ID 号		
pstIseData	ISE 组件处理后的数据流		
nMilliSec	获取数据的超时时间- $1$ 表示阻塞模式; $0$ 表示非阻塞模式; $>0$ 表示阻塞 $\mathrm{nMilliSec}$ 毫秒,超时则		

### 【返回值】

Maria Illinden.	描述 成功 失败,其值见错误码
返回值	描述
0	成功
非 0	失败,其值见错误码

### 【需求】

头文件: mpi\_ise.h

库文件: libmpp ise.so

### 【注意】

绑定模式下,此函数不起作用

【举例】

### 5.2.2.12 AW MPI\_ISE\_ReleaseData

### 【目的】

在非绑定的情况下,从端口释放 ISE 处理后的数据

### 【语法】

ERRORTYPE AW\_MPI\_ISE\_ReleaseData(ISE\_GRP IseGrp, ISE\_CHN IsePort, VIDEO\_FRAME\_INFO\_S \* pstIseData);



A11/	A11/	
参数	描述	输入/输出
IseGrp	组 ID 号	输入
IsePort	端口 ID 号	输入
pstIseData	ISE 组件处理后的数据流	输入

### 【返回值】

返回值	描述	
0		
非 0	失败,其值见错误码	
4F O	失败,其值见错误码	Allinderii Renterii R
	1	
Send	/ Pic	aindeni iindeni

### 【需求】

头文件: mpi ise.h

库文件: libmpp\_ise.so

### 【注意】

绑定模式下,此函数不起作用

【举例】

无

# 5.2.2.13 AW\_MPI\_ISE\_SendPic

### 《首的》

在非绑定的情况下,用户向 ISE 组件发送需要处理的数据

### 【语法】

ERRORTYPE AW\_MPI\_ISE\_SendPic(ISE\_GRP IseGrp, VIDEO\_FRAME\_INFO\_S pstUserFrame0, VIDEO FRAME INFO S pstUserFrame1, AW S32 nMilliSec);

参数	描述		
IseGrp	组 ID 号	ing	is.
pstUserFrame0	第一路源图像数据	allinde <sup>8</sup>	Alindo
pstUserFrame1	第二路源图像数据	SE IL	W. T.
nMilliSec	发送数据的超时时间,	-1 表示阻塞模式; 0 表示非阻塞模式;	>0 表示阻塞 nMilliSec 毫秒,



# 【返回值】

	L-X
返回值	描述
0	成功
非 0	失败,其值见错误码

【需求】

头文件: mpi ise.h

库文件: libmpp ise.so

【注意】

绑定模式下,此函数不起作用

【举例】

# Will the state of 5.2.2.14 AW\_MPI\_ISE\_RegisterCallback

【目的】

在非绑定的情况下,ISE 通知用户帧已经处理完

【语法】

AW\_MPI\_ISE\_RegisterCallback(ISE\_GRP IseGrp, MPPCallbackInfo ERRORTYPE \*pCallback);

参数	描述	输入/输出
IseGrp	组ID号	输入
pCallback	回调参数	输入

【返回值】

返回值	描述	
0	成功	
非0点	失败,	其值见错误码



头文件: mpi ise.h

库文件: libmpp\_ise.so

【注意】

绑定模式下,此函数不起作用

【举例】

无

### 5.2.2.15 AW\_MPI\_ISE\_SetISEFreq

【目的】

设置 ISE 模块时钟频率。

【语法】

ERRORTYPE AW\_MPI\_ISE\_SetISEFreq(ISE\_GRP IseGrp, int nFreq)

### 【参数】

参数	描述	114	输入/输出
IseGrp	组ID号		 输入
nFreq	引擎时钟频率,单位 MHz。	范围: [432-648],且为 12 的倍数单	输入
	目鱼眼和双目拼接模式默认	432 MHz,双目鱼眼模式默认为	
Hini	576MHz。	Mindell	allinder

# 【返回值】

返回值	描述
0	成功
非 0	失败,其值见错误码。

### 【需求】

头文件: mpi\_ise.h

库文件: libmpp\_ise.so

【注意】

丸。



【举例】

# 5.3 数据结构

\*ISE\_GROUP\_ATTR\_S\*

【说明】

定义 ISE 组属性

【定义】

typedef struct

AW U32 iseMode;

PIXEL FORMAT E mPixelFormat;

} ISE\_GROUP\_ATTR\_S;

【成员】

成员名称 描述

设置 ISE 组属性,选择 ISE 组件工作模式 iseMode

设置 ISE 组件 pixel format, 取值参见: PIXEL FORMAT E 定 mPixelFormat

【注意事项】

【相关数据类型及接口】

无

\*WARP\_Type\_MO\*

【说明】

定义单目鱼眼矫正方式

【定义】

typedef enum WARP\_Type\_MC

Real of the state of the state



 $WARP_PANO180 = 0x0001*,*$ 

WARP\_PANO360 = 0x0002\*,\*

WARP NORMAL = 0x0003\*,\*

WARP UNDISTORT = 0x0004\*,\*

WARP 180WITH2 = 0x0005

}WARP\_Type\_MO\*;\*

# 【成员】

	1110		1100
成员名称	WIN.	描述	N. A.
WARP_P	NO180	180 全景校正模式	t and the same of
WARP_PA	ANO360	360 全景校正模式	忧(左右展开) 🕝 🧬
WARP_N	ORMAL	Normal 校正模式	
WARP_U	NDISTORT	镜头畸变校正模式	
WARP_18	30WITH2	360 全景校正模式	(上下展开)
接口】	AU	MIL	à

# 【注意事项】

无

【相关数据类型及接口】

无

\*MOUNT\_Type\_MO\*

【说明】

定义单目鱼眼镜头安装方式

【定义】

typedef enum MOUNT\_Type\_MO{

 $MOUNT_TOP = 0x0001*,*$ 

 $MOUNT_WALL = 0x0002*,*$ 

 $MOUNT_{\sim}BOTTOM = 0x0003$ 

}MOUNT\_Type\_MO\*;\* 【成员】



	成员名称	描述
特別	MOUNT_TOP	顶装模式
7	MOUNT_WALL	壁装模式
	MOUNT_BOTTO	M 地装模式

# 【注意事项】

无

【相关数据类型及接口】

无

\*ISE\_CFG\_PARA\_MO\*

【说明】

定义单目鱼眼模式相关配置

【定义】

typedef struct ISE CFG PARA MO

{

WARP\_Type\_MO dewarp\_mode\*;\*

MOUNT Type MO mount mode\*;\*

int in h\*\*

int in w\*;\*

int in\_luma\_pitch\*;\*

int in\_chroma\_pitch\*;\*

int in yuv type\*;\*

float pan\*;\*

float tilt\*;\*

float zoom;

int out\_en[4];

int out\_h[4];

in out\_w[4];

A STATE OF THE STA

TAKE TO THE PERSON OF THE PERS

版权所有 © 珠海全志科技股份有限公司。保留一切权利

130



```
int out_luma_pitch[4];
int out_chroma_pitch[4];
int out_flip[4];
int out_mirror[4];
int out_yuv_type;
float p*;*
float cx*;*
                            Regi /
float cy*;*
float fx*;*
float fy*;*
float cxd*;*
float cyd*;*
float fxd*;*
float fyd*;*
float k[6]*;*
float p undis[2]*;*
double calib_matr_cv[3][3]*;*
double distort[6]
char reserved[32]*;*
}ISE_CFG_PARA_MO*;*
```

# 【成员】

成员名称	描述		
dewarp_mode	校正模式,静态属性		
$mount\_mode$	安装方式,静态属性		
in_h	源图像高度,静态属性	Refi!	bei
in_w@lift	源图像宽度,静态属性	Alline	- Bline
in_luma_pitch	源图像亮度(Y 分量)	pitch 值,32 的倍数最大值为	3 4096,静态属
XA CONTRACTOR OF THE PARTY OF T	性类		A STATE OF THE STA



ALLWIMER	a dilina	- Allin	文档密级、种
成员名称	描述		
in_chroma_pitch	源图像色度(UV 分量 属性	)pitch 值,32 的倍数	效最大值为 4096,静态
in_yuv_type		表示 NV21 格式,静深	<b></b>
pan	Normal 模式下视场角	自的水平转动,动态属性	ŧ
tilt	Normal 模式下视场角	的垂直转动,动态属性	ŧ
zoom	Normal 模式下视场角	自大小,动态属性	
out_en[4]	是否使能单目鱼眼硬件	通道,硬件通道 0 ~ 3	3 分别对应 Port 0 ~
	30: 不使能 1: 使能 (	out_en[0]: 静态属性	out_en[1~3]: 动态属
	性,若不使能则需要把	l对应的 Port 销毁掉,	在需要使能时重新创建
	Port		
out_h[4]	目标图像的高度,硬件	通道 0 ~ 3 分别对应	Port 0 ~ 3out_h[1 ~
			out_h[0]: 静态属性
A STATE OF THE STA	out h[1~3]: 动态属		
out_w[4]	目标图像的宽度,硬件	- 通道 0 ≈ 3 分别对应	Port 0 ~ 3out_w[1 ~
out_w[4]	120/1	* out_w[0],8 的倍数	
12	out w[1~3]: 动态属	. 7/5-	
out luma pitch[4]		-300	2 的倍数硬件通道 0 ~
		out_luma_pitch[0]:	
	out luma pitch[1~		
out chroma pitch!	[4] 目标图像色度(UV 分		32 的倍数硬件诵道 0 ~
ouc_om_om_proom		3out_chroma_pitch	
	out chroma pitch[1		[○]• H3.VEVIA] IT
out_flip[4]		翻转,硬件通道 0 ~ 3	3 分别对应 Port 0 ~
out_mp[1]			out_flip[1~3]: 动态
deri	属性	geroderi	
out_mirror[4]		g,硬件通道 0 ≈ 3 分别	别对应 Port 0 ~ 30;
AND THE PERSON NAMED IN COLUMN TWO IS NOT THE PERSON NAMED IN COLUMN TO THE PERSON NAMED IN COLU	N/QL	√QL °	ıt_mirror[1~3]: 动和
84.	属性	NA THE REAL PROPERTY OF THE PERTY OF THE PER	<u> </u>
out yuv type	目标图像 YUV 格式,	0 表示 NV21 格式,	静态属性 / / / / / / / / / / / / / / / / / / /
pp		-/_1	镜头焦距需要使用标定
**,	工具进行标定,静态属	-,	**
CX	源图像的镜头参数,光	光心水平坐标,需要使用	目标定工具进行标定静态
	属性		
су	源图像的镜头参数,光	七心垂直坐标,需要使用	目标定工具进行标定静态
J	属性		
fx		参数,畸变图像的水平	· · 方向焦距需要使用标定
	工具进行标定,静态属		
fy			ī方向焦距需要使用标定
.:.de1.	工具进行标定,静态属	76/	\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
cxd			光心水平坐标需要使用
Cxd V	标定工具进行标定,静	190	A TOP TO SERVICE AND A SERVICE
◊′	A MARKAN, A STATE OF THE STATE	A WEEK XX	AND THE REAL PROPERTY.
JA."	<b>近初於左 ◎ 建海人士科共</b>	设份有限公司。保留一切权利	-1(2)



	\$1\Z		
成员名称	描述	A THE STATE OF THE	A A A A A A A A A A A A A A A A A A A
cyd	普通镜头畸变校正模式参数	,畸变校正后图像光心垂直坐	标需要使用
	标定工具进行标定,静态属的	't	
fxd	普通镜头畸变校正模式参数	,畸变校正后图像水平方向焦	距需要使用
	标定工具进行标定,静态属的	生	
fyd	普通镜头畸变校正模式参数	,畸变校正后图像垂直方向焦	距需要使用
	标定工具进行标定,静态属位	生	
k[6]	普通镜头畸变校正模式参数	,径向畸变系数需要使用标定	工具进行标
	定,静态属性		
p_undis[2]	普通镜头畸变校正模式参数	,法向畸变系数需要使用标定	工具进行标
	定,静态属性		
calib_matr[3][3]	扩展类型,暂不支持	Meni	der
calib_matr_cv[3][3]	扩展类型,暂不支持		A STATE OF THE PARTY OF THE PAR
distort[8]	扩展类型,暂不支持	Wallinderi	A TOP TO SERVICE AND A SERVICE
reserved[32]	保留,暂不涉及	A STATE OF THE STA	XXY.

# 【注意事项】

1、由于单目鱼眼模式下不同的校正模式,其输入图像的宽高设置、安装方式也有所不同,如图 2-2 所示。

	校正模式 dewarp_mode		输入图像大小 in_h in_w		安装方式 mount_mode	
	WARP_PANO180 (180模式)		輸入宽高1:1/ 最大支持3000*3000 最小为768*768		gr <sup>i</sup> 无	, S.
	WARP_PANO360 (360模式)		输入贯高1: 1 最大支持2048*2048 最小为768*768	MOUNT_TOP(顶装) MOUNT_BOTTON(地装)	无。	
	WARP_NORMAL	茶筒	输入宽高1: 1	MOUNT_TOP(顶装) MOUNT_WALL(壁装)	P: [0, 360) T: [0, 90] Z: [1, 4] P: [25, 155] T: [25, 155] Z: [1, 4]	-
	(PTZ模式)	宽高为8 的倍数	最大支持3000*3000 最小为768*768	MOUNT_BOTTON(地装)	P: [0, 360] T: [0, 90] Z: [1, 4]	
	WARP_UNDISTORT (镜头畸变矫正模式)		輸入宽高16:9/4:3/1:1 最大支持4096*2304/ 4096*3072/4096*4096 最小为 1152x864/1366x768/768*76		无	
4	WARP_180WITH2 (第二种360模式)		输入宽高1: 1 最大支持2048*2048 最小为768*768	MOUNT_TOP(顶装) MOUNT_BOTTON(地装)	ET THING	3

版权所有 © 珠海全志科技股份有限公司。保留一切权利

133



# 图 2-2

- 2、图像水平翻转和镜像每个 Port 只能使能一个。
- 3、Normal 模式下 PTZ 参数在 ISE 组件运行过程中支持动态设置,动态设置间隔时间不得小于 60ms 且必须在 ISE Group 处于 COMP\_StateExecuting 之后再进行设置。

# 【相关数据类型及接口】

无

\*ISE\_CFG\_PARA\_\* \*BI\*

【说明】

定义双目鱼眼模式相关配置

【定义】

typedef struct ISE CFG PARA BI{

int in h\*;\*

int in w\*;\*

int in luma pitch\*;\*

int in chroma pitch\*;\*

int in yuv type\*;\*

int out\_en[4]\*;\*

int out h[4]\*;\*

int out\_w[4]\*;\*

int out\_flip[4]\*;\*

int out\_mirror[4]\*;\*

int out\_luma\_pitch[4]\*;\*

int out chroma pitch[4]\*;\*

int out yuv type\*;\*

float p0\*;\*

int cx0\*\*

int cy0\*;\*

If the state of th



float

int cx1\*;\*

int cy1\*;\*

float calib\_matr[3][3]\*;\*

double calib matr cv[3][3]\*;\*

double distort[8]\*;\*

float change focus;

# 【成员】

int trans_width; int findpath_width; int pyr_level; int findpath_enable; float inv_scale; char reserved[32]*;*	float change_focus;	
int int pyr_level; int findpath_enable; float inv_scale; char reserved[32]*;*  ISE_CFG_PARA_BI*;*  [成员]  成员名称  描述  in h  源图像高度,4 的倍数取值范围 [320,4032],静态属性 源图像宽度,4 的倍数取值范围 [320,4032],静态属性 in_luma_pitch in_luma_pitch in_chroma_pitch in_chroma_pitch in_yvv_type  out_en[4]  是否使能双目鱼眼硬件通道,硬件通道 0 ~ 3 分别对应 Port 0 ~ 30: 不使能 1: 使能 out_en[0]: 静态属性 out_en[1~3]: 动态属性,若不使能则需要把对应的 Port 销毁掉,在需要使能时重新创建 Port  out_h[4]  目标图像的高度,硬件通道 0 ~ 3 分别对应 Port 0 ~ 30ut_h[0] 为 4 的倍数取值范围 [320,4032]out_h[1 ~ 3] 须满足 [0.125,1] *	int feather;	ninderi ninderi
int int pyr_level; int findpath_enable; float inv_scale; char reserved[32]*;*  ISE_CFG_PARA_BI*;*  [成员]  成员名称  描述  in h  源图像高度,4 的倍数取值范围 [320,4032],静态属性 源图像宽度,4 的倍数取值范围 [320,4032],静态属性 in_luma_pitch in_luma_pitch in_chroma_pitch in_chroma_pitch in_yvv_type  out_en[4]  是否使能双目鱼眼硬件通道,硬件通道 0 ~ 3 分别对应 Port 0 ~ 30: 不使能 1: 使能 out_en[0]: 静态属性 out_en[1~3]: 动态属性,若不使能则需要把对应的 Port 销毁掉,在需要使能时重新创建 Port  out_h[4]  目标图像的高度,硬件通道 0 ~ 3 分别对应 Port 0 ~ 30ut_h[0] 为 4 的倍数取值范围 [320,4032]out_h[1 ~ 3] 须满足 [0.125,1] *	int trans_width;	
int findpath_enable; float inv_scale; char reserved[32]*;*    ISE_CFG_PARA_BI*;*   IMD	int findpath_width;	
Char reserved[32]*;*   SISE_CFG_PARA_BI*;*   INDEXTIFY	int pyr_level;	
Char reserved[32]*;*   SISE_CFG_PARA_BI*;*   INDEXTIFY	int findpath_enable;	
ISE_CFG_PARA_BI*;*   I成员	float inv_scale;	
成员名称 描述 源图像高度,4 的倍数取值范围 [320,4032],静态属性 源图像宽度,4 的倍数取值范围 [320,4032],静态属性 源图像宽度,4 的倍数取值范围 [320,4032],静态属性 源图像亮度(Y 分量)pitch 值,32 倍数,静态属性 源图像色度(UV 分量)pitch 值,32 倍数,静态属性 源图像色度(UV 分量)pitch 值,32 倍数,静态属性 源图像 YUV 格式,0 表示 NV21 格式,静态属性 源图像 YUV 格式,0 表示 NV21 格式,静态属性 0ut_en[4] 是否使能双目鱼眼硬件通道,硬件通道 0 ~ 3 分别对应 Port 0 ~ 30: 不使能 1: 使能 out_en[0]: 静态属性 out_en[1~3]: 动态属性,若不使能则需要把对应的 Port 销毁掉,在需要使能时重新创建 Port 目标图像的高度,硬件通道 0 ~ 3 分别对应 Port 0 ~ 3out_h[0] 为 4 的倍数取值范围 [320,4032]out_h[1 ~ 3] 须满足 [0.125,1] *	char reserved[32]*;*	
成员名称  in_h  in_M  in_W  in_luma_pitch in_chroma_pitch in_yvv_type  out_en[4]  in_H  源图像高度,4 的倍数取值范围 [320,4032],静态属性  源图像高度(Y 分量)pitch 值,32 倍数,静态属性  源图像色度(UV 分量)pitch 值,32 倍数,静态属性  源图像色度(UV 分量)pitch 值,32 倍数,静态属性  源图像 YUV 格式,0 表示 NV21 格式,静态属性  是否使能双目鱼眼硬件通道,硬件通道 0 ~ 3 分别对应 Port 0 ~  30: 不使能 1: 使能 out_en[0]: 静态属性 out_en[1~3]: 动态属  性,若不使能则需要把对应的 Port 销毁掉,在需要使能时重新创建  Port  out_h[4]  目标图像的高度,硬件通道 0 ~ 3 分别对应 Port 0 ~ 3out_h[0] 为  4 的倍数取值范围 [320,4032]out_h[1 ~ 3] 须满足 [0.125,1] *	}ISE_CFG_PARA_BI*;*	
in h	【成员】	
in_w in_luma_pitch in_chroma_pitch in_chroma_pitch in_yuv_type out_en[4]  out_h[4]  in_b	成员名称	描述 glind glind
in_luma_pitch in_chroma_pitch in_chroma_pitch in_yuv_type  out_en[4]  out_h[4]  in_luma_pitch in_slage(Y分量) pitch 值, 32 倍数,静态属性  源图像色度(UV分量) pitch 值, 32 倍数,静态属性  源图像 YUV 格式, 0表示 NV21 格式,静态属性  是否使能双目鱼眼硬件通道,硬件通道 0~3分别对应 Port 0~ 30: 不使能 1: 使能 out_en[0]: 静态属性 out_en[1~3]: 动态属性,若不使能则需要把对应的 Port 销毁掉,在需要使能时重新创建  Port  out_h[4]  目标图像的高度,硬件通道 0~3分别对应 Port 0~3out_h[0]为 4的倍数取值范围 [320,4032]out_h[1~3] 须满足 [0.125,1]*	in h	源图像高度,4 的倍数取值范围 [320,4032],静态属性
in_chroma_pitch 源图像色度(UV 分量)pitch 值,32 倍数,静态属性 源图像 YUV 格式,0 表示 NV21 格式,静态属性 0ut_en[4] 是否使能双目鱼眼硬件通道,硬件通道 0 ~ 3 分别对应 Port 0 ~ 30: 不使能 1: 使能 out_en[0]: 静态属性 out_en[1~3]: 动态属性,若不使能则需要把对应的 Port 销毁掉,在需要使能时重新创建 Port 0ut_h[4] 目标图像的高度,硬件通道 0 ~ 3 分别对应 Port 0 ~ 3out_h[0] 为 4 的倍数取值范围 [320,4032]out_h[1 ~ 3] 须满足 [0.125,1] *	in_w	源图像宽度,4的倍数取值范围 [320,4032],静态属性
in_yuv_type	in_luma_pitch	- \_Y,
out_en[4]       是否使能双目鱼眼硬件通道,硬件通道 0 ~ 3 分别对应 Port 0 ~         30: 不使能 1: 使能 out_en[0]: 静态属性 out_en[1~3]: 动态属性, 若不使能则需要把对应的 Port 销毁掉, 在需要使能时重新创建Port         out_h[4]       目标图像的高度,硬件通道 0 ~ 3 分别对应 Port 0 ~ 3out_h[0] 为4 的倍数取值范围 [320,4032]out_h[1 ~ 3] 须满足 [0.125,1] *	XK,	₩,
30: 不使能 1: 使能 out_en[0]: 静态属性 out_en[1~3]: 动态属性,若不使能则需要把对应的 Port 销毁掉,在需要使能时重新创建Port out_h[4] 目标图像的高度,硬件通道 0~3分别对应 Port 0~3out_h[0]为4的倍数取值范围 [320,4032]out_h[1~3] 须满足 [0.125,1]*		
性,若不使能则需要把对应的 Port 销毁掉,在需要使能时重新创建 Port out_h[4] 目标图像的高度,硬件通道 0 ~ 3 分别对应 Port 0 ~ 3out_h[0] 为 4 的倍数取值范围 [320,4032]out_h[1 ~ 3] 须满足 [0.125,1] *	out_en[4]	
Port out_h[4] 目标图像的高度,硬件通道 0 ~ 3 分别对应 Port 0 ~ 3out_h[0] 为 4 的倍数取值范围 [320,4032]out_h[1 ~ 3] 须满足 [0.125,1] *		
out_h[4]       目标图像的高度,硬件通道 0 ~ 3 分别对应 Port 0 ~ 3out_h[0] 为         4 的倍数取值范围 [320,4032]out_h[1 ~ 3] 须满足 [0.125,1] *		
4 的倍数取值范围 [320,4032]out_h[1 ~ 3] 须满足 [0.125,1] *		
	out_h[4]	<del>-</del>
out_h[0], 4 的倍数 out_h[0]: 静态属性 out_h[1~3]: 动态属性         out_w[4]       目标图像的宽度,硬件通道 0~3分别对应 Port 0~3out_w[0]为         8 的倍数取值范围 [640, 8064]out_w[1~3] 须满足 [0.125,1]*		<del>-</del>
out_w[4]       目标图像的宽度,硬件通道 0 ~ 3 分别对应 Port 0 ~ 3out_w[0] 为         8 的倍数取值范围 [640, 8064]out_w[1 ~ 3] 须满足 [0.125,1]*	- 48/11	
8 的倍数取值范围 [640, 8064]out_w[1 ~ 3] 须满足 [0.125,1]*	out_w[4]	
	THE VIEW OF THE PARTY OF THE PA	\QV\\ \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
out_w[0], 8 的倍数 out_w[0]: 静态属性 out_w[1~3]: 动态属性	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	out_w[0], 8 的倍数 out_w[0]. 静态属性 out_w[1~3]: 动态属性



_				7,12,12,12
-	成员名称	描述	A CONTRACTOR OF THE PARTY OF TH	
Ŕ	out flip[4]	是否使能通道图像水平	网转,硬件通道 0 ~ 3 分别对	d应 Port 0 ~
		/	ut_flip[0]:静态属性 out_fli	Z,'4/
		属性	%ı — - — — — — — — — — — — — — — — — — —	% <sup>T</sup>
	out mirror[4]	是否使能通道图像镜像,	硬件通道 0 ~ 3 分别对应 I	Port 0 ~ 30: 不
	_	使能 1: 使能 out mir	ror[0]: 静态属性 out mir	ror[1~3]: 动态
		 属性	_	
	out luma pitch[4]	目标图像亮度(Y 分量)	pitch 值,16 或 32 的倍数	<b>対硬件通道 0 ~ 3</b>
			- ɪt luma pitch[0]:静态属性	
		out luma pitch[1~3	<del></del>	
	out chroma pitch[4]		- 量)pitch 值,16 或 32 的倍	数硬件通道 0 ~
		•	out chroma pitch [0]: 静	
	Bline	out chroma pitch [1		Allino
	out_yuv_type	- ^/\7	表示 NV21 格式,静态属性	E
->	p0	第一组源图像的镜头参数	数,镜头焦距,需要使用标定	工具进行标定静
,	r ·	态属性		(C) ALIVER
	cx0	第一组源图像的镜头参数	数,光心水平坐标,需要使用	标定工具进行标
	A.K. Tarak	定静态属性		
	cy0		数,光心垂直坐标,需要使用	标定工具进行标
	J	定静态属性	11 11 1	
	p1		数,镜头焦距,需要使用标定	工具进行标定静
	r	态属性	No.	
	cx1	- "	) 效,光心水平坐标,需要使用	标定工具进行标
		定静态属性		
	cy1		数,光心垂直坐标,需要使用	标定工具进行标
	inderi'	定静态属性。	. ndeni	269
	calib matr[3][3]	源图像的镜头参数,镜	头旋转矩阵,需要使用标定工	具进行标定静态
	KAN THE STATE OF T	属性	A THE PARTY OF THE	X. Aller
	change_focus	软件拼接算法参数,视	<b>皇校正系数,静态属性</b>	NA THE REAL PROPERTY OF THE PERSON OF THE PE
	feather	软件拼接算法参数,重	叠区域半径,静态属性	H. H. C.
	trans_width	软件拼接算法参数,过滤	度区大小,静态属性	%T'
	findpath_width	软件拼接算法参数,最低	圭路径查找区域大小,静态属	性
	pyr_level	软件拼接算法参数,金字	字塔级数设定,静态属性	
	findpath_enable	软件拼接算法参数,最低	圭路径使能控制,静态属性	
	inv_scale	软件拼接算法参数,最低	圭路径查找缩放比例,静态属	性
	calib_matr_cv[3][3]	扩展类型,暂不支持		
	distort[8]	扩展类型,暂不支持		
	reserved[32]	保留,暂不涉及		

# 【注意事项】

# 1、输入图像宽高要求 1:1。 —————



2、图像水平翻转和镜像每个 Port 只能使能一个。

【相关数据类型及接口】

无

\*ISE\_CFG\_PARA\_\* \*ST\*\*I\*

【说明】

定义双目拼接模式相关配置

【定义】

typedef struct ISE\_CFG\_PARA\_STI{\}

int ncam\*;\*

int in\_h\*;\*

int in w\*;\*

int pano h\*;\*

int pano\_w\*;\*

int ov\*;\*

int yuv\_type\*;\*

float p0\*;\*

float p1\*;\*

int offset r2l\*;\*

float pano fov\*;\*

float t\_angle\*;\*

int stre en\*;\*

float stre coeff\*;\*

float hfov\*;\*

float wfov\*;\*

float wfov\_rev\*;\*

double calib matr[3][3]\*;\*

double calib\_matr\_cv[3][3]\*;\*

Real Property of the State of t



double distort[8]\*;\*

int in\_luma\_pitch\*;\*

int in\_chroma\_pitch\*;\*

int pano\_luma\_pitch\*;\*

int pano\_chroma\_pitch\*;\*

char reserved[32]\*;\*

}ISE\_CFG\_PARA\_STI;

# 【成员】

	July 1	11000	<sup>l</sup> do	object.
	成员名称	描述	W <sub>1</sub> V	WIV.
2	ncam	相机数量,静态属性	SEX TO SERVICE	A A A A A A A A A A A A A A A A A A A
,	in_h	源图像高度,2 的倍数	取值范围 [120,1728],静态属	性
	in_w	源图像宽度,取值范围	[160,2560] 宽高比为 4:3 时,	,为 8 的倍
	K. The	数; 宽高为 16:9 时系	为 32 的倍数,静态属性	<b>&gt;</b>
	pano_h	目标图像 Port 0 的高原	度,4 的倍数取值范围 [64,204	8],静态属性
	pano_w	目标图像 Port 0 的宽度	度与输入图像宽度的关系为: pa	no_w =
		[0.8,1] * 2 * in_w16	的倍数取值范围 [320,4096] 静	净态属性
	ov	重叠区总宽度,与输出	宽度的关系为: ov < 1/10 * p	ano_w 小于等
		于 320 且为 16 的倍数	<b>牧,静态属性</b>	
	yuv_type	图像 YUV 格式, 0表:	示 NV21 格式,静态属性	
	p0		的夹角,取值范围 [0,180],静态	
	p1 indent	iille	9夹角,取值范围 [0,180],静态	1110
	offset_r2l		偏移像素数,暂不开放,默认为	21/2
	pano_fov	XXXX	约值,需要使用标定工具进行标	定,静态属性
R	t_angle	VAV.	度,暂不开放,默认为 0	NAT THE RESERVENCE OF THE PERSON OF THE PERS
	stre_en	视场拉伸开关,暂不开	45	11 To 12 To
	stre_coeff	视场拉伸系数,>1.0,	X 1/27	
	hfov		角,单位为度,需要使用标定工	
	0	态属性		n> == + =
	wfov		角,单位为度(只配置 wfov 即	」可)需要使用
	C	标定工具进行标定,静		
	wfov_rev		位为度,需要使用标定工具进行	「杯疋,靜念属
	1:1 . [0][0]	性		3 MA
	calib_matr[3][3]		使用标定工具进行标定,静态属	
	calib_matr_cv[3][3]		需要使用标定工具进行标定,静 完工具进行标定,整本层性	# <b>念</b> 属性
	distort[8] in luma pitch		定工具进行标定,静态属性 pitch 使 22 的倍数 整本层域	M- Allino
	in chroma pitch	W.	pitch 值,32 的倍数,静态属的 )pitch 值,32 的倍数,静态原	√Kr.
-	TH_CIII OIIIa_PIUCII	派国隊C及(UV 万里	,piwii ja,32 pyla效,静态/	五江



成员名称	描述		
pano_luma_pitch	目标图像 Port0 亮度	(Y 分量) pitch 值,16 或 3	32 的倍数,静态
	属性		
pano_chroma_pitch	目标图像 Port0 图像	色度(UV 分量)pitch 值,	16 或 32 的倍数
	静态属性		
reserved[32]	保留,暂不涉及		

# 【注意事项】

图像水平翻转和镜像只能使能一个。

・定义】

typedef struct *ISE\_PROCCFG\_PARA\_STI*{

int bgfgmode\_en\*;\*

nt pano\_flip\*;\*

it pano\_mirr;

int

int scalar h[3]\*;\*

scalar\_w[3]\*\*\*\\*;\\*\*\*

int scalar luma pitch[3]\*;\*

int scalar\_chroma\_pitch[3]\*;\*

int scalar\_flip[3]\*;\*

int scalar mirr[3]\*;\*

int scalar\_en[3]\*;\*

char reserved[32]\*;\*

}ISE\_PROCCFG\_PARA\_STI\*;



# 【成员】

X.z. \(\times_1\)	ZX.2.	Y=X,3,	Y=X.3
成员名称	描述	Harris Commencer	Hall the same of t
bgfgmode_en	是否使能背景。	<b>建模 0: 不使能 1: 使能</b>	动态参数。
pano_flip	是否使能目标图	图像 Port0 水平翻转 0:	不使能 1:使能静态属性
pano_mirr	是否使能目标图	图像 Port0 镜像 0:不使	能 1:使能静态属性
scalar_h[3]	scale 通道目标	示图像高度,scalar_h[0	~ 3] 分别对应 Port 1 ~
	_	~ 3] 须满足 [0.125,1]	* pano_h,4 的倍数,动
	态属性		
scalar_w[3]	scale 通道目标	示图像宽度,scalar_w[(	) ~ 3] 分别对应 Port 1 ~
	3scalar_w[1	~ 3] 须满足 [0.125,1]	* pano_w,8 的倍数,动
in	态属性		in.
scalar_luma_pitch[3]	目标图像亮度	(Y 分量) pitch 值, 16	或 32 的倍数
N. IV	scalar_luma_	pitch[0 ~ 3] 分别对应	Port 1 ~ 3,动态属性
scalar_chroma_pitch[3]	目标图像色度	(UV 分量) pitch 值,	l6 或 32 的倍数 scalar_
	chroma _pito	:h[0 ~ 3] 分别对应 Por	t 1 ~ 3,动态属性
scalar_flip[3]	是否使能通道图	图像水平翻转,scalar_f	lip[0 ~ 3] 分别对应 Port
**************************************	1 ~ 30: 不使	能 1:使能动态属性	THE THE PARTY OF T
scalar_mirr[3]	是否使能通道图	图像镜像,scalar_mirr	0 ~ 3] 分别对应 Port 1 ~
	30: 不使能 1	:使能动态属性	
scalar_en[3]	是否使能双目抗	拼接 scale 硬件通道,sc	calar_en[0 ~ 3] 分别对应
	Port 1 ~ 30:	不使能 1:使能 scalar	_en [1~3]: 动态属性,若
	不使能则需要抗	巴对应的 Port 销毁掉,	生需要使能时重新创建 Port
reserved[32]	保留,暂不涉及	Ż	

# 【注意事项】

- 1、背景建模功能使能在 ISE 组件运行过程中支持动态设置。
- 2、图像水平翻转和镜像每个 Port 只能使能-

【相关数据类型及接口】

无

# \*ISE\_BGFG\_PARA\_STI\*

【说明】

定义双目拼接模式背景建模相关配置

【定义】

int bgfg\_intvl\*;\*



int getbgd\_intvl\*;\*

int bgfg\_sleep\_ms\*;\*

}ISE\_BGFG\_PARA\_STI\*;\*

【成员】

成员名称	
bgfg_intvl getbgd_intvl bgfg_sleep_ms	每隔多少个 SLEEP_MS 进行一次 model 每隔多少帧获取背景图片 SC16 队列非空时,sleep 多少 MS
W.V.	ALTHER AND
ISE 组件运行过程	中支持动态设置。
及接口 <b>】</b>	中支持动态设置。
*	WINE
式属性	ninderi/

# 【注意事项】

背景建模参数在 ISE 组件运行过程中支持动态设置。

【相关数据类型及接口】

无

\*MODE ATTR\*

【说明】

定义 ISE 组件模式属性

【定义】

typedef union {

FISH mFish\*;\*

DFISH mDFish\*;\*

ISE mIse\*;\*

} MODE ATTR\*;\*

【成员】

成员名称 描述 mode attr 定义 ISE 组件模式属性,选择 ISE 组件工作模式

【注意事项】

文档密级: 秘密



【相关数据类型及接口】

〕无

\**ISE\_*\*\**CHN*\*\*\*\*\*\_ATTR\_S\*\*\*\*

【说明】

定义 ISE 输出端口属性

【定义】

typedef struct

{

MODE\_ATTR mode\_attr\*;\*

int buffer\_num;

} ISE\_CHN\_ATTR\_S\*;\*

【成员】

成员名称 描述

mode\_attr 设置相应模式下的 ISE 端口属性

buffer\_num 设置 ISE 端口 buffer 个数,每个端口 buffer 个数可以单独设置,默认为 5 个

【注意事项】

无

【相关数据类型及接口】

无.

# 5.4 错误码

错误码	宏定义	描述
0xa0e58002	AW_ERR_ISE_INVALID_CHNID	无效 ISE 端口号
0xa0e58003	AW_ERR_ISE_INVALID_PARAM	无效参数
0xa0e58004	AW_ERR_ISE_EXIST	试图申请或创建已经存在的端口或资源
0xa0e58005	AW_ERR_ISE_UNEXIST	试图申请或创建不存在的端口或资源
0xa0e58006	AW_ERR_ISE_NULL_PTR	函数参数中有空指针
0xa0e58007	AW_ERR_ISE_NOT_CONFIG	使用前未配置

版权所有 © 珠海全志科技股份有限公司。保留一切权利

142



A.1V		A1V	
错误码	宏定义	TA TO THE TANK THE THE TANK TH	描述
0xa0e58008	AW_ERR_ISE_NOT_SUPPORT		不支持的参数或功能
0xa0e58009	AW_ERR_ISE_NOT_PERM		该操作不允许执行,如试图修改静态
0xa0e5800c	AW_ERR_ISE_NOMEM		无可用内存
0xa0e5800d	AW_ERR_ISE_NOBUF		无可用缓存
0xa0e5800e	AW_ERR_ISE_BUF_EMPTY		缓存为空
0xa0e58010	AW_ERR_ISE_SYS_NOTREADY		系统没有初始化或没有加载相应模块
0xa0e58012	AW_ERR_ISE_BUSY		设备忙
0xa0e58013	AW_ERR_ISE_TIMEOUT		设备超时
0xa0e58014	AW_ERR_ISE_SAMESTATE		状态相同(常见于状态转换)
0xa0e58015	AW_ERR_ISE_INVALIDSTATE		无效的状态
0xa0e58016	AW_ERR_ISE_INCORRECT_STAT	E_TRANSITION	不正确的状态转换
0xa0e58017	AW_ERR_ISE_INCORRECT_STAT	E_OPERATION	不正确的状态操作

表 4-1 错误码

Will Little White Head of the late of the



版权所有 © 珠海全志科技股份有限公司。保留一切权利

143



# 6.1 概述

VENC 模块,即视频编码模块。本模块支持多路实时编码,且每路编码独立,编码协议和编码 profile 可以不同。

VENC 模块的输入源包括以下二类:

- (1) 用户态读取图像文件向编码模块发送数据;
- (2) 视频输入(VIU)模块采集的图像直接发送到编码模块;

芯片支持的编码规格如下表所示。

的输入源包括以下二类	Ellindeni	15	Blinden		FATARIA VALIMAN
取图像文件向编码模块	发送数据	<b>3</b> ;		,	EXPANSE.
、(VIU)模块采集的图	像直接发	送到编码模块;		(S) AND THE REAL PROPERTY OF THE PERTY OF TH	3
扁码规格如下表所示。		***************************************	16	珠龍	
H.264	JPEG	MOTION JPEG	H.265	MPEG-4	
BP MP HP 支持 支持 支持	支持	支持	MP 支持	不支持	

# 6.2 功能描述

典型的编码流程包括了输入图像的接收、图像的编码、以及码流的输出等过程。通道支持接收 YUV 格式图像输入,支持格式为 semi-planar YUV4:2:0、semi-planar YVU4:2:0、planar YUV4:2:0 以及全志自定义 aw-afbc 格式 (yuv)。通道模块接收外部原始图像数据,而不关心图 像数据是来自哪个外部模块。

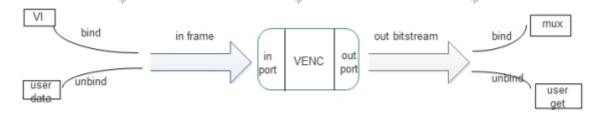


图 6-1: img

每个 venc 通道最多能绑定一个输入端口,一个输出端口。

模块 buffer 使用情况: venc 模块包含输入 frame manager 与输出编码 bit stream man-



ager, 二者的 buffer 使用/处理情况如下。

- 对于输入待编码 frame(yuv): venc 模块本身不对外提供 frame 的 buffer,由 venc 输入端 自已解决输入 frame 的 buffer,venc 模块内部 frame buffer manager 在获取 in frame 后,会通知 venc 输入端 frame 数据已经被处理,可以释放该 frame 的 buffer。对于 venc 模块输入端绑定情况,由 venc 模块跟 venc 模块输入端口组件内部自行完成 frame buffer 的操作交互,不需要调用者再额外处理;对于 venc 输入端口非绑定的情况,此时组件会发出
- MPP\_EVENT\_RELEASE\_VIDEO\_BUFFER 消息,调用者收到此消息后可以自行对 frame buffer 进行释放等操作。
- 对于编码输出 bit stream: 由 venc 模块本身提供输出 bit stream 的 buffer,调用者在获取 完已编码数据后必须及时归还 buffer 给 venc 模块。对于 venc 输出端口绑定方式,由 venc 模块跟 venc 输出端口组件内部自行完成 bit stream buffer 操作交互,不需要再额外处理;对于 venc 输出端非绑定方式,调用者在调用 AW\_MPI\_VENC\_GetStream() 接口成功获取已编码 bit stream 数据 (处理) 后,需调用 AW\_MPI\_VENC\_ReleaseStream() 将 bit stream buffer 还给 venc 模块。

具体使用可参考 sample\_venc(输入输出端口都非绑定)、sample\_venc2muxer(输入端口非绑定输出绑定)、sample\_vi2venc(输入端口绑定输出端口非绑定)、sample\_vi2venc2muxer(输入端口与输出端口都绑定)。

注意事项:模块(通道)注销时,会等待自己所有对外提供的数据 buffer 回收,因此对于绑定方式来讲,要注意各相关模块的注销顺序。如: venc -> mux 方式时,注销时先注销 mux,再注销 venc。

# 6.2.1 缩放功能

编码器支持对输入源进行放大或者缩小编码,宽度和高度可分别最大放大 8 倍,最小缩小 4 倍;实际缩放时的比例根据用户在 VencBaseConfig 设定的输入源的分辨率和编码的分辨率自动计算。输入源的宽度和编码输出的宽度要求 16 对齐,输入源的高度要求 8 对齐,如果其不是 8 对齐,建议对输入源的非 8 对齐部分用最后一行的数据填充。

# 6.2.2 旋转功能

编码器支持对输入源进行旋转编码,旋转角度为顺时针 90 度、180 度、360 度。

# 6.2.3 码率控制

H264 和 H265 编码支持 CBR,VBR,FIXQP,QPMAP 的码率控制方式,MJPEG 编码只支持 CBR 和 FIX\_QUALITY 的码率控制方式。



- CBR: 固定比特率,即在码率统计时间内保证码率平稳,当前默认的码率统计时间是 1s;如果 用户设定的帧率与实际的帧率不一致,则实际的码率与设定的码率与帧率成线性比例;
- VBR:可变比特率,即在码率统计时间内编码码率波动,从而保证编码图像质量稳定;当前通过编码驱动内部统计已编码帧的 mv 信息,对整体运动状况作出估计,为静态场景帧少分配目标比特量,为动态场景帧多分配目标比特量。用户可通过 MotionParam 数据结构设定运动和静止场景的等级,以及运动帧和静止帧占用码率的比例,具体请参看 MotionParam 定义;
- FIXQP: 使用固定 qp 值,在整个编码过程中,所有帧的所有宏块都使用固定的 qp 值,I 帧和 P 帧可以使用不同的固定值;
- QPMAP: 在该模式下,用户可以获取到上一帧的编每个宏块的 qp 信息,通过该信息可控制下一帧编码的每个宏块的 qp 信息。用户可通过 VideoEncSetParameter 的接口 VENC\_IndexParamMBInfoOutput,开启每一帧的编码信息输出功能,包括每一个宏块的 qp 值,mad 值,sse 值,调用该接口时传递数据结构 VencMBInfo 的指针;通过 VideoEncGetParameter 获取上一帧的整帧的 qp、mad、sse 值,接口为VENC\_IndexParamMBSumInfoOutput 数据结构为 VencMBSumInfo;用户可通过VideoEncSetParameter 的接口 VENC\_IndexParamMBModeCtrl 设置下一帧的编码细节,传递的数据结构为 VencMBModeCtrl,在 VencMBModeCtrl 数据结构中有指针指向VencMBModeCtrlInfo 数组,该数组成员包括宏块 qp 值,是否优先使用 skip 预测模式,是否打开该宏块的用户控制使能位,即用户可通过该接口控制每一个宏块的编码模式和 qp 值;

# 6.2.4 裁剪编码

编码器支持从输入源中裁剪一部分进行编码。用户需要设定裁剪区域的起始坐标(x, y),以及裁剪区域的大小(width, height),起始坐标要求 16 对齐,裁剪区域的宽度和高度需要 16 对齐。

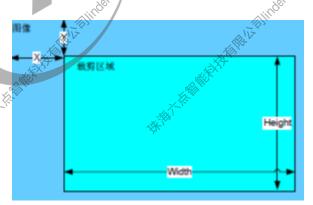


图 6-2: img

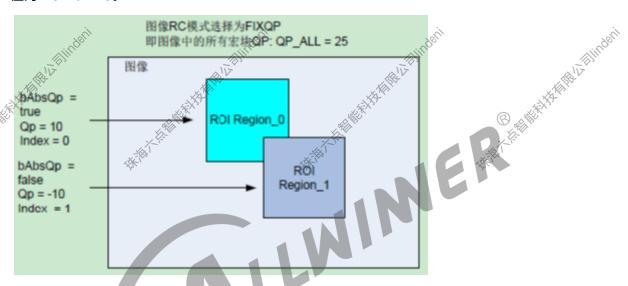
# 6.2.5 ROI 编码

ROI 是(region of interrest)的缩写,即感兴趣区域编码。编码器可以对用户设置的 roi 区域进行特殊编码,roi 的设置包括起始坐标(x,y),区域大小(width,height),以及 QP 设置



(QP 设置分为绝对 QP 和相对 QP,可通过 VencROIConfig 数据结构的 roi\_abs\_flag 使能位进行控制,roi\_abs\_flag 为 1 表示绝对 QP 模式,直接使用该设定的 QP 值作为整个区域的 QP 值,roi\_abs\_flag 为 0 表示相对 QP 模式,使用帧级 QP 值加上该设定的 QP 值作为区域的 QP 值,),编码器共支持 8 个该区域的设置,如果其中有区域重叠的话,则重叠区域使用 index 值大的区域的 QP 值,优先级从低到高依次为 0-7。ROI 的起始坐标和宽度高度均需要 16 对齐。

下图示例编码图像采用 FixQp 模式,设置图像 Qp 为 25,即图像中所有宏块 Qp 值为 25。ROI 区域 0 设置为绝对 Qp 模式,Qp 值为 10,索引为 0;ROI 区域 1 设置为相对 Qp 模式,Qp 为-10,索引为 1。区域 0 的 index 小于区域 1 的 index,所以在发生互相重叠的图像区域按高优先级的区域(区域 1)Qp 设置。区域 0 除了发生重叠的部分的 Qp 值等于 10。区域 1 的 Qp 值为 25-10=15。



# 6.2.6 非 ROI 区域低帧率

非 ROI 区域低帧率编码,即 ROI 区域按正常帧率编码,非 ROI 区域低帧率编码;用户可根据需要设置非 ROI 区域的编码帧率。

# 6.2.7 P 帧帧内刷新

P 帧刷新 ISlice/Intra 宏块行,可以为客户提供码流非常平滑的编码方式,每个 I 帧和 P 帧的大小可以非常接近。在网络带宽有限(如无线网络)的情况下,降低 I 帧过大带来的网络冲击,降低网传延时,降低网络传输出错的概率。

# 6.2.8 osd 叠加

venc 支持 osd 叠加功能, osd 叠加最多可设置 64 个区域,区域之间可以互相重合; osd 叠加包括三种类型,普通叠加功能,将用户指定的 argb 数据叠加到指定的位置上,将用户指定的 argb



数据叠加到指定的位置上并且根据背景的明暗程度反色叠加的图层亮度(即如果背景偏黑色则将 图层叠加为白色,如果背景偏白色则将图层叠加为黑色),将用户指定的 yuv 数据填充到用户指 定的区域中。

各个区域显示由优先级决定,优先级越大显示越在上层。

# 6.2.9 输入数据压缩模式省带宽

V459 venc 支持输入数据为 LBC 模式的压缩数据格式,可有效节省 dram 带宽,提高编码速 度。





# Venc 组件内部状态设定为:

• COMP StateLoaded: 组件初始创建状态。

• COMP StateIdle: 组件完成初始化,参数设置、资源配置完毕,随时可以运行的状态。

• COMP StateExecuting: 运行状态。

• COMP StateInvalid: 异常状态。

API AW\_MPI\_VENC\_CreateChn() 的实现过程会经过 COMP\_StateLoaded 状态,到达 COMP\_StateIdle。

组件内部状态转换的函数是: SendCommand(..., COMP\_CommandStateSet, 目标COMP\_State, ...);

# 6.3.2 API 和状态

每个 API 只能在允许的状态下调用,如果不在允许的状态下调用 API,则无效

API 列表如下: (允许被调用的状态栏填写 Y)

COMD	C+MAIMID	State Execution at the Hard in State Execution at the Hard in the State Execution at the St
COMI	SIGNATION	Stateurwerdintibildienerrendin

AW MPI VENC CreateChn

AW MPI VENC DestroyChn

AW MPI VENC SetChnlPrioYity

AW MPI VENC GetChnlPriority

AW MPI VENC ResetChn

AW MPI VENC StartRecvPic

AW\_MPI\_VENC\_StartRecvPicEx

AW VENC StopRecvPic Y

AW MPI VENC Query Y

AW\_MPI\_VENC\_SetChnAttrY

AW MPI VENC G&tChnAttrY

AW MPI VENC GetStream Y

AW MPI VENC ReleaseStream

AW MPI VENC In VertUser Data

AW MPI VENC SendFrameY

引起状态转换。创建组件,完成后 状态为 COMP\_StateIdle

销毁组件

重置组件到初始化状态。

引起状态转换。到 Executing。

引起状态转换。到 Executing,再

恢复到 Idle。

引起状态转换。到 Idle。

设置编码通道属性,只能设置动态

属性。

获取编码通道属性。

获取编码后的码流。只能用于非绑

定模式。

归还码流。只能用于非绑定模式。

插入用户数据,编码为 SEI 单元。 发送待编码的图像。只能用于非绑

定模式。



# 

AW MPI VENC SetMaxStreYmDuration

AW\_MPI\_VENC@GetMaxStreamDuration

AW MPI VENC RequestIDRY

AW MPI VENC SetRoiCfg Y

AW MPI VENC GetRoiCfg Y

AW MPI VENC SetH264Sli&eSplit

AW MPI VENC G&tH264Sli&eSplit

AW MPI VENC SetH264InterPred

AW\_MPI\_VENC\_GetH264InterPred

AW\_MPI\_VENC\_SetH264IntMaPred

AW MPI VENC GetH264IntraPred

AW MPI VENC SettH264Trayns

AW MPI VENC GetH264Trans

AW MPI VENC SetH264Entropy

AW MPI VENC GetH264Entropy

AW\_MPI\_VENC\_SetH264Po&

AW MPI VENC G&tH264PoX

AW\_MPI\_VENC\_SetH264Dblk

AW\_MPI\_VENC GetH264Dblk

AW MPI VENC SetH264VuiY

AW MPI VENC Gath264Vul

AW MPI VENC SetJpegParam

AW MPI VENC GattpegParam

AW MPI VENC SetMJpegParam

AW MPI VENC GetMJpegParam

AW MPI VENC SetFrameRate

AW MPI VENC GetFrameRate

AW\_MPI\_VENC\_SetRcParamY

AW MPI VENC GetRcParany

AW MPI VENC SetRefParan

AW MPI VENC GetRefParam

AW MPI VENC SetColor2GYey

AW MRI VENC GatColor2GYev

AW MPI VENC SetCrop

AW MPI VENC GetCrop

设定编码 buffer 的最大缓冲时

0ل<sup>ر</sup>

获取编码 buffer 的最大缓冲时

间。

立即编码 I 帧。

设定 ROI 区域

获取 ROI 区域

设置 slice 分割属性

获取 slice 分割属性

设置熵编码模式

获取熵编码模式

设置 H264 编码 POC 类型。

获取 H264 编码 POC 类型。

设置 H264 编码的 Deblocking

类型

获取 H264 编码的 Deblocking

类型

设置 H264 的 VUI 参数。

获取 H264 的 VUI 参数。

设置 IPEG 编码参数。

获取 IPEG 编码参数。

设置 MIPEG 编码参数。

获取 MJPEG 编码参数。

设置编码通道的帧率控制属性。

获取编码通道的帧率控制属性。

设置编码通道的码率控制属性

获取编码通道的码率控制属性

设置 h264/h265 编码通道高级跳

帧参考参数

获取 h264/h265 编码通道高级跳

帧参考参数

开启或关闭一个通道的彩转灰功能

获取一个通道是否开启彩转灰功能



# COMP StateMile State Execution at the Hyalid

AW MPI VENC SettlpegSnan/Mode AW MPI VENC Get pegSnap Mode

AW MPI VENCE YableIDR Y

AW MPI VENC GetStreamBufInfo

AW MPI VENC SetRcPriority AW MPI VENC GetRcPriorily

AW MPI VENC SetH265Sli&eSplit

AW MPI VENC GetH265SliveSplit

AW\_MPI\_VENC\_SetH265PredUnit

AW MPI VENC GatH265PradUnit

AW MPI VENC SettH265Trams

AW MPI VENC Gath 265Trains

AW MPI VENC SetH265Entropy

AW MPI VENC GetH265Entropy

AW\_MPI\_VENC SetH265Dblk

AW MPI VENC GetH265Dblk

AW MPI VENC SetH265SaY

AW MPI VENC GetH265Say

AW MPI VENC SetH265Timing

AW MPI VENC GMtH265Tinking

C With the light the light to t AW MPI VENC SetFrameLostStrategy

AW MPI VENC GatFrameLastStrategy

AW MPI VENC SetSuperFrameCfg

AW MPI VENC GetSuperFrameCfg

AW MPI VENC SetIntraReffesh

AW MPI VENC GetIntraRefYesh

获取码流 buffer 的物理地址和大 小。即 VBVBuffer 的整体大小。 设置码率控制的优先级类型。 获取码率控制的优先级类型。 设置 h265 分割属性。如果是按 LCU line 分割, LCU line 最大 值: (图像高 +63)/64。

# 6.3.3 API 参考

视频编码模块主要提供视频编码通道的创建和销毁、视频编码通道的复位、开启和停止接收图 像、设置和获取编码通道属性、获取和释放码流等功能。

# 6.3.3.1 AW MPI VENC CreateChn

【目的】



# 创建一个编码通道

# 【语法】

ERRORTYPE AW MPI\_VENC\_CreateChn(VENC\_CHN VeChn, const VENC\_CHN\_ATTR\_S \*pAttr)

# 【参数】

参数	描述	输入/输出
VeChn	通道 ID 号,范围 [0, VENC_MAX_CHN_NUM)	 输入
pAttr	编码通道属性	输入

<u> </u>			
【返回值】	.Xi	JR 12 Blinde	描述 成功 提供機能
A STATE OF THE STA	AXXX.	返回值	描述
ALIA CONTRACTOR OF THE PARTY OF	- Little Hall	0	成功
	操稿	非 0	失败,其值见错误码。
【需求】 • 头文件: mr	n comm venc.l	n, mm	common h
	media_mpp.so		
	media_mpp.so		
【注意】		. de	ari / Hilled Tallinderi
<ul><li>创建后状态:</li></ul>	为 COMP_State	Idle。	
No.	The state of the s	NO.	

# 【举例】

# 6.3.3.2 AW\_MPI\_VENC\_DestroyChn

# 【目的】

销毁一个编码通道

# 【语法】

ERRORTYPE AW MPI VENC DestroyChn(VENC CHN VeChn)

# 【参数】



参数 描述

通道 ID 号, 范围: [0, VENC MAX\_CHN\_NUM) VeChn

【返回值】

返回值	描述	
0 非 0	成功 失败,	其值见错误码。

【需求】

Sp. 美文件: mm\_comm\_venc.h mm\_common.h

库文件: libmedia\_mpp.so

【注意】

无。

【举例】

无。

6.3.3.3 AW\_MPI\_VENC\_ResetChn

【目的】

重置编码通道到初始化状态

【语法】

ERRORTYPE AW MPI VENC ResetChn(VENC CHN VeChn)

【参数】

描述 参数 通道 ID 号。范围: [0, VENC MAX CHN NUM) VeChn





返回值 描述 0 成功 非 0 失败,其值见错误码。

【需求】

• 头文件: mm\_comm\_venc.h、mm\_common.h

• 库文件: libmedia\_mpp.so

【注意】

6.3.3.4 AW\_MPI\_VENC\_StartRecvPic
【目的】
FP始编码
【语法】
RRORTYPE

ERRORTYPE AW\_MPI\_VENC\_StartRecvPic(VENC\_CHN VeChn)

【参数】

描述 参数

通道 ID 号。范围: [0, VENC\_MAX\_CHN\_NUM) 输入 VeChn

【返回值】

描述 返回值 0 成功 非 0 失败,其值见错误码。



头文件: mm\_comm\_venc.h mm\_common.h

• 库文件: libmedia mpp.so

# 【注意】

• 引起状态转换,切换到 Executin。

# 【举例】

无。

# AW\_MPI\_VENC\_StartRecvPicEx

# 【目的】

启动编码,并且编码指定的帧数。

# 【语法】

Company of the state of the sta ERRORTYPE AW MPI VENC StartRecvPicEx(VENC CHN VeChn, VENC RECV PIC PARAM S \*pRecvParam);

# 【参数】

参数	描述	<u>,                                     </u>	
VeChn	通道 ID 号。范围:	[0, VENC_MAX_CHN_NUM)	输入
pRecvParam	指定接收帧数。	A IV	输入

# 【返回值】

返回值	描述
0	成功
非 0	失败,其值见错误码。

# 【需求】

- 头文件: mm comm venc.h、mm common.h
- 库文件: libmedia\_mpp.so

# 注意】



• 引起状态转换,切换到 Executing。

【举例】

无。

# ${\bf 6.3.3.6 \quad AW\_MPI\_VENC\_StopRecvPic}$

【目的】

停止编码

【语法】

ERRORTYPE AW\_MPI\_VENC\_StopRecvPic(VENC\_CHN VeChn)

【参数】

参数 描述

VeChn 编码通道号。范围: [0, VENC\_MAX\_CHN\_NUM) 输入

【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,其值见错误码。

【需求】

• 头文件: mm comm venc.h、mm common.h

• 库文件: libmedia mpp.so

【注意】

• 引起状态转换,切换到 Idle 状态。

【举例】

无。



# 6.3.3.7 AW\_MPI\_VENC\_Query

【目的】

查询编码通道状态

【语法】

ERRORTYPE AW\_MPI\_VENC\_Query(VENC\_CHN VeChn, VENC\_CHN\_STAT\_S \*pStat)

# 【参数】

参数	描述	deni		indeni		
VeChn	编码通道号。范	· 古围:[0, V	VENC_	MAX_CHN_NUM)	输入	ALIV TO THE REAL PROPERTY.
pStat	状态			NA TOP TO THE PERSON NAMED IN COLUMN	输出	A TANKS

# 【返回值】

返回值	描述	
0	成功	
非 0	失败,其值见错误码。	5

# 【需求】

- 头文件: mm comm venc.h、mm common.h
- 库文件: libmedia mpp.so

【注意】

无。

【举例】

无。

# ${\bf 6.3.3.8 \quad AW\_MPI\_VENC\_RegisterCallback}$

【目的】

设置编码通道回调函数。



# 【语法】

ERRORTYPE AW\_MPI\_VENC\_RegisterCallback(VENC\_CHN VeChn, MPPCallback-Info \*pCallback)

# 【参数】

参数	描述		
VeChn	编码通道号。	范围: [0, VENC_MAX_CHN_NUM)	输入
pCallback	回调参数		输入

# 【返回值】

返回值 描述

0 成功
非 0 失败,其值见错误码。

# 【需求】

- 头文件: mm comm venc.h、mm common.h
- 库文件: libmedia\_mpp.so

# 【注意】

无。

【举例】

无。

# 6.3.3.9 AW\_MPI\_VENC\_SetChnAttr

【目的】

设置编码通道属性

【语法】

ERRORTYRE AW\_MPI\_VENC\_SetChnAttr(VENC\_CHN VeChn, const VENC\_CHN\_ATTR\_S \*pAttr)

【参数】



参数 描述

编码通道号。范围: [0, VENC MAX\_CHN\_NUM) VeChn 通道属性 输入 pAttr

【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

E Refer to the state of the sta 头文件: mm comm venc.h、mm common.h

• 库文件: libmedia mpp.so

【注意】

• 只能设置动态属性。

【举例】

无。

AW\_MPI\_VENC\_GetChnAttr

【目的】

获取编码通道属性

【语法】

ERRORTYPE AW MPI VENC GetChnAttr(VENC CHN VeChn, VENC CHN ATTR S \*pAttr)

【参数】

<b>少</b> 数	抽处 "%"	ninde.	
VeChn	编码通道号。范围:	[0, VENC_MAX_CHN_NUM)	输入
pAttr	通道属性	A TOTAL CONTRACTOR OF THE PARTY	输出



# 【返回值】

λ-X *		LeX '
A SHAPE TO S	返回值	描述
	0 非 0	成功失败,其值见错误码。

# 【需求】

• 头文件: mm comm venc.h、mm common.h

• 库文件: libmedia\_mpp.so

# 

【语法】

ERRORTYPE AW MPI\_VENC\_GetStream(VENC\_CHN VENC\_STREAM\_S VeChn, \*pStream, int nMilliSec)

# 【参数】

参数	描述	输入/输出
VeChn	编码通道号。范围:[0, VENC_MAX_CHN_NUM)	输入
pStream	编码后的码流	输出
nMilliSec	获取数据的超时时间。-1 表示阻塞模式;0 表示非阻塞模式;>0	输入
	表示阻塞 s32MilliSec 毫秒,超时则报错返回。取值范围: (0,	
	+∞)	

【返回值】



返回值	描述	
0	成功	# <b>/</b> # \
非 0	失败,	其值见错误码。

【需求】

• 头文件: mm\_comm\_venc.h、mm\_common.h

• 库文件: libmedia\_mpp.so

【注意】

仅限于 venc 组件非绑定模式。与 AW\_MPI\_VENC\_ReleaseStream 必须成对使用,否则编 码器 bit stream 内存不会被释放。 MER

【举例】

无。

# 6.3.3.12 AW\_MPI\_VENC\_ReleaseStream

【目的】

释放已获取的编码器编码码流(内存)

【语法】

ERRORTYPE AW\_MPI\_VENC\_ReleaseStream(VENC\_CHN VeChn, VENC\_STREAM\_S \*pStream)

【参数】

参数	描述		
VeChn	编码通道号。范围: [0, VENC_MAX_CHN_NUM)	输入	
pStream	通过 getstream 获取的编码码流		

【返回值】

返回值	描述	all allino
0	成功	



返回值 描述

失败,其值见错误码。 非 0

【需求】

• 头文件: mm comm venc.h、mm common.h

• 库文件: libmedia mpp.so

【注意】

仅限于 venc 组件非绑定模式。与 AW\_MPI\_VENC\_GetStream 必须成对使用,否则编码器 bit stream 内存不会被释放。 INE

【举例】

无。

# 6.3.3.13 AW\_MPI\_VENC\_SendFrame

【目的】

向编码器传送待编码图像数据帧

【语法】

ERRORTYPE AW MPI\_VENC\_SendFrame(VENC\_CHN VeChn, VIDEO\_FRAME\_INFO\_S \*pFrame ,int nMilliSec)

# 【参数】

参数 描述

VeChn 编码通道号。范围: [0, VENC MAX CHN NUM)

pFrame 图像数据帧

发送数据的超时时间。-1表示阻塞模式; 0表示非阻塞模式; >0表示阻塞 s32MilliSec 毫秒,超图 nMilliSec

【返回值】

返回值 描述 成功



返回值 描述 非 0 失败,其值见错误码。

【需求】

• 头文件: mm comm venc.h、mm common.h

• 库文件: libmedia mpp.so

【注意】

ERRORTYPE AW\_MPI\_VENC\_RequestIDR(VENC\_CHN VeChn, BOOL bInstant)

【参数】

参数	描述				- <u>1</u>	
VeChn	编码通道号。	范围:	[0, VENC	_MAX_CHN	NUM)	输入
bInstant	未使用。					输入

【返回值】

返回值 描述 成功 失败,其值见错误码



#### 【需求】

• 头文件: mm\_comm\_venc.h、mm\_common.h

• 库文件: libmedia\_mpp.so

【注意】

无。

【举例】

无。

【返回值】

描述 返回值

句柄编号 >= 0

< 0 失败,其值见错误码。

#### 【需求】

• 头文件: mm\_comm\_venc.h、mm\_common.h

库文件: libmedia\_mpp.so

【注意】



无必

【举例】

无。

### 6.3.3.16 AW\_MPI\_VENC\_SetRoiCfg

【目的】

设置编码器感兴趣区域

【语法】

ERRORTYPE AW\_MPI\_VENC\_SetRoiCfg(VENC\_CHN\_VeChn, VENC\_ROI\_CFG\_S \*pVencRoiCfg)

#### 【参数】

参数	描述	*******	12 4	输入/输出
VeChn	编码通道号。范围	围:[0, VENC_MAX_	CHN_NUM)。	输入
pVencRoiCfg	感兴趣区域			输入

#### 【返回值】

返回值	描述	
0 llinder	成功	
非 0	失败,	其值见错误码。

#### 【需求】

• 头文件: mm\_comm\_venc.h、mm\_common.h

• 库文件: libmedia mpp.so

【注意】

无。

【举例】

无。



## 6.3.3.17 AW\_MPI\_VENC\_GetRoiCfg

【目的】

获取编码器感兴趣区域

【语法】

ERRORTYPE AW\_MPI\_VENC\_GetRoiCfg(VENC\_CHN VeChn, unsigned int nIndex, VENC ROI CFG S \*pVencRoiCfg)

#### 【参数】

参数	描述 ····································	inderi	
VeChn	编码通道号,范围	: [0, VENC_MAX_CHN_NUM)。	输入
pVencRoiCfg	感兴趣区域	A TOP TO THE PERSON NAMED IN COLUMN	输出

【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,其值见错误码。

【需求】

- 头文件: mm comm venc.h、mm common.h
- 库文件: libmedia mpp.so

【注意】

无。

【举例】

无。

### ${\bf 6.3.3.18} \quad AW\_MPI\_VENC\_SetRoiBgFrameRate$

【目的】

设置感兴趣区域编码的背景帧率



#### 【语法】

ERRORTYPE AW\_MPI\_VENC\_SetRoiBgFrameRate(VENC\_CHN VeChn, const VENC\_ROIBG\_FRAME\_RATE\_S \*pstRoiBgFrmRate);

#### 【参数】

参数	描述	
VeChn	编码通道号,范围: [0, VENC_MAX_CHN_NUM)。	输入
pstRoiBgFrmRate	背景帧率信息	

#### 【返回值】

返回值 描述

0 成功
非 0 失败,其值见错误码。

#### 【需求】

- 头文件: mm\_comm\_venc.h、mm\_common.h
- 库文件: libmedia\_mpp.so

#### 【注意】

无。

【举例】

## 6.3.3.19 AW\_MPI\_VENC\_GetRoiBgFrameRate

#### 【目的】

获取感兴趣区域编码的背景帧率。

#### 【语法】

ERRORTYPE AW\_MPI\_VENC\_GetRoiBgFrameRate(VENC\_CHN VeChn, VENC\_ROIBG\_FRAME\_RA'\* \*pstRoiBgFrmRate);



参数	描述			Set.		XA KINGO
VeChn	编码通道号,	范围:	[0, VENC	MAX_CHI	N_NUM)。	输入
pstRoiBgFrmRate	背景帧率信息				-162	输出

返回值	描述	
0 非 0	成功 失败,	其值见错误码。

Will the state of 头文件: mm comm venc.h、mm common.h

• 库文件: libmedia mpp.so

#### 【注意】

无。

【举例】

无。

## AW\_MPI\_VENC\_GetH264SpsPpsInfo

#### 【目的】

获取编码器 H264 编码的 spspps 头信息

#### 【语法】

ERRORTYPE AW\_MPI\_VENC\_GetH264SpsPpsInfo(VENC\_CHN VeChn, VencHeader-Data\*pH264SpsPpsInfo)

参数	描述		
VeChn	编码通道号,范围:	[0, VENC_MAX_CHN_NUM)。	输入
pH264SpsPpsInfo	spspps 信息	-Alino	输出



λ-X *		4-8
A STATE OF THE STA	返回值	描述
The state of the s	0 非 0	成功失败,其值见错误码。

【需求】

• 头文件: mm comm venc.h、mm common.h

• 库文件: libmedia\_mpp.so

【注意】

【举例】

无。

# fo 6.3.3.21 AW\_MPI\_VENC\_GetH265SpsPpsInfo

【目的】

获取编码器 H265 编码的 spspps 头信息

【语法】

ERRORTYPE AW\_MPI\_VENC\_GetH265SpsPpsInfo(VENC\_CHN VeChn, VencHeader-Data \*pH264SpsPpsInfo)

#### 【参数】

参数	描述	输入/输出
VeChn	编码通道号,范围: [0, VENC_MAX_CHN_NUM)。	输入
pH265SpsPpsInfo	spspps 信息	输出

#### 【返回值】

返回值	描述	.8
QIV.	成功	
<b>手 0</b>	失败,	其值见错误码。



#### 【需求】

• 头文件: mm\_comm\_venc.h、mm\_common.h

• 库文件: libmedia\_mpp.so

【注意】

无。

【举例】

无。

## AW\_MPI\_VENC\_SetJpegParam

#### 【目的】

设置 JPEG 协议编码通道的高级参数。

#### 【语法】

CHN' ERRORTYPE AW\_MPI\_VENC\_SetJpegParam(VENC\_CHN VeChn, const VENC\_PARAM\_JPEG\_S \*pJpegParam)

#### 【参数】

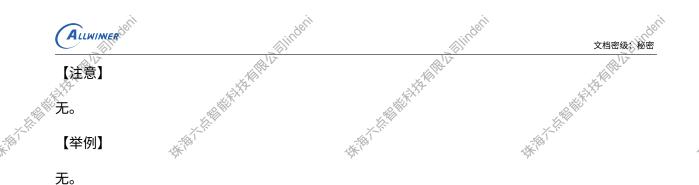
参数	描述	.nderi	.nderil
VeChn	编码通道号,范围	: [0, VENC_MAX_CHN_NUM)。	输入
pJpegParam	jpeg 编码参数	A CONTRACTOR OF THE PARTY OF TH	输入

#### 【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

#### 【需求】

- 头文件: mm\_comm\_venc.h、mm\_common.h
- 库文件: libmedia\_mpp.sc



## ${\bf 6.3.3.23 \quad AW\_MPI\_VENC\_GetJpegParam}$

【目的】

获取编码器 jpeg 编码参数

【语法】

ERRORTYPE AW\_MPI\_VENC\_SetJpegParam(VENC\_CHN VeChn, const VENC\_PARAM\_JPEG\_S \*pJpegParam)

【参数】

参数	描述	1188	
VeChn	编码通道号,范围:	[0, VENC_MAX_CHN_NUM)。	输入
pJpegParam	jpeg 编码参数	Ma.	输出

【返回值】

返回值 描述 0 成功 非 0 失败,其值见错误码。

【需求】

• 头文件: mm\_comm\_venc.h、mm\_common.h

• 库文件: libmedia\_mpp.so

【注意】

无。

【至例】

池。



# 6.3.3.24 AW\_MPI\_VENC\_SetJpegExifInfo

【目的】

设置编码器 jpegExif 编码参数

【语法】

 $ERRORTYPE\ AW\_MPI\_VENC\_SetJpegExifInfo(VENC\_CHN\ VeChn,\ const\ VENC\_EXIFINFO\_S*pJpegExifInfo(VENC\_CHN\ VeChn,\ const\ VeChn$ 

#### 【参数】

<del></del> 参数	描述		
VeChn	编码通道号,范围:	[0, VENC_MAX_CHN_NUM)。	 输入
pJpegExifInfo	jpegExif 编码参数	NA IV	输入

【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,其值见错误码。

#### 【需求】

- 头文件: mm\_comm\_venc.h、mm\_common.h
- 库文件: libmedia\_mpp.so

【注意】

无。

【举例】

无。

## ${\bf 6.3.3.25 \quad AW\_MPI\_VENC\_GetJpegExifInfo}$

【目的】

获取编码器 jpegExif 编码参数

【语法】



ERRORTYPE AW\_MPI\_VENC\_SetJpegExifInfo(VENC\_CHN VeChn, const VENC\_EXIFINFO\_S\*pJpeg

#### 【参数】

参数	描述	W. T.	茶梅	
VeChn	编码通道号,	范围: [0, VENC_M	AX_CHN_NUM)。	输入
pJpegExifInfo	jpegExif 编	码参数		输出

#### 【返回值】

返回值	描述	-	
0 0	成功	- Reli	deni
非0	成功 失败,其值见错误码。		III.
	大败,其值见错误的。 common.h	A C REAL PROPERTY AND A SECOND PORCE OF THE PROPERTY OF THE PR	
der	<b>&gt;</b> /	sei.	deni

【需求】

• 头文件: mm\_comm\_venc.h、mm\_common.h

• 库文件: libmedia\_mpp.so

【注意】

无。

【举例】

无。

6.3.3.26 AW\_MPI\_VENC\_GetJpegThumbBuffer

【目的】

获取编码器 jpeg 缩略图编码 buffer

【语法】

 $ERRORTYPE\ AW\_MPI\_VENC\_GetJpegThumbBuffer(VENC\_CHN\ VeChn,\ VENC\_JPEG\_THUMB\_BUFFDThumbBuffer)$ 



参数	描述		
VeChn	编码通道号,范	围: [0, VENC_MAX_C	HN_NUM)。 输入
pThumbBuffer	缩略图 buffer	-100	输出

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

#### 【需求】

White Party to the state of the 头文件: mm comm venc.h、mm common.h

• 库文件: libmedia mpp.so

#### 【注意】

无。

【举例】

无。

# AW\_MPI\_VENC\_SetFrameRate

#### 【目的】

设置编码通道帧率控制属性

#### 【语法】

 $ERRORTYPE\ AW\_MPI\_VENC\_SetFrameRate (VENC\_CHN\ VeChn,\ const\ VENC\_FRAME\_RATE\_S$ \*pFrameRate)

参数	描述		
VeChn	编码通道号,范围:	[0, VENC_MAX_CHN_NUM)。	输入
pFrameRate	帧率属性	Alline	输入

	10,0
( ALL	V/WER

V-Y.		4-X
THE THE PARTY OF T	返回值	描述
	0 <b>1</b> € 0	成功失败,其值见错误码。
	∃F U	大败,共阻见垣误吗。

【需求】

• 头文件: mm\_comm\_venc.h、mm\_common.h

• 库文件: libmedia\_mpp.so

【注意】

【举例】

无。

【目的】

获取编码通道帧率控制属性

【语法】

ERRORTYPE AW\_MPI\_VENC\_GetFrameRate(VENC\_CHN VeChn, const VENC\_FRAME\_RATE\_S \*pFrameRate)

【参数】

参数	描述	
VeChn	编码通道号,范围: [0, VENC_MAX_CHN_NUM)。	输入
pFrameRate	帧率属性	输出

#### 【返回值】

返回值	描述	.00
O'V	成功	
<b>準 0</b>	失败,	其值见错误码。



#### 【需求】

• 头文件: mm\_comm\_venc.h、mm\_common.h

• 库文件: libmedia\_mpp.so

【注意】

无。

【举例】

无。

## 6.3.3.29 AW\_MPI\_VENC\_SetTimeLapse

#### 【目的】

设置编码通道缩时/慢摄影状态下,用于编码的帧间隔时间。

#### 【语法】

 $\label{lem:error} ERRORTYPE \ AW\_MPI\_VENC\_SetTimeLapse(VENC\_CHN \ VeChn, \ int 64\_t \ nTime-Lapse)$ 

#### 【参数】

参数	描述	deri		nderi!		输入/输出
VeChn	编码通道号,	范围: [0, VE	NC_MAX_	CHN_NUM)。		输入
nTimeLapse	编码帧的间隔	扇时间,单位微	秒。设置后,	编码通道根据编码	马帧间隔	输入
XXX	时间选定输	<b>、</b> 帧进行编码,统	编码后的输出	出帧的 PTS 根据编	码通道	
	目标输出帧率	率重新设置,从	0 开始,以	确保录制的文件播	放时按	
	照输出帧率)	进行播放。静态/	属性。			

#### 【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

## 【需求】

◆ 头文件: mm\_comm\_venc.h、mm\_common.h



• 库文件: libmedia\_mpp.so

【注意】

无。

【举例】

无。

#### 6.3.3.30 AW\_MPI\_VENC\_GetTimeLapse

【目的】

获取编码通道缩时/慢摄影帧间隔时间

【语法】

ERRORTYPE AW\_MPI\_VENC\_GetTimeLapse(VENC\_CHN VeChn, int64\_t \*pTime-Lapse)

#### 【参数】

 参数	描述	
VeChn	编码通道号,范围:[0, VENC_MAX_CHN_NUM)。	 输入
pTimeLapse	编码帧的间隔时间,单位微秒。设置后,编码通道根据编码帧间隔时	输出
is	间选定输入帧进行编码,编码后的输出帧的 PTS 根据编码通道目标输	
Allinderii	出帧率重新设置,从0开始,以确保录制的文件播放时按照输出帧率	llind
THE LET	进行播放。	ARAIN .

#### 【返回值】

返回值	描述
0	成功
非 0	失败,其值见错误码。

#### 【需求】

- 头文件: mm\_comm\_venc.h、mm\_common.h
- 库文件: libmedia\_mpp.so

# 注意】



#### 6.3.3.31 AW\_MPI\_VENC\_SetColor2Grey

【目的】

开启或关闭一个通道的彩转灰功能。

【语法】

ERRORTYPE AW\_MPI\_VENC\_SetColor2Grey(VENC\_CHN VeChn, const VENC\_COLOR2GREY\_S \*pChnColor2Grey);

#### 【参数】

参数	K. K.	描述	******			tite it	
VeChn		编码通道号,	范围: [0,	VENC_MAX_	CHN_	NUM)。	输入
pChnCol	or2Grey	彩转灰配置信	言息。	1112			输入

【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,其值见错误码。

【需求】

【注意】

无。

【举例】

无。

### 6.3.3.32 AW\_MPI\_VENC\_GetColor2Grey

【目的】



获取一个通道是否开启彩转灰功能。

#### 【语法】

ERRORTYPE AW MPI\_VENC\_GetColor2Grey(VENC\_CHN VeChn, VENC\_COLOR2GREY\_S \*pChnColor2Grey);

#### 【参数】

参数	描述	
VeChn	编码通道号,范围: [0, VENC_MAX_CHN_NUM)。	输入
pChnColor2Grey	获取开启或关闭彩转灰功能的参数。	输入

【返回值】

返回值 描述 0 成功 其值见错误码。

【需求】

【注意】

无。

【举例】

无。

6.3.3.33 AW\_MPI\_VENC\_SetCrop

【目的】

设置编码通道编码裁剪区域

【语法】

 $\label{lem:const} ERRORTYPE\ AW\_MPI\_VENC\_SetCrop(VENC\_CHN\ VeChn,\ const\ VENC\_CROP\_CFG\_S\ *pCropCfg)$ 



参数 描述

编码通道号,范围: [0, VENC\_MAX\_CHN\_NUM)。 VeChn pCropCfg、裁剪区域。动态属性。 输入

【返回值】

返回值	描述	
0 非 0	成功 失败,	其值见错误码。

White Party of the state of the 头文件: mm comm venc.h、mm common.h

• 库文件: libmedia mpp.so

【注意】

无。

【举例】

无。

AW\_MPI\_VENC\_GetCrop

【目的】

获取编码通道编码裁剪区域

【语法】

ERRORTYPE AW\_MPI\_VENC\_GetCrop(VENC\_CHN VeChn, VENC\_CROP\_CFG\_S \*pCropCfg)

【参数】

参数	描述		
VeChn	编码通道号,范围:	[0, VENC_MAX_CHN_NUM)。	输入
	裁剪区域(	- Alline	输出

版权所有 © 珠海全志科技股份有限公司。保留一切权利



A STATE OF THE STA	返回值	描述
A THE	0 非 0	成功失败,其值见错误码。

#### 【需求】

• 头文件: mm\_comm\_venc.h、mm\_common.h

• 库文件: libmedia\_mpp.so

【注意】

【举例】

无。

AW\_MPI\_VENC\_SetIntraRefresh

设置刷 I 宏块的参数

#### 【目的】

P 帧帧内刷新。设置刷 I 宏块的参数。

【语法】

ERRORTYPE AW\_MPI\_VENC\_SetIntraRefresh(VENC\_CHN VeChn, VENC\_PARAM\_INTRA\_REFRES \*pIntraRefresh)

#### 【参数】

参数	描述	
VeChn	编码通道号,范围: [0, VENC_MAX_CHN_NUM)。	输入
pIntraRefresh	P 帧帧内刷新,刷 I 宏块的设置参数。	输入

#### 【返回值】

返回值	描述	.00
O'V	成功	
<b>準 0</b>	失败,	其值见错误码。



#### 【需求】

• 头文件: mm\_comm\_venc.h、mm\_common.h

• 库文件: libmedia\_mpp.so

【注意】

无。

【举例】

无。

## 6.3.3.36 AW\_MPI\_VENC\_GetIntraRefresh

【目的】

【语法】

ERRORTYPE AW\_MPI\_VENC\_SetIntraRefresh(VENC\_CHN VeChn, VENC\_PARAM\_INTRA\_REFRES \*pIntraRefresh)

#### 【参数】

参数	描述	
VeChn	编码通道号,范围: [0, VENC_MAX_CHN_NUM)。	输入
pIntraRefresh	P 帧帧内刷新,刷 I 宏块的设置参数。	输出

#### 【返回值】

返回值	描述
0	成功
非 0	失败,其值见错误码。

#### 【需求】

- 头文件: mm\_comm\_venc.h、mm\_common.h
- 库文件: libmedia\_mpp.so

# 注意】

【举例】

无。

#### 6.3.3.37 AW\_MPI\_VENC\_SetSmartP

【目的】

P帧 smart 编码参数。

【语法】

ERRORTYPE AW\_MPI\_VENC\_SetSmartP(VENC\_CHN\_VeChn, VencSmartFun\_\*pSmartPParam)

【参数】

参数	描述	W. Commission of the Commissio		
VeChn	编码通道号,	范围: [0, VENC_MA	X_CHN_NU	M)。 输入
pSmartPParam	smart 编码都	参数。		输入

【返回值】

返回值 描述 成功 **#**0 失败,其值见错误码。

【需求】

【注意】

无。

【举例】

无。

## 6.3.3.38 AW\_MPI\_VENC\_GetSmartP

【目的】



获取 smart 编码参数。

#### 【语法】

ERRORTYPE AW\_MPI\_VENC\_GetSmartP(VENC\_CHN VeChn, VencSmartFun \*pS-martPParam)

#### 【参数】

<del></del> 参数	描述	输入/输出
VeChn	编码通道号,范围: [0, VENC_MAX_CHN_NUM)。	 输入
pSmartPParam	smart 编码参数。	输出

#### 【返回值】

RIV.	可值	描述 成功 失败,其值见错误码。	
返[	回值	描述	
0		成功	
非	0	失败,其值见错误码。	
	deri	LMIN	

【需求】

【注意】

无。

【举例】

无。

## 6.3.3.39 AW\_MPI\_VENC\_SetBrightness

#### 【目的】

配置 h264 和 h265 编码的亮暗阈值属性,和 smart 编码配合使用。

#### 【语法】

ERRORTYPE AW\_MPI\_VENC\_SetBrightness(VENC\_CHN VeChn, VencBrightnessS \*pBrightness);



参数	描述		A TANK	7		X
VeChn	编码通道号,	范围:	[0, VENC_MA	X_CHN	_NUM)。	输入
pBrightness	亮暗阈值属性。	<b>)</b>	-100		-123	输入

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

【需求】

【注意】

无。

【举例】

无。

# S Report of the state of the st 6.3.3.40 AW\_MPI\_VENC\_GetBrightness

【目的】

获取亮暗阈值属性。

【语法】

ERRORTYPE AW\_MPI\_VENC\_GetBrightness(VENC\_CHN\_VeChn, VencBrightnessS \*pBrightness)

#### 【参数】

 参数	描述	输入/输出
VeChn	编码通道号,范围: [0, VENC_MAX_CHN_NUM)。	输入
pSmartPParam	亮暗阈值属性。	输出

【返回值】

返回值	描述	a liz
<b>20</b>	成功	- A A A A A A A A A A A A A A A A A A A



返回值 描述

失败,其值见错误码。 非 0

【需求】

【注意】

无。

【举例】

无。

## AW\_MPI\_VENC\_SetVEFreq

【目的】

设置编码引擎时钟频率。

【语法】

Ver ERRORTYPE AW\_MPI\_VENC\_SetVEFreq(VENC\_CHN VeChn, int nFreq); //nFreq: MHz;

#### 【参数】

参数	描述	odeni/	298/1	deri	输入/输出
VeChn	编码通道号,范围:	[0, VENC_M	AX_CHN_NUM)	A LIZ	输入
nFreq	引擎时钟频率,单位	立 MHz。范围:	[480, 532, 600],默认 480,	推荐 532。动态属性。	输入

#### 【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

【需求】

【注意】

【举例】



无必

## 6.3.3.42 AW\_MPI\_VENC\_Set3DNR

【目的】

设置 3D 降噪。

【语法】

ERRORTYPE AW\_MPI\_VENC\_Set3DNR(VENC\_CHN VeChn, int b3DNRFlag);

#### 【参数】

				•
参数。	描述		RIV	输入/输出
VeChn	编码通道号,范围:[0,	VENC_MAX_CHN_NUM)。	NA N	输入
b3DNRFlag	3d 降噪等级。范围: [0	, 6]。推荐值3,数值越大降噪效果越强。	动态属性。	输入

【返回值】

 返回値
 描述

 0
 成功

 非 0
 失败,其值见错误码。

【需求】

【注意】

<del>.</del>...

【举例】

无。

## ${\bf 6.3.3.43 \quad AW\_MPI\_VENC\_Get3DNR}$

【目的】

获取 3D 降噪等级。

【语法】

ERRORTYPE AW\_MPI\_VENC\_Get3DNR(VENC\_CHN VeChn, int \*b3DNRFlag);





#### 【参数】

参数	描述	H. H		输入/输出
VeChn	编码通道号,范围:[	0, VENC_MAX_CHN_NUM)。		输入
b3DNRFlag	3d 降噪等级。范围:	[0,6]。推荐值3,数值越大降噪效	效果越强。动态属性。	输出

#### 【返回值】

返回值	描述	
0	 成功	•
非0	、 <b>失败,其值见错误码</b> 。	ins, in
MA TO THE OWNER OF THE OWNER OWNER OF THE OWNER O	成功失败,其值见错误码。	Allinderii Renterii R
1	I'ha.	
NC_Ge	tCacheState	

【需求】

【注意】

无。

【举例】

无。

## 6.3.3.44 AW\_MPI\_VENC\_GetCacheState

【目的】

获取视频编码库缓冲状态。

【语法】

ERRORTYPE AW\_MPI\_VENC\_GetCacheState(VENC\_CHN VeChn, CacheState \*pCacheState);

#### 【参数】

参数	描述	输入/输出
VeChn	编码通道号,范围: [0, VENC_MAX_CHN_NUM)。	输入
pCacheState	视频编码库缓冲状态。动态属性。	输出

【返回值】

	A11/		A11/
XX	返回值	描述	A TANKS
	0	成功	A STATE OF THE STA
	非 0	失败,	其值见错误码。
			/, \·

【需求】

ALLWIMER

【注意】

无。

【举例】

无。

AW\_MPI\_VENC\_SetRefParam

【目的】

设置编码高级跳帧参考。

【语法】

Ch.  $ERRORTYPE\ AW\ MPI\_VENC\_SetRefParam(VENC\_CHN\ VeChn,\ const\ VENC\_PARAM\_REF\_S$ \*pstRefParam);

#### 【参数】

, o.l.,			
参数	描述	a dina	输入/输出
VeChn	编码通道号,范围:	[0, VENC_MAX_CHN_NUM)。	输入
pstRefParam	高级跳帧参考参数。	静态属性。	输入

【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

【需求】

【注意】



【举例】

泛无。

## 6.3.3.46 AW\_MPI\_VENC\_GetRefParam

【目的】

获取编码高级跳帧参考参数。

【语法】

ERRORTYPE AW\_MPI\_VENC\_GetRefParam(VENC\_CHN VeChn, VENC\_PARAM\_REF\_S \*pstRefParam);

## 【参数】

参数	描述		输入/输出
VeChn	编码通道号,范围	: [0, WENC_MAX_CHI	N_NUM)。 输入
pstRefPara	m 高级跳帧参考参数	。静态属性。	输入

【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,其值见错误码。

【需求】

【注意】

无。

【举例】

无。

#### 6.3.3.47 AW\_MPI\_VENC\_SetHorizonFlip

【目的】

设置编码水平镜像。



#### 【语法】

ERRORTYPE AW\_MPI\_VENC\_SetHorizonFlip(VENC\_CHN VeChn, BOOL bHorizonFlipFlag);

#### 【参数】

参数	描述	输入/输出
VeChn	编码通道号,范围: [0, VENC_MAX_CHN_NUM)。	输入
$b \\ Horizon \\ Flip \\ Flag$	是否水平镜像。动态属性。	输入

#### 【返回值】

【需求】

【注意】

无。

【举例】

无。

## 6.3.3.48 AW\_MPI\_VENC\_GetHorizonFlip

#### 【目的】

获取编码水平镜像。

#### 【语法】

ERRORTYPE AW\_MPI\_VENC\_GetHorizonFlip(VENC\_CHN VeChn, BOOL\*bpHorizonFlipFlag);

	描述	Bei'l	输入/输出
VeChn	编码通道号,范围: [0, VENC_MA	X_CHN_NUM)。	输入
bHorizonFlipFlag	是否水平镜像。动态属性。	\$ . ·	输出



^ <u>_</u> ^		4-6
HELD WHILE	返回值	描述
ZY.	0 非 0	成功 失败,其值见错误码。

【需求】

【注意】

无。

【举例】

AW\_MPI\_VENC\_SetAdaptiveIntraInP 6.3.3.49

【目的】

设置自适应调整 P 帧帧内预测等级属性。

【语法】

Intr-ERRORTYPE AW MPI VENC SetAdaptiveIntraInP(VENC CHN VeChn, BOOL bAdaptiveIntraInPFlag);

【参数】

参数	描述	A TOP TO THE PERSON OF THE PER		输入/输出
VeChn	编码通道号,范围	: [0, VENC_MAX_CHN	I_NUM)。	输入
bAdaptiveIntraInI	PFlag 是否打开自适应调	整 P 帧帧内预测等级属性	功能。静态属性。	输入

【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

【需求】

【注意】



无必

【举例】

无。

#### 6.3.3.50 AW\_MPI\_VENC\_EnableNullSkip

【目的】

打开插空帧功能,在编码帧率不够的情况下,插入空帧达到预期帧率。

【语法】

ERRORTYPE AW\_MPI\_VENC\_EnableNullSkip(VENC\_CHN VeChn, BOOL bEnable);

#### 【参数】

参数	描述		输入/输出
VeChn	编码通道号,	范围:[0, VENC_MAX_CI	HN_NUM)。 输入
bEnable	打开插空帧功	能。静态属性。	输入

【返回值】

返回值 描述	
0 成功	
非 0 失败,	其值见错误码。

【需求】

【注意】

无。

【举例】

无。

#### 6.3.3.51 AW\_MPI\_VENC\_EnablePSkip

【目的》

打开插 skip-P 帧功能,在编码帧率不够的情况下,插入 skip-P 帧达到预期帧率。



#### 【语法】

ERRORTYPE AW\_MPI\_VENC\_EnablePSkip(VENC\_CHN VeChn, BOOL bEnable);

#### 【参数】

参数	描述	输入/输出
VeChn	编码通道号,范围: [0, VENC_MAX_CHN_NUM)。	输入
bEnable	打开插 skip-P 帧功能。静态属性。	输入

#### 【返回值】

返回值。描述
成功
非 0 失败,其值见错误码。

【需求】

【注意】

无。

【举例】

无。

# 6.3.3.52 AW\_MPI\_VENC\_SaveBsFile

【目的】

设置编码库保存码流。

【语法】

ERRORTYPE AW\_MPI\_VENC\_SaveBsFile(VENC\_CHN VeChn, VencSaveBsFile \*pSaveParam);

参数	描述	输入/输出	
VeChn	编码通道号,范围: [0, VENC_MAX_CHN_NUM)。	 输入	
pSaveParam	设置参数决定保存码流的细节。动态属性。	输入	1



Transfer of the second	返回值	描述
	0 非 0	成功失败,其值见错误码。
The state of the s	0	成功

【需求】

【注意】

无。

【举例】

AW\_MPI\_VENC\_SetProcSet 6.3.3.53

【目的】

设置编码库 proc 信息的配置。

【语法】

VEN-ERRORTYPE AW MPI VENC SetProcSet(VENC CHN VeChn, VeProcSet \*pVeProc-Set);

参数	描述		输入/输出
VeChn	编码通道号,范围:[	[0, VENC_MAX_CHN_NUM)。	输入
pVeProc	Set proc 信息配置属性。	动态属性。	输入

【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

【需求】 (注意]

无必

【举例】

无。

## 6.4 数据结构说明

## 6.4.1 H264E\_NALU\_TYPE\_E

【说明】

定义 H264 码流 NALU 类型。

【定义】

typedef enum H264E\_NALU\_TYPE\_E

{

H264E NALU PSLICE = 1, /PSLICE types/

H264E NALU ISLICE = 5, /ISLICE types/

H264E\_NALU\_SEI = 6, /SEI types/

H264E\_NALU\_SPS = 7, |SPS types|

H264E\_NALU\_PPS = 8, /PPS types/

H264E NALU IPSLICE = 9

H264E\_NALU\_BUTT

} H264E NALU TYPE E;

【成员】

描述
———————— 码流为 P 帧类型。
码流为 I 帧类型。
暂未使用。
暂未使用。
暂未使用。
暂未使用。

in.

WE THE VEILING

茶店厂

The Inden



#### 【注意事项】

【相关数据类型及接口】

## 6.4.2 JPEGE\_PACK\_TYPE\_E

#### 【说明】

定义 JPEG 码流的 PACK 类型。

#### 【定义】

typedef enum JPEGE PACK TYPE E

JPEGE\_PACK\_ECS = 5, /ECS types/

JPEGE\_PACK\_APP  $\neq$  6, /APP types/

JPEGE PACK VDO = 7, /VDO types/

JPEGE PACK PIC = 8, /PIC types/

JPEGE\_PACK\_BUTT

} JPEGE PACK TYPE E;

#### 【成员】

EGE_PACK_TYPE_E	We fix Hallinderi
EGE_PACK_TYPE_E	
CS = 5, $/ECS$ types/	and the state of t
$P \neq 6$ ,  APP types	B. L.
OO = 7, /VDO types/	
C = 8, /PIC types/	1111
TTT TYPE_E;	
	in.
成员名称	描述 未使用 未使用
JPEGE_PACK_ECS	未使用
JPEGE_PACK_APP	未使用
JPEGE_PACK_VDO	未使用 未使用 码流为完整的 IPEC 格式图片
JPEGE_PACK_PIC	码流为完整的 JPEG 格式图片。

#### 【注意事项】

【相关数据类型及接口】

## 6.4.3 H265E\_NALU\_TYPE\_E

【说明】

定义 H265 码流 NALU 类型



#### 【定义】

typedef enum H265E NALU\_TYPE\_E

-

H265E NALU PSLICE = 1, /P SLICE types/

H265E\_NALU\_ISLICE = 19, /I SLICE types/

H265E\_NALU\_VPS = 32, /VPS types/

H265E NALU SPS = 33, /SPS types/

H265E\_NALU\_PPS = 34, /PPS types/

H265E\_NALU\_SEI = 39, /SEI types/

H265E\_NALU\_BUTT

} H265E\_NALU\_TYPE\_E;

【成员】

: 34,  PPS types  39,  SEI types  E_E;	A Company of the light of the l	Carthelight Hall Religion of the Control of the Con
成员名称	描述	K\$ C
H265E_NALU_PSLICE	—————————————————————————————————————	
H265E_NALU_ISLICE	码流为I帧类型。	
H265E_NALU_VPS	暂未使用。	
H265E_NALU_SPS	暂未使用。	
H265E_NALU_PPS	暂未使用。	À
H265E_NALU_SEI	暂未使用 (1)	- Blindeni

【注意事项】

【相关数据类型及接口】

## 6.4.4 VENC\_DATA\_TYPE\_U

【说明】

编码输出的码流类型。

【定义】

typedef union VENC\_DATA\_TYPE\_U



H264E\_NALU\_TYPE\_E enH264EType; /H264E NALU types/

JPEGE\_PACK\_TYPE\_E enJPEGEType; /JPEGE pack types/

MPEG4E\_PACK\_TYPE\_E enMPEG4EType; /MPEG4E pack types/

H265E\_NALU\_TYPE\_E enH265EType; /H264E NALU types/

}VENC\_DATA\_TYPE\_U;

#### 【成员】

 成员名称	描述	-
enH264EType	H264 码流包类型	- ni
enH264EType enJPEGEType	JPEG 码流包类型。	Alinde.
enMPEG4EType	MPEG4 码流包类型。不支持。	WILL TO THE PARTY OF THE PARTY
enH265EType	H265 码流包类型	
		· O Hall Her Hall Hall I linderi
	MIN	
PACK_S	r.	
<b>属性结构</b>		178

【注意事项】

【相关数据类型及接口】

## 6.4.5 VENC PACK S

【说明】

定义视频编码包输出属性结构。

【定义】

typedef struct VENC\_PACK\_S

{

//unsigned int mPhyAddr; /the physics address of stream/
unsigned char mpAddr0; /the virtual address of stream\*/
unsigned char \*mpAddr1;
unsigned int mLen0; /the length of stream/

unsigned int mLen1;

uint64\_t mPTS; /PTS/

BOOL mbFrameEnd; /frame end? /

版权所有 © 珠海全志科技股份有限公司。保留一切权利

199



VENC\_DATA\_TYPE\_U mDataType; /the type of stream/

unsigned int mOffset;

unsigned int mDataNum;

VENC\_PACK\_INFO\_S mPackInfo[8];

}VENC\_PACK\_S;

# 【成员】

成员名称	描述
mpAddr0	编码数据 buf0 部份地址(虚拟地址)
mpAddr1	编码数据 buf1 部份地址(虚拟地址)
mLen0	编码数据 buf0 部份长度
mLen1	编码数据 buf1 部份长度
mPTS mPTS	编码数据时间戳,单位微秒。
mbFrameEnd	帧结束标识。范围:[TRUE],一定为 TRUE。
mDataType	数据流类型
mOffset	码流包有效数据和码流包首地址的偏移。一般为 0。
mDataNum	未使用。
mPackInfo[8]	未使用。

# 【注意事项】

buf1 部份不一定有

【相关数据类型及接口】

# 6.4.6 VENC STREAM S

# 【说明】

编码通道编码输出流结构。

# 【定义】

 $typedef\ struct\ VENC\_STREAM\_S$ 

{

VENC PACK S mpPack; /stream pack attribute\*/

unsigned int mPackCount; /the pack number of one frame stream/

版权所有 © 珠海全志科技股份有限公司。保留一切权利



```
unsigned int mSeq; /the list number of stream/
```

union

{

VENC STREAM INFO H264 S mH264Info; /the stream info of h264/

VENC\_STREAM\_INFO\_JPEG\_S mJpegInfo; /the stream info of jpeg/

VENC STREAM INFO MPEG4 S mMpeg4Info; /the stream info of mpeg4/

VENC STREAM INFO H265 S mH265Info; /the stream info of h265/

**}**;

**}VENC STREAM S**;

# 【成员】

万	<b></b>	描述。
n	npPack	数据包数组,当前只填充第一个元素。
n	nPackCount	数据包个数,取值范围:[1],当前只支持一次一个码流包,一个码流包就是一帧的模式。
n	nSeq	编码库内部装载该帧的 buffer 的 id 号。取值范围:[0,255]。
n	nH264Info	H264 编码包信息,未使用。
n	nJpegInfo	Jpeg 编码包信息,未使用。
n	nMpea4Info	Mpeg4 编码包信息,未使用。

【注意事项】

mH265Info

【相关数据类型及接口】

# 6.4.7 VENC\_ATTR\_H264\_S

H265 编码包信息,

【说明】

编码通道 H264 编码属性结构。

【定义】

 $typedef\ struct\ VENC\_ATTR\_H264\_S$ 

[

unsigned int MaxPicWidth;



unsigned int MaxPicHeight

unsigned int BufSize;

unsigned int Profile;

BOOL bByFrame;

unsigned int PicWidth;

unsigned int PicHeight;

unsigned int BFrameNum;

unsigned int RefNum;

#BOOL mbLongTermRef;

BOOL FastEncFlag;

int IQpOffset;

//int mVirtualIFrameInterval;

BOOL mbPIntraEnable;

unsigned int mThreshSize;

描述

**}VENC ATTR H264 S**;

# 【成员】

成员名称

BFrameNum

	砂块口小	THE AND
1	MaxPicWidth	编码通道最大输出图片宽度,范围 [192,4096]。静态属性。
	MaxPicHeight	编码通道最大输出图片高度,范围 [96,4096]。静态属性。
	BufSize	编码器输出 buffer 的长度,范围 [0,1610241024],0 表示系统自行决
		定输出 buffer 的长度。静态属性。
	Profile	profile 类型。0: baseline; 1:MP; 2:HP; 静态属性。
	bByFrame	输入流模式: frame or filed 静态属性。
	PicWidth	编码通道图片编码输出宽度。在满足 MaxPicWidth 范围的前提下,在当
		前输入图片宽度(VENC_ATTR_S::SrcPicWidth)的 [1/4, 4] 倍之间
		任意取值 (必须是偶数)。静态属性。
	PicHeight	编码通道图片编码输出高度。在满足 MaxPicWidth 范围的前提下,在当
	ing.	前输入图片高度(VENC ATTR S::SrcPicHeight)的 [1/4, 4] 倍之间

A Report of the state of the st

版权所有 © 珠海全志科技股份有限公司。保留一切权利

B 帧的数量(0表示不支持 B 帧)。范围[0],不支持 B 帧编码

任意取值 (必须是偶数)。静态属性。



	A 11/	A11/	A11/
成员名称	描述		
RefNum	参考帧数(defaul: 0)。	。范围 [0],不支持配置参考	<b>ś帧数。静态属性</b> 。
FastEncFlag	开启快速编码。开启后编	<b>码器会通过降低编码质量来</b>	提高编码速度。静态
**	属性。	***	4× iv
IQpOffset	单独降低 I 帧编码质量,	从而减少 I 帧的体积,缩小	I 帧和 P 帧的体积差
	距,便于网络平滑传输。	范围 [0,10)。建议设置为(	O,如果超过 6 则 I 帧
	质量大幅下降。静态属性	Ė,	
mbPIntraEnable	开启 P 帧帧内预测,提高	高编码质量。静态属性。	
mThreshSize	编码后一帧大小的上限值	i。0:表示取系统默认值。	静态属性。

【相关数据类型及接口】

# 6.4.8 VENC\_ATTR\_H265\_S

# 【说明】

编码通道 H265 编码属性结构。

#### 【定义】

typedef struct VENC\_ATTR\_H265\_S

ĺ

unsigned int mMaxPicWidth;

unsigned int mMaxPicHeight;

unsigned int mBufSize;

unsigned int mProfile;

BOOL mbByFrame;

unsigned int mPicWidth;

unsigned int mPicHeight;

unsigned int mBFrameNum;

unsigned int mRefNum;

//BOOL mbLongTermRef; //enable long term reference IDR Frame

BOOL mFastEncFlag; //for fast video encoder

S ARTHUR HER BERN AND THE STREET OF THE STRE

That like Indeed

版权所有 © 珠海全志科技股份有限公司。保留一切权利



int IQpOffset; //IQp offset value to offset I frame Qp to decrease I frame size

//int mVirtualIFrameInterval; /\* (5, IDR\_frame\_interval), dynamic param \*/

BOOL mbPIntraEnable; //enalbe p frame intra

unsigned int mThreshSize;

}VENC\_ATTR\_H265\_S;

# 【成员】

成员名称	描述
mMaxPicWidth	编码通道最大输出图片宽度,范围 [192,4096]。静态属性。
mMaxPicHeight	编码通道最大输出图片高度,范围 [96,4096]。静态属性。
mBufSize	编码器输出 buffer 的长度,范围 [0,1610241024],0 表示系统自行决
	定输出 buffer 的长度。静态属性。
mProfile	profile 类型。0: baseline; 1:MP; 静态属性。
mbByFrame	输入流模式:frame or filed。范围 TRUE。只支持 frame 模式。静态
K.	属性。
mPicWidth	编码通道图片编码输出宽度。在满足 MaxPicWidth 范围的前提下,在当
	前输入图片宽度(VENC_ATTR_S::SrcPicWidth)的 [1/4, 4] 倍之间
	任意取值 (必须是偶数)。静态属性。
mPicHeight	编码通道图片编码输出高度。在满足 MaxPicWidth 范围的前提下,在当
	前输入图片高度(VENC_ATTR_S::SrcPicHeight)的 [1/4, 4] 倍之间
	任意取值 (必须是偶数)。静态属性。
mBFrameNum	B 帧的数量。范围:[0]。不支持编码 B 帧。静态属性。
mRefNum	参考帧数。范围:[0]。不支持配置参考帧数。静态属性。
mFastEncFlag	开启快速编码。开启后编码器会通过降低编码质量来提高编码速度。静态
THIRD.	属性。
IQpOffset	单独降低 I 帧编码质量,从而减少 I 帧的体积,缩小 I 帧和 P 帧的体积差
Ÿ	距,便于网络平滑传输。范围 [0,10)。建议设置为 0,如果超过 6 则 I 帧
	质量大幅下降。静态属性。
	开启 P 帧帧内预测,提高编码质量。静态属性。
mThreshSize	编码后一帧大小的上限值。0:表示取系统默认值。静态属性。

# 【注意事项】

【相关数据类型及接口】

# 6.4.9 VENC\_ATTR\_MJPEG\_S

说明】



编码通道 MJPEG 编码属性结构。

# 【定义】

typedef struct VENC\_ATTR\_MJPEG\_S

unsigned int mMaxPicWidth;

unsigned int mMaxPicHeight;

unsigned int mBufSize;

# 【成员】

BOOL mbByFran	ne;
unsigned int mPi	icWidth;
unsigned int mPi	icHeight;
}VENC_ATTR_M	JPEG_S;
【成员】	
	描述
MaxPicWidth	
MaxPicHeight	编码通道最大输出图片高度,范围 [96, 4096]。静态属性。
BufSize	编码器输出 buffer 的长度,范围 [0,1610241024],0 表示系统自行决定
	输出 buffer 的长度。静态属性。
bByFrame	输入流模式:frame or filed。静态属性。
PicWidth	编码通道图片编码输出宽度。在满足 MaxPicWidth 范围的前提下,在当前
ARIVE TO STATE OF THE PROPERTY	输入图片宽度(VENC_ATTR_S::SrcPicWidth)的 [1/4, 4] 倍之间任意
A TOP OF THE PROPERTY OF THE P	取值 (必须是偶数)。静态属性。
PicHeight	编码通道图片编码输出高度。在满足 MaxPicWidth 范围的前提下,在当前
	输入图片高度(VENC_ATTR_S::SrcPicHeight)的 [1/4, 4] 倍之间任意
Š	取值 (必须是偶数)。静态属性。

# 【注意事项】

【相关数据类型及接口】

# 6.4.10 VENC\_ATTR\_JPEG\_S

【说明】

编码通道 JPEG 编码属性结构。



# 【定义】

typedef struct VENC\_ATTR\_JPEG\_S

unsigned int MaxPicWidth; /maximum width of a picture to be encoded, in pixel/
unsigned int MaxPicHeight; /maximum height of a picture to be encoded, in pixel/
unsigned int BufSize; /stream buffer size, 0:adaptive to bitRate /
BOOL bByFrame; /get stream mode is field mode or frame mode/

unsigned int PicWidth; /width of a picture to be encoded, in pixel/ unsigned int PicHeight; /height of a picture to be encoded, in pixel/

BOOL bSupportDCF, support dcf/

}VENC\_ATTR\_JPEG\_S;

# 【成员】

成员名称	描述
MaxPicWidth	编码通道最大输出图片宽度,范围 [192, 4096]。静态属性。
MaxPicHeight	编码通道最大输出图片高度,范围 [96,4096]。静态属性。
BufSize	编码器输出 buffer 的长度,范围 [0,1610241024],0 表示系统自行决定
indeni	输出 buffer 的长度。静态属性。
bByFrame	输入流模式:frame or filed。静态属性。
PicWidth	编码通道图片编码输出宽度。在满足 MaxPicWidth 范围的前提下,在当前
XXX	输入图片宽度(VENC_ATTR_S::SrcPicWidth)的 [1/4, 4] 倍之间任意
	取值 (必须是偶数)。动态属性。
PicHeight	编码通道图片编码输出高度。在满足 MaxPicWidth 范围的前提下,在当前
5	输入图片高度(VENC_ATTR_S::SrcPicHeight)的 [1/4, 4] 倍之间任意
	取值 (必须是偶数)。动态属性。
bSupportDCF	是否支持 DCF,范围:FALSE。不支持 DCF。静态属性。

# 【注意事项】

【相关数据类型及接口】



# 6.4.11 VENC\_ATTR\_S

```
【说明】
```

```
编码通道编码属性结构。
```

#### 【定义】

```
typedef struct VENC_ATTR_S
{

PAYLOAD, TYPE E Type, 4th a f
```

PAYLOAD\_TYPE\_E Type; /the type of payload/

union

VENC\_ATTR\_H264\_S AttrH264e; /attributes of h264/

VENC\_ATTR\_MJPEG\_S AttrMjpeg; /attributes of mjpeg/

 ${\tt VENC\_ATTR\_JPEG\_S~AttrJpeg;~/} \textit{attributes of jpeg/}$ 

VENC ATTR H265 S AttrH265e; /attributes of h265/

**}**;

int MaxKeyInterval; /\* wanted key frame interval\*/

unsigned int SrcPicWidth; /\* source width of a picture sent to venc channel, in pixel\*/
unsigned int SrcPicHeight; /\* source height of a picture sent to venc channel, in pixel\*/

VIDEO\_FIELD\_E Field;

PIXEL FORMAT E PixelFormat;

enum v412 colorspace mColorSpace;

ROTATE E Rotate; /encoder rotate angle./

}VENC\_ATTR\_S;

# 【成员】

成员名称	描述	
Type	编码类型,支持 PT_H264,PT_H265,PT_JPEG,	PT_MJPEG
AttrH264e	H264 编码属性结构	
AttrMjpeg	Mjpeg 编码属性结构	X TO SECOND
AttrJpeg	Jpeg 编码属性结构	NA THE REAL PROPERTY OF THE PARTY OF THE PAR
		W. Ch.



V1/\		
成员名称	描述。	A A A A A A A A A A A A A A A A A A A
AttrH265e	H265 编码属性结构	
MaxKeyInterval	关键帧间隔,范围 [0,+∞)。静态属性。	The state of the s
SrcPicWidth	输入图片宽度,范围[192,4096]。静态属性	· ·
SrcPicHeight	输入图片高度,范围 [96,4096]。静态属性。	
Field	filed or frame 模式。静态属性。	
PixelFormat	像素格式。静态属性。	
mColorSpace	色域空间。静态属性。	
Rotate	旋转角度。静态属性。	

【相关数据类型及接口】

# S Republication of the second 6.4.12 VENC\_ATTR\_H264\_CBR\_S

# 【说明】

定义 H.264 编码通道 CBR 属性结构。

#### 【定义】

 $typedef\ struct\ VENC\_ATTR\_H264\_CBR\_S$ 

{

unsigned int mGop;

unsigned int mStatTime;

unsigned int mSrcFrmRate;

unsigned int fr32DstFrmRate;

unsigned int mBitRate;

unsigned int mFluctuateLevel;

unsigned int mMaxQp; /\* the max qp \*/

unsigned int mMinQp;

} VENC\_ATTR\_H264 CBR S;

版权所有 © 珠海全志科技股份有限公司。保留一切权利



成员名称 描述 mGop 暂不支持。范围:[0]。 mStatTime 统计时间的速率。暂不支持。范围: [0]。 mSrcFrmRate 编码通道输入帧率。已废弃。使用 VENC FRAME RATE S 替代。范围:[0] fr32DstFrmRate 编码通道输出帧率。已废弃。使用 VENC FRAME RATE S 替代。范围: [0] mBitRate 平均码率,单位 bps,范围: (0,+∞)。 最大码率相对平均码率的波动等级。暂不支持。取值范围:[0]。 mFluctuateLevel mMaxQp 最大 Qp 值

# 【注意事项】

mMinQp

【相关数据类型及接口】

# Real of the state 6.4.13 VENC\_ATTR\_H264\_VBR\_S

最小 Qp 值

# 【说明】

定义 H.264 编码通道 VBR 属性结构。

#### 【定义】

typedef struct VENC\_ATTR\_H264\_VBR\_S

unsigned int mGop;

unsigned int mStatTime;

unsigned int mSrcFrmRate;

unsigned int fr32DstFrmRate;

unsigned int mMaxBitRate;

unsigned int mMaxQp;

unsigned int mMinQp;

unsigned int mMovingTh;

int mQuality;

**}VENC ATTR H264 VBR** 

【成员】



	成员名称	描述	A TOP TO THE PERSON OF THE PER	
NA CO	mGop	暂不支持。范围: [0]。		A STATE OF THE STA
,	mStatTime	统计时间的速率。暂不支	持。范围: [0]。	The state of the s
	mSrcFrmRate	。 编码通道输入帧率。已废	奔。使用 VENC_FRAM	IE_RATE_S 替代。范
		围: [0]		
	fr 32 Dst Frm Rate	编码通道输出帧率。已废	弃。使用 VENC_FRAM	IE_RATE_S 替代。范
		围:[0]		
	mMaxBitRate	编码器输出最大码率,单	位 bps,范围:(0,+∞)。	
	mMaxQp	编码器输出图像最大 Qp	值。取值范围:(mMinC	(p, 51],建议 [30, 45]
	mMinQp	编码器输出图像最小 Qp	值。取值范围:[0,51],	建议 [10,20]。
	mMovingTh	,	mes are moving, 31:ha	ave no moving frame,
	::rden'	default: 20, 0 means	use default value.	8000
	mQuality	. 117	quality, 10:best quality	y, default:5, 0 means
	A STATE OF THE STA	use default value		
	XXT			
*	【注意事项】			
	【/工总事项】	331	W. T.	N. T. S.
	【相关数据类型及接口	□]		
			1 MY	
	6.4.14 VENC	CATTR_H264_F	IXQP_S	
	【说明】			
	ウツロの仏神神学学	t Risson RM4th		
	定义 H.264 编码通道	B FIXUD 禹性结构。	. Heri	-8
	【定义私 <sup>jing</sup>	Bliffe	Allina	Alline

【定义】

typedef struct VENC ATTR H264 FIXQP S

unsigned int mGop;

unsigned int mSrcFrmRate;

unsigned int fr32DstFrmRate;

unsigned int mIQp;

unsigned int mPQp;

} VENC\_ATTR\_H264\_FIXQP\_S;

【成员】

版权所有 © 珠海全志科技股份有限公司。保留一切权利



	成员名称	描述		-XX		× .	
Ú,	mGop	暂不支持。范围:[0	0]。		A STATE OF THE STA		
	mSrcFrmRate	编码通道输入帧率。	已废弃。使用	VENC_FRAME	_RATE_S 替代。	范围:	[0]
	fr32DstFrmRate	编码通道输出帧率。	已废弃。使用	VENC_FRAME	_RATE_S 替代。	范围:	[0]
	mIQp	I 帧 Qp 值。取值范	围: (0,51]				
	mPQp	P 帧 Qp 值。取值范	5国: [0,51]				

【相关数据类型及接口】

# に定义】 typedef struct VENC\_ATTR\_MJPEG\_CBR\_S { unsigned int mSrcFrmRate: nsigned int psigned int psig

unsigned int fr32DstFrmRate;

unsigned int mBitRate;

unsigned int mFluctuateLevel;

} VENC\_ATTR\_MJPEG\_CBR\_S;

# 【成员】

成员名称	描述
mStatTime	统计时间的速率。暂不支持。范围:[0]。
mSrcFrmRate	编码通道输入帧率。已废弃。使用 VENC_FRAME_RATE_S 替代。范围: [0]
fr32DstFrmRate	编码通道输出帧率。已废弃。使用 VENC_FRAME_RATE_S 替代。范围: [0]
mBitRate	平均码率,单位 bps,范围: (0,+∞)。
mFluctuateLevel	最大码率相对平均码率的波动等级。暂不支持。取值范围: [0]。



【相关数据类型及接口】

# 6.4.16 VENC\_ATTR\_MJPEG\_FIXQP S

【说明】

定义 MJPEG 编码通道 FixQp 属性结构。

【定义】

typedef struct VENC\_ATTR\_MJPEG\_FIXQP\_S

unsigned int mSrcFrmRate;

unsigned int fr32DstFrmRate;

unsigned int mQfactor;

**}VENC ATTR MJPEG FIXQP S;** 

【成员】

成员名称 描述

mSrcFrmRate

编码通道输入帧率。已废弃。使用 VENC\_FRAME\_RATE\_S 替代。范围: [0]

fr32DstFrmRate

编码通道输出帧率。已废弃。使用 VENC FRAME RATE S 替代。范围:[0]

mQfactor

JPEG 图像编码质量。取值范围:[0,100],建议取值 [50,100],值越大,图像质量越高

# 【注意事项】

【相关数据类型及接口】

# 6.4.17 VENC\_ATTR\_H265\_CBR\_S

【说明】

定义 H.265 编码通道 CBR 属性结构。

【定义】

typedef struct VENC ATTR H265 CBR S



unsigned int mGop;

unsigned int mStatTime;

unsigned int mSrcFrmRate;

unsigned int fr32DstFrmRate;

unsigned int mBitRate;

unsigned int mFluctuateLevel;

unsigned int mMaxQp; /\* the max qp \*/

unsigned int mMinQp;

WENC ATTR H264 CBR S

# 【成员】

<b>成页名</b> 称	描述	100		12
mGop	等 暂不支持。范围:	[0]。	( )	(C)
mStatTime	统计时间的速率。	暂不支持。范围:[0]。		

mSrcFrmRate 编码通道输入帧率。已废弃。使用 VENC\_FRAME\_RATE\_S 替代。范围: [0] fr32DstFrmRate 编码通道输出帧率。已废弃。使用 VENC FRAME RATE S 替代。范围: [0]

mBitRate 平均码率,单位 bps,范围: (0,+∞)。

mFluctuateLevel 最大码率相对平均码率的波动等级。暂不支持。取值范围: [0]。

mMaxQp 最大 Qp 值 mMinQp 最小 Qp 值

# 【注意事项】

【相关数据类型及接口】

# 6.4.18 VENC\_ATTR\_H265\_VBR\_S

#### 【说明】

定义 H.265 编码通道 VBR 属性结构。

#### 【定义】

 $typedef\ struct\ VENC\_ATTR\_H265\_VBR\_S$ 

unsigned int mGop;

版权所有 © 珠海全志科技股份有限公司。保留一切权利



unsigned int mStatTime;

unsigned int mSrcFrmRate;

unsigned int fr32DstFrmRate;

unsigned int mMaxBitRate;

unsigned int mMaxQp;

unsigned int mMinQp;

unsigned int mRatioChangeQp;

int mQuality;

}VENC\_ATTR\_H265\_VBR\_S;

# 【成员】

成员名称	描述
mGop	暂不支持。范围: [0]。
mStatTime	统计时间的速率。暂不支持。范围:[0]。
mSrcFrmRate	编码通道输入帧率。已废弃。使用 VENC_FRAME_RATE_S 替代。范围: [0]
fr 32 Dst Frm Rate	编码通道输出帧率。已废弃。使用 VENC_FRAME_RATE_S 替代。范围: [0]
mMaxBitRate	编码器输出最大码率,单位 bps,范围: (0,+∞)。
mMaxQp	编码器输出图像最大 Qp 值。取值范围: (mMinQp, 51],建议 [30, 45]
mMinQp	编码器输出图像最小 Qp 值。取值范围: [0, 51],建议 [10,20]。
mRatio Change Qp	range[50,100], default:85
mQuality	range[1,13], 1:worst quality, 13:bestquality

# 【注意事项】

【相关数据类型及接口】

# 6.4.19 VENC\_ATTR\_H265\_FIXQP\_S

# 【说明】

定义 H.265 编码通道 FixQp 属性结构。

# 【定义】

typedef struct VENC ATTR H265 FIXQP S



```
unsigned int mGop;
```

unsigned int mSrcFrmRate;

unsigned int fr32DstFrmRate;

unsigned int mIQp;

unsigned int mPQp;

} VENC ATTR H264 FIXQP S;

# 【成员】

成员名称	描述	Abril.	deri
mGop	暂不支持。范围: [0]。	ONS.	
mSrcFrmRate	编码通道输入帧率。已废弃。	使用 VENC_FRAME_	RATE_S 替代。范围: [0]
fr32DstFrmRate	编码通道输出帧率。已废弃。	使用 VENC_FRAME_	RATE_S 替代。范围: [0]
mIQp	I 帧 Qp 值。取值范围: (0,	51]	
mPQp	P 帧 Qp 值。取值范围: [0,	51]	
		NE	
【注意事项】		1114	
【相关数据类型及接	::1	N.	
6.4.20 VENO	C ATTR H265 AB	RS	

# 【注意事项】

# 6.4.20 VENC\_ATTR\_H265\_ABR\_S

【说明】

定义 H.265 编码通道 ABR 属性结构。

# 【定义】

typedef struct VENC\_ATTR\_H265\_ABR\_S

{

unsigned int mGop;

unsigned int mStatTime;

unsigned int mSrcFrmRate;

unsigned int fr32DstFrmRate;

unsigned int mAvgBitRate



unsigned int mMaxBitRate;

unsigned int mMinStaticPercent;

unsigned int mMaxIQp;

unsigned int mMinIQp;

}VENC\_ATTR\_H265\_ABR\_S;

# 【成员】

成员名称	描述
mGop	暂不支持。范围; [0]。
mStatTime	统计时间的速率。暂不支持。范围:[0]。
mSrcFrmRate	编码通道输入帧率。已废弃。使用 VENC_FRAME_RATE_S 替代。范围: [0]
fr32DstFrmRate	编码通道输出帧率。已废弃。使用 VENC_FRAME_RATE_S 替代。范围: [0]
mAvgBitRate	编码器输出平均码率。单位 bps,范围: (0,+∞)。
mMaxBitRate	编码器输出最大码率,单位 bps,范围: (0,+∞)。
mMinStaticPercent	静态场景下最小码率和最大码率的比值百分比,范围:[20,50]。推荐40。
mMaxIQp	编码器 I 帧最大 Qp 值。取值范围:(mMinIQp,45]。
mMinIQp	编码器 I 帧最小 Qp 值。取值范围: [25, 32]。
mMaxBitRate mMinStaticPercent mMaxIQp	编码器输出最大码率,单位 $bps$ ,范围: $(0,+\infty)$ 。 静态场景下最小码率和最大码率的比值百分比,范围: $[20,50]$ 。推荐 $40$ 。 编码器 $I$ 帧最大 $Qp$ 值。取值范围: $(mMinIQp,45]$ 。

# 【注意事项】

【相关数据类型及接口】

# 6.4.21 VENC\_RC\_ATTR\_S

# 【说明】

编码通道码率控制属性。

#### 【定义】

```
typedef\ struct\ VENC\_RC\_ATTR\_S
```

{

VENC\_RC\_MODE\_E mRcMode; /the type of rc/

union

{

VENC\_ATTR\_H264\_CBR\_S mAttrH264Cbr;



VENC ATTR H264 VBR SmAttrH264Vbr;

VENC ATTR H264 FIXQP S mAttrH264FixQp;

VENC\_ATTR\_H264\_ABR\_S mAttrH264Abr

VENC ATTR H264 QPMAP S mAttrH264QpMap;

VENC ATTR MPEG4 CBR S mAttrMpeg4Cbr;

VENC ATTR MPEG4 FIXQP S mAttrMpeg4FixQp;

VENC ATTR MPEG4 VBR S mAttrMpeg4Vbr;

VENC\_ATTR\_MJPEG\_CBR\_S mAttrMjpegeCbr;

NE RELEASER BEEFFERENCE FOR THE SECOND SECON VENC ATTR MJPEG FIXQP S mAttrMjpegeFixQp;

VENC\_ATTR\_MJPEG\_VBR\_S mAttrMjpegeVbr;

VENC ATTR H265 CBR S mAttrH265Cbr;

VENC ATTR H265 VBR S mAttrH265Vbr;

VENC ATTR H265 FIXQP S mAttrH265FixQp;

VENC ATTR H265 ABR S mAttrH265Abr;

VENC\_ATTR\_H265\_QPMAP\_S mAttrH265QpMap;

**}**;

void\* pRcAttr ;

**}VENC RC ATTR S**;

【成员】

. 7/5		. 7/2"	\\Z_{\infty}	
操作	成员名称	描述	**************************************	
	mRcMode	RC 模式(码率控制模式)	。静态属性。	
	mAttrH264Cbr	H.264 协议编码通道 Cb	r 模式属性。	
	mAttrH264Vbr	H.264 协议编码通道 Vb:	r 模式属性。	
	mAttrH264FixQp	H.264 协议编码通道 Fix	qp 模式属性。	
	mAttrH264Abr	H.264 协议编码通道 Ab	r 模式属性(暂不支持)	0
	mAttrMjpegeFixQp	Mjpeg 协议编码通道 Fix	αqp 模式属性。	
	mAttrMjpegeCbr	Mjpeg 协议编码通道 cb	r 模式属性。	
	mAttrMjpegeVbr	Mjpeg 协议编码通道 vb	r 模式属性(暂不支持)	-8
	mAttrH265Cbr	H.265 协议编码通道 Cb	r 模式属性。	- Allino
A STATE OF THE STA	mAttrH265Vbr	H.265 协议编码通道 Vb	r 模式属性。	TO SEPLIA
XXX	mAttrH265FixQp	H.265 协议编码通道 Fix	qp 模式属性。	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
		the state of the s		
- KB	· · · · · · · · · · · · · · · · · · ·	反权所有 © 珠海全志科技股份有限公司。保	留一切权利	217
-Ai	. A1.	γι.	·γ.	



成员名称	描述		
mAttrH265Abr	H.265 协议编码通道	道 Abr 模式属性	HE PHOTO TO THE PARTY OF THE PA
mAttrH265QpMap	H.265 协议编码通道	道 QpMap 模式属性	(暂不支持)

【相关数据类型及接口】

VENC GOP ATTR S

# 6.4.22 VENC\_GOP\_MODE\_E

# 【说明】

定义 H264/H265 GOP 类型。

# 【定义】

typedef enum VENC GOP MODE E

 $VENC\_GOPMODE\_NORMALP = 0$ 

 $VENC_GOPMODE_DUALP = 1$ ,

 $VENC_GOPMODE_SMARTP = 2$ ,

VENC\_GOPMODE\_BIPREDB = 3,

VENC\_GOPMODE\_LOWDELAYB = 4,

VENC\_GOPMODE\_BUTT,

**}VENC GOP MODE E;** 

# 【成员】

成员名称	描述
VENC_GOPMODE_NORMALP	编码单参考帧 P 帧 GOP 类型。
VENC_GOPMODE_DUALP	编码双参考帧 P 帧 GOP 类型。
VENC_GOPMODE_SMARTP	编码智能 P 帧 GOP 类型。
VENC_GOPMODE_BIPREDB	不支持。
VENC_GOPMODE_LOWDELAYB	不支持。

Real Property of the State of t

【相关数据类型及接口】

无。

# 6.4.23 VENC\_GOP\_NORMALP\_S

【说明】

定义编码单参考帧 P 帧 GOP 属性结构体。

未使用该数据结构,无意义。

【定义】

Stri / typedef struct VENC\_GOP\_NORMALP\_S

int mIPQpDelta;

}VENC GOP NORMALP S;

【成员】

【注意事项】

【相关数据类型及接口】

无。

# 6.4.24 VENC\_GOP\_DUALP\_S

【说明】

定义编码双参考帧 P 帧 GOP 属性结构体。

未使用该数据结构,无意义。

【定义】

typedef struct VENC\_GOP\_DUALP\_S

unsigned int mSPInterval;



```
int mSPQpDelta;
int mIPQpDelta;
}VENC_GOP_DUALP_S;
```

【成员】

【注意事项】

【相关数据类型及接口】

无。

# Will the state of 6.4.25 VENC\_GOP\_SMARTP\_S

【说明】

定义编码智能 P 帧 GOP 属性结构体。

【定义】

typedef struct VENC\_GOP\_SMARTP\_S

{

unsigned int mBgInterval;

int mBgQpDelta;

int mViQpDelta; /\*\*/

int mVirtualIFrameInterval;

}VENC\_GOP\_SMARTP\_S;

【成员】

成员名称	描述
mBgInterval	未使用
mBgQpDelta	未使用
mViQpDelta	未使用
mVirtual IF rame Interval	虚拟Ⅰ帧间隔。

【注意事项】

【相关数据类型及接口】



# 6.4.26 VENC\_GOP\_ATTR\_S

# 【说明】

编码通道 GOP 属性。

# 【定义】

```
typedef struct VENC GOP ATTR S
```

{

union

VENC\_GOP\_NORMALP\_S stNormalP; /attributes of normal P/
VENC\_GOP\_DUALP\_S stDualP; /attributes of dual P/
VENC\_GOP\_SMARTP\_S stSmartP

VENC GOP BIPREDB S stBipredB; /attributes of b /

**}**;

int mGopSize;

}VENC\_GOP\_ATTR\_S;

# 【成员】

成员名称	描述	***************************************	※1
en Gop Mode	GOP 模式。静	态属性。	
stNormalP	编码单参考帧:	P 帧 GOP 属性结构体。	未使用。
stDualP	编码双参考帧:	P 帧 GOP 属性结构体。	未使用。
stSmartP	编码双参考帧:	P 帧 GOP 属性结构体。	
stBipredB	不支持。		

# 【注意事项】

【相关数据类型及接口】

VENC\_GOP\_ATTR\_S

# 6.4.27 VENC\_CHN\_ATTR\_S

【说明】

编码通道属性结构。

【定义】

typedef struct VENC CHN ATTR S

{

A Republish the state of the st VENC\_ATTR\_S VeAttr; /the attribute of video encoder/

VENC RC ATTR S RcAttr; /the attribute of rate ctrl/

VENC\_GOP\_ATTR\_\$ GopAttr;

**}VENC CHN ATTR S**;

【成员】

成员名称 描述

VeAttr

编码属性

RcAttr

码率控制属性

GopAttr

Gop 属性

【注意事项】

【相关数据类型及接口】

# 6.4.28 VENC\_CHN\_STAT\_S

【说明】

编码通道状态结构。

【定义】

typedef struct VENC CHN STAT S

unsigned int mLeftPics; (left picture number /



unsigned int mLeftStreamBytes; /left stream bytes/

unsigned int mLeftStreamFrames; /left stream frames/

unsigned int mCurPacks; /pack number of current frame/

unsigned int mLeftRecvPics;

/Number of frames to be received. This member is valid after AW\_MPI\_VENC\_StartRecvPicEx is called./

unsigned int mLeftEncPics;

/Number of frames to be encoded. This member is valid after AW\_MPI\_VENC\_StartRecvPicEx is called.

**}VENC CHN STAT S**;

# 【成员】

成员名称	描述	
mLeftPics	剩余图片量,待编码。	
mLeftStreamBytes	编码流剩余 Byte 数,待取走。	NN
mLeftStreamFrames	编码流剩余帧数,待取走。	
mCurPacks	当前帧的码流包个数,取值范围	围: [1],当前只支持一帧一个码流包的模式。
mLeftRecvPics	待接收图片量(仅 AW_MPI_V	VENC_StartRecvPicEx 时有效),未使用
mLeftEncPics	已编码图片量(仅 AW_MPI_V	VENC_StartRecvPicEx 时有效),未使用

# 【注意事项】

【相关数据类型及接口】

# 6.4.29 VENC\_EXIFINFO\_S

# 【说明】

aw EXIF 信息结构,配置缩略图大小,以及供编码器生成 jpeg 图片信息,信息可以随意填写。

#### 【定义】

{

typedef struct VENC\_EXIFINFO\_S //aw

unsigned char CameraMake[MM INFO LENGTH];

unsigned char CameraModel[MM INFO LENGTH];

版权所有 © 珠海全志科技股份有限公司。保留一切权利



unsigned char DateTime[MM\_DATA\_TIME\_LENGTH];

unsigned int ThumbWidth;

unsigned int ThumbHeight;

int Orientation; //value can be 0,90,180,270 degree

unsigned int fr32ExposureTime; //tag 0x829A, FRACTION32()

unsigned int fr32FNumber; //tag 0x829D, FRACTION32()

short ISOSpeed;//tag 0x8827

Real Astallar Control of the Control short MeteringMode; //tag 0x9207, ExifMeteringModeType

unsigned int fr32FocalLength; //tag 0x920A

short WhiteBalance; //tag 0xA403

// gps info

int enableGpsInfo;

double gps latitude;

double gps longitude;

double gps altitude;

long gps timestamp;

unsigned char gpsProcessingMethod[MM GPS PROCESS METHOD LENGTH];

unsigned char CameraSerialNum[128]; //tag 0xA431 (exif 2.3 version)

short FocalLengthIn35mmFilm; // tag 0xA405

unsigned char ImageName[128]; //tag 0x010D

unsigned char ImageDescription[128]; //tag 0x010E

int thumb quality; //[20, 100]

} VENC EXIFINFO S;

【成员】

成员名称	描述	
CameraMake	camera 厂商。动态属性。	
CameraModel	camera 型号动态属性。	
DateTime	日期时间动态属性。	



	~ \l	
操制技术	成员名称	描述
A STATE OF THE STA	ThumbWidth	编码通道生成的缩略图宽度。动态属性。
1/2 - 1/2 -	ThumbHeight	编码通道生成的缩略图高度。动态属性。
T. T	Orientation	方向。动态属性。
	fr32ExposureTime	曝光值。动态属性。
	fr32FNumber	?
	ISOSpeed	?
	MeteringMode	?
	fr32FocalLength	?
	WhiteBalance	?
	enableGpsInfo	?
8	gps_latitude	? Meri
	gps_longitude	?
	gps_altitude	?
The state of the s	gps_timestamp	?
High	gpsProcessingMethod	?
-19h \ -1	CameraSerialNum	?
₹K-1	FocalLengthIn35mmFilm	? ***
	ImageName	?
	ImageDescription	?
	thumb_quality	设置缩略图编码质量

【相关数据类型及接口】

# 6.4.30 VENC\_JPEG THUMB\_BUFFER\_S

```
【说明】
```

# 【定义】

```
typedef\ struct\ VENC\_JPEG\_THUMB\_BUFFER\_S\ /\!/aw
```

{

unsigned char\* ThumbAddrVir;

unsigned int ThumbLen;

} VENC\_JPEG\_THUMB\_BUFFER\_S;

# 【成员】

版权所有 © 珠海全志科技股份有限公司。保留一切权利



成员名称	描述		٠.
ThumbAddrVir	jpeg 图片 ]	buffer 中的 thumbPic	的起始虚拟地址,
ThumbLen	thumbPic	的长度。	

【相关数据类型及接口】

# 6.4.31 VENC\_PARAM\_JPEG\_S

【说明】

# 【定义】

typedef struct VENC\_PARAM\_JPEG\_S

Republish the state of the stat unsigned int Qfactor; /image quality:[1,99]/

unsigned char YQt[64]; /\* y qt value \*/

unsigned char CbQt[64]; /\* cb qt value \*/

unsigned char CrQt[64]; /\* cr qt value \*/

unsigned int MCUPerECS; /default value: 0, MCU number of one ECS/

} VENC PARAM JPEG S;

# 【成员】

成员名称	描述				162	
Qfactor	JPEG 编码质量。	范围:	[0,100],	值越大,	编码质量越高。	动态属性。
YQt	未使用					
CbQt	未使用					
CrQt	未使用					
MCUPerECS	未使用					

#### 【注意事项】

【相关数据类型及接口】



# 6.4.32 VENC\_ROLEFG\_S

# 【说明】

定义编码感兴趣区域信息。

#### 【定义】

typedef struct VENC ROI CFG S

{

unsigned int Index; /\* Index of an ROI. The system supports indexes ranging from 0

BOOL bEnable; /\* Whether to enable this ROI \*/

BOOL bAbsQp; /\* QP mode of an ROI.FALSE: relative QP.TURE: absolute QP.\*/

#### 【成员】

nt Qp; /* QP RECT_S Rect	value.*/ e; /* Region of an ROI*/
VENC_ROI_	CFG_S;
【成员】	
成员名称	描述
Index	索引号。范围: [0,7]。动态属性。
bEnable	使能标记。动态属性。
bAbsQp	是否绝对 Qp 值(FALSE: 相对 Qp 值 TRUE: 绝对 Qp 值)动态属性。
$\mathbf{Q}\mathbf{p}^{V}$	当 qp 模式是相对 Qp 值时,取值范围 [0,51],表示用该帧的 qp 值减去
X	Qp,作为绝对 Qp 值;当 Qp 模式是绝对 qp 值时,取值范围 [0,51]。动态
	属性。
Rect	区域范围,必须在图像范围内, $X,Y,Width,Height$ 必须 $16$ 对齐。动态属
	性影響

# 【注意事项】

【相关数据类型及接口】

# 6.4.33 VENC\_COLOR2GREY\_S

【说明】

彩转灰结构



# 【定义】

typedef struct VENC\_COLOR2GREY\_S

BOOL bColor2Grey; /\* Whether to enable Color2Grey.\*/

**}VENC COLOR2GREY S;** 

【成员】

成员名称

描述

bColor2Grey 开启或关闭一个通道的彩转灰功能。TRUE:开启;FALSE:关闭。动态属性。

【注意事项】

【相关数据类型及接口】

6.4.34 VENC\_CROP\_CFG\_S

【说明】

裁剪区域结构

【定义】

typedef struct VENC\_CROP\_CFG\_S

BOOL bEnable; /\* Crop region enable \*/

RECT\_S Rect; /\* Crop region, note: X must be multi of 16 \*/

**}VENC CROP CFG S;** 

【成员】

成员名称 描述

是否使能 bEnable

裁剪区域 Rect

【注意事项】

【相关数据类型及接口】

版权所有 © 珠海全志科技股份有限公司。保留一切权利



# 6.4.35 VENC\_FRAME\_RATE\_S

```
【说明】
```

帧率设置结构

# 【定义】

```
typedef struct VENC_FRAME_RATE_S
{
```

int SrcFrmRate; /\* Input frame rate of a channel\*/

int DstFrmRate; /\* Output frame rate of a channel\*/

} VENC\_FRAME\_RATE\_S

# 【成员】

成员名称	描述	**************************************			THE STATE OF THE S
SrcFrmRate	进入编码通道的帧率	率,取值范围: [1	,240]。	静态	属性。
DstFrmRate	编码通道输出帧率,	,取值范围: (0,S	rcFrm	Rate	]。静态属性。

# 【注意事项】

【相关数据类型及接口】

# 6.4.36 VENC\_STREAM\_BUF\_INFO\_S

# 【说明】

vbvBuffer 信息的结构体。

# 【定义】

```
typedef\ struct\ VENC\_STREAM\_BUF\_INFO\_S
```

{

unsigned int PhyAddr;

void \*pUserAddr;

unsigned int BufSize;

YVENC\_STREAM\_BUF\_INFO\_S;

版权所有 © 珠海全志科技股份有限公司。保留一切权利



# 【成员】

7. b.	
成员名称  描述	
PhyAddr vbvBuffer 的起始物理地址 pUserAddr vbvBuffer 的起始虚拟地址	
BufSize vbvBuffer 的大小。	.0

# 【注意事项】

【相关数据类型及接口】

# 

unsigned int RefreshLineNum;

unsigned int ReqIQp;

}VENC\_PARAM\_INTRA\_REFRESH\_S;

# 【成员】

成员名称	描述
bRefreshEnable	是否开启 P 帧帧内刷新功能。TRUE:开启;FALSE:关闭。静态属性。
bISliceEnable	未使用。
RefreshLineNum	图像帧按列划分的区域个数。例如分为 $10$ 个区域,则每 $10$ 帧刷新一次。取值范围:( $0$ , +
ReqIQp	未使用。



# 6.4.38 VENC\_PARAM\_REF\_S

【说明】

高级跳帧参考参数。

【定义】

 $typedef\ struct\ VENC\_PARAM\_REF\_S$ 

{

unsigned int Base; /Base layer period/

unsigned int Enhance; /Enhance layer period/

BOOL bEnablePred; /Whether some frames at the base layer are referenced by other frames at the base layer. When bEnablePred is FALSE, all frames at the base layer reference IDR frames.

} VENC\_PARAM\_REF\_S;

【成员】

成员名称 描述

Base 层的周期。范围: [0, +∞)。

Enhance 层的周期。范围: [0, +∞)。

bEnablePred base 层的帧是否被 base 层其他帧用作参考。FALSE 表示 base 层所有帧都参考 IDR 帧。

【注意事项】

【相关数据类型及接口】

# 6.4.39 VencHeaderData

【说明】

spspps 数据信息。

【定义】

typedef struct VencHeaderData {

unsigned char\* pBuffer;

unsigned int nLength;

版权所有 © 珠海全志科技股份有限公司。保留一切权利



# }VencHeaderData;

# 【成员】

成员名称 描述

pBuffer spspps 信息的 buffer 地址, buffer 是编码库内部 buffer。

Report of the state of the stat

nLength buffer 的有效数据长度

# 【注意事项】

【相关数据类型及接口】

# 6.4.40 VencSmartFun

# 【说明】

smart 编码参数。

# 【定义】

typedef struct {

unsigned char smart fun en;

unsigned char img bin en;

unsigned int img\_bin\_th;

unsigned int shift\_bits;

}VencSmartFun;

# 【成员】

成员名称	描述		**
smart_fun_en	Smart 功能开关标志。静态属	性。	
img_bin_en	二值化输出开关标志,当 smai	rt 标志设置为 1 时该标志	忌强制为 1 静态属性。
$img\_bin\_th$	运动区域判别阈值,取值范围:	[],默认值 27。静态属	性。
shift_bits	阈值计算移位位数,取值范围:	[],默认值 2。静态属性	Eo

# 【注意事项】

【相关数据类型及接口】



# 6.4.41 VencBrightnessS

# 【说明】

配置 h264 和 h265 编码的亮暗阈值属性,与 smart 功能配合使用,对于 smart 检索之外的非运动

区域(即背景区域),如果超出过亮或过暗阈值,将会被 Smart 功能处理,默认值设置为 60 / 200,将这两个阈值往平均值调节,将会提高背景区域的压缩效率,但是显示效果可能会变差;

# 【定义】

typedef struct VencBrightnessS {

unsigned int dark\_th; //dark threshold, default 60, range[0, 255]

unsigned int bright\_th; //bright threshold, default 200, range[0, 255]

#### 【成员】

}VencBrightnessS;

成员名称	描述		1 NN	
dark_th	暗阈值,范	围:[0,255]	,默认值 60 静	 态属性。
bright_th	亮阈值,范	围: [0, 255]	, 默认值 200 🏻	静态属性。

#### 【注意事项】

【相关数据类型及接口】

# 6.4.42 CacheState

#### 【说明】

编码库 vbvbuffer 的缓冲状态。

#### 【定义】

 $typedef\ struct\ CacheState\ \{$ 

unsigned int mValidSizePercent; // 0~100

unsigned int mValidSize; // unit:kB

unsigned int mTotalSize; // unit:kB

\*CacheState;

版权所有 © 珠海全志科技股份有限公司。保留一切权利



# 【成员】

72X3	(EX.)
成员名称	描述
mValidSizePercent	有效数据的百分比。
mValidSize	buffer 的有效数据长度。
mTotalSize	buffer 的总长度。

# 【注意事项】

【相关数据类型及接口】

# 6.4.43 VencSaveBSFile

# 【说明】

White Little High the little has been a second as the little h 配置参数详细描述编码库保存码流的细节。

# 【定义】

typedef struct VencSaveBSFile {

char filename[256];

unsigned char save bsfile flag;

unsigned int save\_start\_time;

unsigned int save end time;

}VencSaveBSFile;

# 【成员】

成员名称 💉	描述	**************************************
filename[256]	码流保存路径及名称。	
save_bsfile_flag	是否开启保存码流功能	
save_start_time	距离开始编码的时间间隔	,以该间隔作为保存码流的起始时间
save_end_time	距离开始编码的时间间隔	,以该间隔作为保存码流的结束时间

# 【注意事项】

【相关数据类型及接口】



# 6.4.44 VeProcSet

# 【说明】

编码库的 proc 信息设置。

# 【定义】

typedef struct VeProcSet {

unsigned char bProcEnable;

unsigned int nProcFreq;

# 【成员】

unsigned int nProcFi	req;				
unsigned int nStatis	BitRateTime;		Beri	Mindeni	
unsigned int nStatis	FrRateTime;	A TO THE STATE OF		A Callinger	
}VeProcSet;		A STATE OF THE STA	© think hit		
【成员】	\forall \tag{\sigma}	W. The		~×	A PARTY OF THE PAR
成员名称	描述		6		
bProcEnable	是否开启 proc 调试	<b></b> 切能。			
nProcFreq	开启 proc 调试功能	的每隔多少帧更新一次	<b></b>		
nStatisBitRateTime	码率统计时间间隔	(即以该时间间隔作为 ]	proc 信息中的瞬时码	率统计时间间隔)	,单位
nStatisFrRateTime	帧率统计时间间隔	(即以该时间间隔作为 ]	proc 信息中的瞬时帧	率统计时间间隔)	,单位

# 【注意事项】

【相关数据类型及接口】

# 6.5 错误码

错误码	宏定义		描述
0xA0088002	ERR_VENC_INVALID_CHNID		无效的编码通道号
0xA0088003	ERR_VENC_ILLEGAL_PARAM		编码参数设置无效
0xA0088004	ERR_VENC_EXIST		编码通道已经创建
0xA0088005	ERR_VENC_UNEXIST		编码通道未创建
0xA0088006	ERR_VENC_NULL_RTR	18/1	空指针
0xA0088007	ERR_VENC_NOT_CONFIG	- fillindeni	编码通道未配置
0xA0088008	ERR_VENC_NOT_SUPPORT	RIVE IN	操作不支持
0xA0088009	ERR_VENC_NOT_PERM	AN.	操作不允许
			NATA CARACTER STATE OF THE STAT



Will Little White Head of the late of the

ALLWIMER	allinderi allinderi	文档密级、秘密
错误码	宏定义	描述
0xA008800C	ERR_VENC_NOMEM	系统内存不足
0xA008800D	ERR_VENC_NOBUF	编码通道缓存分配失败
0xA008800F	ERR_VENC_BUF_FULL	编码通道缓存满
0xA0088010	ERR_VENC_SYS_NOTREADY	系统没有初始化
0xA0088012	ERR_VENC_BUSY	编码通道忙
0xA0088014	ERR_VENC_SAMESTATE	编码通道状态相同
0xA0088015	ERR_VENC_INVALIDSTATE	编码通道无效的状态
0xA0088016	ERR_VENC_INCORRECT_STATE_TRANSITION	编码通道不正确的状态转换
0xA0088017	ERR_VENC_INCORRECT_STATE_OPERATION	编码通道不正确的状态操作

Marketing of the state of the s 操構才提機構排機構 Will Little White High State of the State of

版权所有 © 珠海全志科技股份有限公司。保留一切权利



# 7.1 概述

VDEC 模块,即视频解码模块。本模块支持多路解码,且每路解码通道独立。

一个完整的本地文件播放流程如下所示。这里我们说的 VDEC 模块仅指送数据包到解码器解码输 出图像帧 (YUV 数据) 的过程。



# 7.2 功能描述

VDEC 模块接收待解码流输入,内部线程完成解码工作。

VDEC 模块支持 2 种数据输入输出方式:

绑定方式。主控模块将 VDEC 模块的解码通道和待解码流输入组件、显示组件绑定,组件间内部 传递数据。不需主控模块干预数据流入流出。

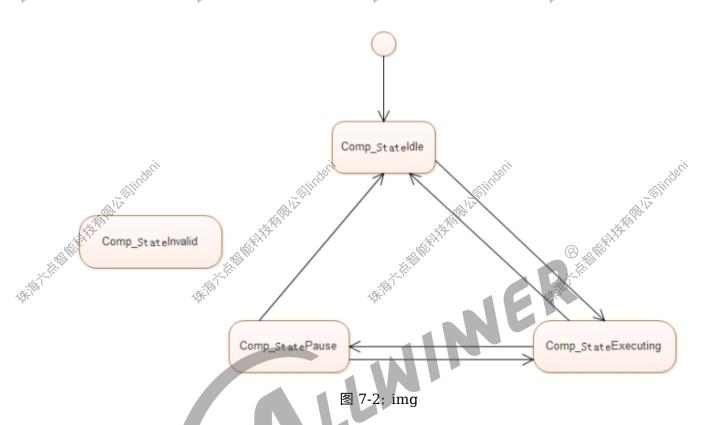
非绑定方式。主控模块调用 VDEC 模块的 mpi 层调用接口送入待解码流,获取解码帧、归还解 码帧。

VDEC 模块支持图像缩放、旋转等功能。宽度和高度可分别缩小至 1/8 倍,实际宽高缩放比例用 户可在解码器属性设置里面自定义设置 (1/2 1/4 1/8)。支持旋转角度为: 90、180、270 度。



# 7.3 状态转换与 API 接口

# 7.3.1 状态图



Vdec 组件内部状态设定为:

COMP StateLoaded: 组件初始创建状态。

COMP StateIdle: 组件完成初始化,参数设置、资源配置完毕,随时可以运行的状态。

COMP StateExecuting: 运行状态。

COMP\_StatePause: 暂停状态。

COMP\_StateInvalid: 异常状态。

API AW\_MPI\_VDEC\_CreateChn() 的实现过程会经过 COMP\_StateLoaded 状态,到达 COMP StateIdle。

组件内部状态转换的函数是: SendCommand(..., COMP\_CommandStateSet, 目标COMP\_State, ...);

版权所有 © 珠海全志科技股份有限公司。保留一切权利



# 7.3.2 API 和状态

能够引起状态变化的 API,见状态转换图。

每个 API 只能在允许的状态下调用,如果不在允许的状态下调用 API,则无效。

API 列表如下: (允许被调用的状态栏填写 Y)

	COMP_	COMP_		COMP_	
	StateI-	StateEx-	$COMP_{-}$	StateIn-	
API	dle	ecuting	StatePau	sevalid	说明
AW_MPI_					 引起状态转换。创建
VDEC		, deni		deni	组件,完成后状态为
CreateChn		The state of the s		Washing.	COMP_StateIdle
AW_MPI_	Y	A STATE OF THE STA		A THINK THE PARTY OF THE PARTY	销毁组件
VDEC_	XX		20,00	XX.	a with
DestroyChn	17-11 HV		12-11-11-11-11-11-11-11-11-11-11-11-11-1		S HAND
AW_MPI_	X	Y	Y		
VDEC_	\$K.,		άk,	-15	KINC.
GetChnAttr				W	
$AW_MPI_$	Y		Y		
VDEC_					
StartRecv					
Stream					
AW_MPI_		Y	Y		
VDEC_ à		à/		i	
StopRecv		Jiindell.		Tlindell	
Stream		ALIV TO		ALIV TON	ALV TO
AW_MPI_	XA	Y		XX	XX TANK
VDEC_ Pause	A PORTON A PROPERTY AND A PROPERTY A			XX	
AW_MPI_	1/12		Y		引起状态转换。到
VDEC_	-163,1		-1/2/1 T		Executing,再恢复
Resume	ζı,		×y1 *		到 Idle。
AW_MPI_	Y	Y	Y		
$VDEC\_Seek$					
AW_MPI_	Y	Y	Y		
VDEC_ Query					
AW_MPI_					createChn 后必须
VDEC_					马上 setcallback
Register		ine		ling	
Callback		Minde		Minde	Slin
RIV		MR Vallinderi		SELIZ	A TOP TO SERVICE OF THE PARTY O
Callback	, XA	(A)	, 2	XA CONTRACTOR OF THE PARTY OF T	A THE PARTY OF THE
	HE WHITE TO		, HANNE	HA TO Blinderi	A STATE OF THE STA
	- 15 - 15 - 15 - 15 - 15 - 15 - 15 - 15		1/2		The state of the s

版权所有 © 珠海全志科技股份有限公司。保留一切权利



	ALLWIMER		- Blindeni	- Allinderi		文档密级:秘密
	A TOP TO THE PERSON NAMED IN COLUMN	COMP	COMP_	COMP_		THE TOTAL PROPERTY OF THE PARTY
A REAL PROPERTY.	7	State I-	StateEx-	COMP StateIn-	×.	XA.
Hall The second	API	dle	ecuting	StatePausevalid	说明	
21		Y	Y	**************************************		
	VDEC_	` 1	1	441×	×y1*	
	SetStream					
	Eof					
	AW_MPI_	Y				
	VDEC_	1				
	ResetChn					
		Y				
	AW_MPI_ VDEC_	1	in	in		
	SetChn Param		allinde.	allinde		lling
	117	Y	N. C. C.	v Riv		WIND,
×	AW_MPI_	1	XI	1		XA TON
1237	VDEC_ GetChn	A PARTY NO.		Y Y	(B) A HOLE	Κ'
. 7/5		1/2				
,	Param	165,1 27,2		操物	THE THE	
	AW_MPI_	Y				
	VDEC_			4 MN "		
	SetProtocol					
	Param	77		JM ,		
	AW_MPI_	Y	Y	Y		
	VDEC_					
	GetProtocol					
	Param		Y Hinderi	<i>i</i> no.		
	AW_MPI_	Y	Y	Y Allinder.		Alin
	VDEC_	<u> </u>	SELIV.	WIV.		SELV
, X.	SendStream	XX	)	Y Y **********************************		XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
7.49	AW_MPI_	Y	Y	Y AND THE PARTY OF	ALL VIEW TO	
	VDEC_	1/-		T. T	7/-	
1	GetImage		••	-\$\disp\	W. T. L.	
	AW_MPI_	Y	Y	Y		
	VDEC_					
	ReleaseImage					
	AW_MPI_	Y				
	VDEC_					
	SetRotate					
	AW_MPI_	Y	Y	Y		
	VDEC		AB(i)	<sub>18</sub> ri		,
	GetRotate		- Allino	- Allino		_ Allin
	VDECGetRotate	Jin	Me La Alinderi	Signification of the state of t		HA HIRL TO HILL
XXX	+	A XXXX	7	NEW YORK	4	X
THE PARTY OF THE P		SE VINT		EL HETT	EL VINCE	

版权所有 © 珠海全志科技股份有限公司。保留一切权利



حالم		Alla	N. VIII	حالم
This contract of the second	COMP	COMP_	COMP_	A TANKS
s'	StateI-	StateEx-	COMP_StateIn-	
API	dle	ecuting	StatePausevalid	说明
AW_MPI_	**	Y	W. Francisco	- (%)
VDEC_				
GetChnLuma				
AW_MPI_	Y			
VDEC_				
SetVEFreq				
AW_MPI_	Y			
VDEC_				
SetVideo		A State of the sta	the latter in the little of the latter in th	O ANT THE REAL VERY LINE
StreamInfo		- Allin	- Allin	
AW_MPI_	Y	THE IN		
VDEC	A XXX	X.	A XAPE	A XXXX
ForceFrame				(S) All Marin
Package	-1621 - 152		162	
	ZX.		******	The state of the s
4 4 5 7 4	·> +/		4111	
.4 API 🦠	多考		1	
			Ma.	

# 7.4 API 参考

视频解码模块主要提供视频解码通道 (在本文档中通道等同于组件实例) 的创建和销毁、视频解码 通道的复位、开启和停止接收码流解码等功能。

# 7.4.1 AW\_MPI\_VDEC\_CreateChn

# 【描述】

创建视频解码通道

# 【语法】

ERRORTYPE AW MPI VDEC CreateChn(VDEC CHN VdChn, const VDEC CHN ATTR S \*pstAttr);

# 【参数】

参数名称	描述	输入/输出
VdChn	解码通道号。取值范围:[0, VDEC_MAX_CHN_NUM)。	 输入
pstAttr	视频解码通道属性指针	输入



# 【返回值】

	L-X
返回值	描述
0 非 0	成功。
U	TIK.

【需求】

头文件: mm\_comm\_vdec.h、mm\_common.h

库文件: libmedia mpp.so

# 7.4.2 AW\_MPI\_VDEC\_DestroyChn 【描述】 消毀视频解码通道 【语法】 RRORTYPF

ERRORTYPE AW\_MPI\_VDEC\_DestroyChn(VDEC\_CHN\_VdChn);

【参数】

参数名称	描述				7.7	输入/输出
VdChn	解码通道号。	取值范围: [(	O, VDEC_M	AX_CHN_	NUM)	 输入

【返回值】

返回值	描述	
0	成功	
非 0 、	失败,	参见错误码。



头文件: mm\_comm\_vdec.h, mm\_common.h

库文件: libmedia\_mpp.so

【注意】

无。

【举例】

无。

# $7.4.3~AW\_MPI\_VDEC\_GetChnAttr$

【描述】

获取解码通道属性。

【语法】

ERRORTYPE AW\_MPI\_VDEC\_GetChnAttr(VDEC\_CHN\_VdChn, VDEC\_CHN\_ATTR\_S \*pstAttr);

# 【参数】

参数名称	描述	输入/输出
VdChn	解码通道号。取值范围:[0, VDEC_MAX_CHN_NUM)。	输入
pstAttr	解码通道属性指针。	输出

# 【返回值】

返回值	描述
0	成功。
非 0	失败,参见错误码。

# 【需求】

头文件: mm comm vdec.h、mm common.h

库文件: libmedia\_mpp.so

【注意】

无。



# 【举例】

无。

# $7.4.4\ AW\_MPI\_VDEC\_StartRecvStream$

# 【描述】

开启解码通道,接收输入码流进行解码。组件状态转换为 Comp StateExecuting。

# 【语法】

ERRORTYPE AW MPI VDEC StartRecvStream(VDEC CHN/VdChn);

# 【参数】

参数名称 描述 输入/输出

VdChn 解码通道号。取值范围: [0, VDEC\_MAX\_CHN\_NUM)。 输入

# 【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,参见错误码。

# 【需求】

头文件: mm\_comm\_vdec.h, mm\_common.h

库文件: libmedia mpp.so

#### 【注意】

如果通道未创建,则返回失败 AW\_ERR\_VENC\_UNEXIST

如果当前已经开启接收,此接口也返回成功。

只有开启接收之后解码器才开始接收码流解码。

# 【举例】

无。



# 7.4.5 AW\_MPI\_VDEC\_StopRecvStream

# 【描述】

停止编码通道接收输入数据。

#### 【语法】

ERRORTYPE AW MPI VDEC StopRecvStream(VDEC CHN VdChn);

# 【参数】

描述			输入/输出
解码通道号。	取值范围:	[0, VDEC_MAX_CHN_NUM)。	输入
J)	SELV .	The state of the s	RIV.
	<i>?</i>	A A A A A A A A A A A A A A A A A A A	输入
-1654 -1654 -1654 -1654 -1654 -1654 -1654 -1654 -1654 -1654 -1654 -1654 -1654 -1654 -1654 -1654 -1654 -1654 -1	返回值	描述	
A.K.	0	成功	
	非 0	失败,参见错误码。	
		1 MIL	
	解码诵道号。	解码通道号。取值范围: 返回值 0	解码通道号。取值范围: [0, VDEC_MAX_CHN_NUM)。

# 【返回值】

返回值	描述
0	成功
非 0	失败,参见错误码。

# 【需求】

头文件: mm\_comm\_vdec.h、mm\_common.h

库文件: libmedia mpp.so

#### 【注意】

如果通道未创建,则返回失败。

此接口并不判断当前是否停止接收,即如果当前已经停止接收,调用此接口也返回成功。 此接口用于解码通道停止接收码流解码,在解码通道销毁或复位前必须停止接收码流。 调用此接口仅停止接收码流解码,码流 buffer 并不会被清除。

# 【举例】

无。

# 7.4.6 AW\_MPI\_VDEC\_Pause

【描述】



暂停解码,解码组件状态转为 COMP StatePause。

# 【语法】

ERRORTYPE AW MPI\_VDEC\_Pause(VDEC\_CHN VdChn);

# 【参数】

参数名称	描述		输入/输出
VdChn	解码通道号。	取值范围: [0, VDEC_MAX_CHN_NUM)。	 输入

#### 【返回值】

返回值 描述

0 成功
非 0 失败,参见错误码。

mm\_common.h

# 【需求】

头文件: mm\_comm\_vdec.h、mm\_common.h

库文件: libmedia mpp.so

#### 【注意】

只能从 Comp\_StateExecuting 状态转换到 Comp\_StatePause 状态,其他状态下返回失败。 此接口并不判断当前是否已经暂停,即如果当前已经暂停,调用此接口也返回成功。

# 【举例】

**%** 

# 7.4.7 AW\_MPI\_VDEC\_Resume

# 【描述】

继续解码,解码组件状态从 COMP\_StatePause 转为 Comp\_StateExecuting。

# 【语法】

ERRORTYPE AW MPI VDEC Resume(VDEC CHN VdChn);

#### 【参数】



参数名称 描述 输入/输出 解码通道号。取值范围: [0, VDEC\_MAX\_CHN\_NUM)。 VdChn 输入

【返回值】

返回值	描述
0	成功
<b>‡</b> 0	失败,参见错误码。

【需求】

・エ思】 只能从状态 Comp\_StatePause 状态执行,其他状态下返回失败。 【举例】 无。 7.4.8 AW\_MPI\_VDEC See」 【描述1.magni

【描述》

设置视频解码通道完成跳播后播放的准备。

【语法】

ERRORTYPE AW MPI VDEC Seek(VDEC CHN VdChn);

【参数】

参数名称	描述			输入/输出
VdChn	解码通道号。	取值范围:	[0, VDEC_MAX_CHN_NUM)。	输入

【返回值】

返回值	描述	A La Jime
0	成功	-134 Mary



返回值 描述 失败,参见错误码。 非 0

【需求】

头文件: mm comm vdec.h、mm common.h

库文件: libmedia mpp.so

【注意】

无

【举例】

7.4.9 AW\_MPI\_VDEC\_Query

【描述】

查询视频解码通道状态。

【语法】

VDEC ERRORTYPE AW\_MPI\_VDEC\_Query(VDEC\_CHN\_VdChn, VDEC\_CHN\_STAT\_S \*pst-

【参数】

输入/输出 参数名称 描述 输入 VdChn 解码通道号。取值范围: [0, VDEC\_MAX\_CHN\_NUM)。 通道当前状态 输出 pstStat

【返回值】

返回值 描述 0 成功 非 0 失败,参见错误码。

【需求】

头文件: mm\_comm\_vdec.k mm\_common.h



库文件: libmedia\_mpp.so

【注意】

无。

【举例】

无。

# $7.4.10\ AW\_MPI\_VDEC\_RegisterCallback$

【描述】

设置解码通道回调

【语法】

ERRORTYPE AW\_MPI\_VDEC\_RegisterCallback(VDEC\_CHN\_VdChn, MPPCallback-Info \*pCallback);

# 【参数】

参数名称	描述	输入/输出
VdChn	解码通道号。取值范围: [0, VDEC_MAX_CHN_NUM)。	输入
pCallback	回调信息。	输出

#### 【返回值】

返回值	描述	A TOP OF THE PERSON NAMED IN COLUMN TO THE PERSON NAMED IN COLUMN
0	成功	W. W
非 0	失败,	参见错误码。

# 【需求】

l 头文件: mm\_comm\_vdec.h、mm\_common.h

l 库文件: libmedia mpp.so

【注意】

无。

【举例】



无态

# 7.4.11 AW\_MPI\_VDEC\_SetStreamEof

【描述】

设置解码输入码流结束标志

【语法】

ERRORTYPE AW\_MPI\_VDEC\_SetStreamEof(VDEC\_CHN VdChn, BOOL bEofFlag);

#### 【参数】

//^\'	//^\*	//^\*	
参数名称	描述		输入/输出
VdChn	解码通道号。取值范围:	[0, VDEC_MAX_CHN_NUM)。	输入
bEofFlag	结束标记,TRUE:码流	流结束;FALSE:码流未结束。	输入

【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,参见错误码。

【需求】

头文件: mm\_comm\_vdec.h、mm\_common.h

库文件: libmedia mpp.so

【注意】

无。

【举例】

无。

# 7.4.12 AW\_MPI\_VDEC\_ResetChn

【描述】

复位解码通道,但不会重置已设置的解码参数,不会释放已分配的码流缓冲和视频帧。解码通道



复位后,缓冲数据都被清空,随时等待再次解码。

# 【语法】

ERRORTYPE AW MPI\_VDEC\_ResetChn(VDEC\_CHN VdChn);

# 【参数】

参数名称	描述		输入/输出
VdChn	解码通道号。	取值范围: [0, VDEC_MAX_CHN_NUM)。	 输入

# 【返回值】

返回值 描述
0 成功
非 0 失败,参见错误码。
mm\_common.h

#### 【需求】

头文件: mm\_comm\_vdec.h、mm\_common.h

库文件: libmedia mpp.so

#### 【注意】

Reset 并不存在的通道,返回失败 AW ERR VENC UNEXIST。

如果一个通道没有停止接收码流而 reset 通道,则返回失败。

# 【举例】

洗。

# 7.4.13 AW\_MPI\_VDEC\_SetChnParam

# 【描述】

设置解码通道参数。

#### 【语法】

ERRORTYPE AW\_MPI\_VDEC\_SetChnParam(VDEC\_CHN VdChn, VDEC\_CHN\_PARAM\_S\* pstParam);

#### 【参数】



参数名称 描述 输入/输出 解码通道号。取值范围: [0, VDEC\_MAX\_CHN\_NUM)。 VdChn 输入 解码通道参数 输入 pstParam

【返回值】

返回值 描述 0 成功 非 0 失败,参见错误码。

【需求】

头文件: mm comm vdec.h, mm common.h

库文件: libmedia mpp.so

【注意】

无。

【举例】

无。

# God. 7.4.14 AW\_MPI\_VDEC\_GetChnParam

【描述】

获取解码通道参数。

【语法】

ERRORTYPE AW MPI VDEC GetChnParam(VDEC CHN VdChn, VDEC CHN PARAM S\* pstParam);

#### 【参数】

参数名称	描述	输入/输出
VdChn	解码通道号。取值范围:[0, VDEC_MAX_CHN_NUM)。	输入
pstParam	解码通道参数指针	输出



返回值	描述	-1X4 (A)
0	成功	HELDER TO SERVICE AND ADDRESS OF THE PERSON
非 0	失败,	参见错误码。

【需求】

头文件: mm\_comm\_vdec.h、mm\_common.h

库文件: libmedia mpp.so

【注意】

7.4.15 AW\_MPI\_VDEC\_SendStream
【描述】
发送待解码码流给解码通道进行解码。
【语法】
\*\*RRORTYRE AW\_MPI\_VPpstStream\*\* \*pstStream, int s32MilliSec);

# 【参数】

参数名称	描述		输入/输出
VdChn	解码通道号。取值范围:[0, VDEC_MAX_CF	IN_NUM)。	输入 ※
pstStream	码流信息结构指针	Y	输入
s32MilliSec	发送码流超时时间。取值范围: [-1,+∞)-1: 凡	且塞 0: 非阻塞 >0: 超时时间	输入

# 【返回值】

返回值	描述	
0	成功	
0 非00 <sup>deri</sup>	失败,	参见错误码。



# 【需求】

头文件: mm\_comm\_vdec.h、mm\_common.h

库文件: libmedia mpp.so

#### 【注意】

超时时间 s32MilliSec 的含义,-1 表示必须等到该 stream 进入解码通道 vbvBuffer 中; 0: 立 刻返回结果,如果当前等待解码的 vbvBuffer 满,返回失败; >0: 如果 vbvBuffer 满,等待到 设定的时间再返回超时。

仅用于组件非绑定方式

# 7.4.16 AW\_MPI\_VDEC\_GetImage

# 【描述】

获取解码后的帧。

# 【语法】

DEC. ERRORTYPE AW\_MPI\_VDEC\_GetImage(VDEC\_CHN VdChn, VIDEO\_FRAME\_INFO\_S \*pstFrameInfo,int s32MilliSec);

	112	112	110	n .	112
×1	参数名称	描述			输入/输出
5.	VdChn	解码通道号。取值范围	: [0, VDEC_MAX	_CHN_NUM)。	输入
	pstFrameInfo	解码帧结构体指针	The same of the sa	7-12	输出
	s32MilliSec	获取解码帧的超时时间	。取值范围:	- <u>*</u> ***********************************	输入
		[-1, +∞)-1:阻塞 0:	非阻塞 0: 超时时间	],单位毫秒	

#### 【返回值】

返回值	描述	
0	成功	
非0点	失败,	参见错误码。

# 【需求】



头文件: mm\_comm\_vdec.h, mm\_common.h

库文件: libmedia\_mpp.so

【注意】

仅用于组件非绑定方式

【举例】

无。

# $7.4.17 AW\_MPI\_VDEC\_ReleaseImage$

【描述】

释放解码帧给解码通道。

【语法】

ERRORTYPE AW\_MPI\_VDEC\_ReleaseImage(VDEC\_CHN VdChn, VIDEO\_FRAME\_INFO\_S \*pstFrameInfo);

# 【参数】

参数名称	描述	输入/输出
VdChn	解码通道号。取值范围:[0, VDEC_MAX_CHN_NUM)。	输入
pstFrameInfo	帧结构体指针,使用时只需填写 pstFramaInfo->mId 即可。	输入

# 【返回值】

返回值	描述
0	成功。
非 0	失败,参见错误码。

# 【需求】

头文件: mm comm vdec.h、mm common.h

库文件: libmedia\_mpp.so

【注意】

仅用于组件非绑定方式



【举例】

泛无。

# 7.4.18 AW\_MPI\_VDEC\_SetRotate

# 【描述】

设置解码旋转顺时针角度。

# 【语法】

ERRORTYPE AW\_MPI\_VDEC\_SetRotate(VDEC\_CHN VdChn, ROTATE\_E enRotate);

#### (参数)

参数名称	描述		输入/输出
VdChn		[0, VDEC_MAX_CHN	_NUM)。 输入
enRotate	旋转角度枚举类型。静态	<b>态属性</b> 。	输入

# 【返回值】

返回值	描述	
0	成功	
非 0	失败,参见错误码	0

# 【需求】

头文件: mm\_comm\_vdec.h、mm\_common.h

库文件: libmedia\_mpp.so

# 【注意】

必须在解码开始前设置,解码过程中设置无效。

# 【举例】

无。

# 7.4.19 AW\_MPI\_VDEC\_GetRotate

描述】



获取解码旋转顺时针角度。

# 【语法】

ERRORTYPE AW\_MPI\_VDEC\_GetRotate(VDEC\_CHN VdChn, ROTATE\_E \*penRotate);

# 【参数】

参数名称	描述	输入/输出
VdChn	解码通道号。取值范围:[0, VDEC_MAX_CHN_NUM)。	输入
penRotate	旋转角度枚举类型指针	输出

#### 【返回值】

返回值 描述
0 成功
非 0 失败,参见错误码。

# 【需求】

头文件: mm\_comm\_vdec.h、mm\_common.h

库文件: libmedia mpp.so

【注意】

无。

(举例)

无。

# 7.4.20 AW\_MPI\_VDEC\_ReopenVideoEngine

# 【描述】

重置解码引擎。解码库内部销毁 vbvBuffer 和 frame buffers。需要在解码通道检测到图像分辨率变化并通过 callback 通知之后调用。

# 【语法】

ERRORTYPE AW\_MPI\_VDEC\_ReopenVideoEngine(VDEC\_CHN VdChn);

#### 【参数】



参数名称 描述 输入/输出

解码通道号。取值范围: [0, VDEC\_MAX\_CHN\_NUM)。 VdChn 输入

【返回值】

返回值 描述 0 成功 非 0 失败,参见错误码。

【需求】

头文件: mm\_comm\_vdec.h、mm\_common.h

库文件: libmedia mpp.so

【注意】

【举例】

无。

30 September 1988 Sep 7.4.21 AW\_MPI\_VDEC\_SetVEFreq

【描述】

设置 VE 运行频率。

【语法】

ERRORTYPE AW\_MPI\_VDEC\_SetVEFreq(VDEC\_CHN VeChn, int nFreq);

【参数】

参数名称 描述 输入/输出 解码通道号。取值范围: [0, VDEC MAX CHN NUM) 和 MM INVALID CHN。 VeChn 输入 频率,单位: MHz nFreq 输入

【返回值】

返回值 描述 成功 非 0 失败, 参见错误码。



# 【需求】

头文件: mm\_comm\_vdec.h、mm\_common.h

库文件: libmedia mpp.so

【注意】

【举例】

无。

# $7.4.22\ AW\_MPI\_VDEC\_SetVideoStreamInfo$

# 【描述】

设置码流参数,供解码器配置解码参数进行解码。

#### 【语法】

ERRORTYPE AW\_MPI\_VDEC\_SetVideoStreamInfo(VDEC\_CHN VdChn, VideoStreamInfo \*pVideoStreamInfo)

# 【参数】

参数名称	描述	输入/输出
VdChn	解码通道号。取值范围: [0, VDEC_MAX_CHN_NUM)。	输入
pVideoStreamInfo	码流信息	输入

# 【返回值】

返回值	描述
0	成功。
非 0	失败,参见错误码。

# 【需求】

头文件: mm comm vdec.h、mm common.h

库文件: libmedia\_mpp.so

【注意】

【举例】



无。

# $7.4.23\ AW\_MPI\_VDEC\_ForceFramePackage$

# 【描述】

设置帧边界是否确定。如果帧边界确定,解码器将跳过检测帧头的过程,加快解码速度。

# 【语法】

ERRORTYPE AW MPI VDEC ForceFramePackage(VDEC CHN VdChn, BOOL bFlag);

#### 【参数】

参数名称	描述	NA TOP TO SERVICE AND THE PROPERTY OF THE PROP	输入/输出
VdChn	解码通道号。取值范围:	[0, VDEC_MAX_CHN_NUM)。	输入
bFlag	帧边界是否确定	Harris Control of the	输入
	(本)	W. B.	(R)
回值】		ANG.	<u> </u>
	返回值	描述	

# 【返回值】

返回值	描述	
0	成功	
非 0	失败,参见错误码。	

# 【需求】

头文件: mm\_comm\_vdec.h、mm\_common.h

库文件: libmedia mpp.so

【注意】

【举例】

无。

# 7.5 数据结构说明

# 7.5.1 VIDEO\_MODE\_E

【说明】



# 解码器输入码流方式

# 【定义】

typedef enum VIDEO\_MODE\_E {

VIDEO MODE STREAM = 0, /\* send by stream \*/

VIDEO MODE FRAME, /\* send by frame \*/

VIDEO MODE BUTT

} VIDEO MODE E;

# 【成员】

inderit	- Jijing	eri (S. H.
成员名称	描述	A VV
VIDEO_MODE_STREAM	流模式	AND THE PERSON NAMED IN COLUMN TO A COLUMN
VIDEO_MODE_FRAME	帧模式	O MANTE
VIDEO_MODE_BUTT	无效	The state of the s
ALLWII	M	sti ssti

【注意事项】

无。

【相关数据类型及接口】

无。

VDEC\_CHN\_ATTR\_S

【说明】

【定义】

typedef struct VDEC\_CHN\_ATTR\_S {

PAYLOAD\_TYPE\_E mType; /\* video type to be decoded \*/

unsigned int mBufSize; /\* stream buf size(Byte) \*/

unsigned int mPriority; /\* priority \*/

unsigned int mPicWidth; /\* max pic width \*/

unsigned int mPicHeight; /\* max pic height \*/

int minitRotation; //clockwise rotation: val=0 no rotation, val=1 90 degree; val=2 180 degree, val=3 270 degree



PIXEL\_FORMAT\_E mOutputPixelFormat;

union {

VDEC\_ATTR\_JPEG\_S mVdecJpegAttr; /\* structure with jpeg or mjpeg type \*/

VDEC\_ATTR\_VIDEO\_S mVdecVideoAttr; /\* structure with video ( h264/mpeg4) \*/

**}**;

int mnFrameBufferNum; /\* set frame number which is malloc by vdeclib, 0 means any number. only valid to jpeg.\*/

int mExtraFrameNum; /\* let vdeclib malloc more frame buffer base on initial frame number.\*

} VDEC CHN ATTR S;

# 【成员】

成员名称	描述		
mType	解码类型	***	
mBufSize	vbvBuffer 的长度	。单位字节。取值范围	: [0, +∞)。0 表示解码通道
	自行决定。静态属性	生。	
mPriority	未使用。	M	
mPicWidth	解码输出图片的最大	大宽度。范围: [0,384	40]。0 表示没有限制。如果
	原图宽度超过最大	宽度,解码通道将进行	压缩,压缩比为
	1/2,1/4,1/8。原图	宽度不能超过 3840,	否则无法解码。静态属性。
mPicHeight	解码输出图片的最大	大高度。范围:[0,216	60]。0 表示没有限制。如果
Ellinde	原图高度超过最大高	高度,解码通道将进行	压缩,压缩比为
THE ITE	1/2,1/4,1/8。原图	高度不能超过 2160,	否则无法解码。静态属性。
mInitRotation	解码输出图片旋转角	角度。范围: [0,3]。1	: 顺时针旋转 90 度 2: 顺
<u> </u>	时针旋转 180 度;	3: 顺时针旋转 270 層	度。静态属性。

mOutputPixelFormat解码输出图片相素格式。静态属性。

mVdecJpegAttr 未使用。 mVdecVideoAttr 未使用。

mnFrameBufferNumset frame number which is malloc by vdeclib, 0 means any

number. only valid to jpeg

 $mExtraFrameNum \quad let \ vdeclib \ malloc \ more \ frame \ buffer \ base \ on \ initial \ frame$ 

number

#### 【注意事项】

无。

【相关数据类型及接口】

# 7.5.2 VDEC\_STREAM\_S

【说明】

解码器输入码流

【定义】

typedef struct VDEC STREAM S {

unsigned char* pAdd	r; /* stream address	s*/	nindeni	
unsigned int mLen; /*	stream len */	S*/	alika la lilindeni	
uint64_t mPTS; /* time stamp */				
BOOL mbEndOfFram	e; /* is the end of a	frame */		
BOOL mbEndOfStream; /* is the end of all stream */				
} VDEC_STREAM_S;				
【成员】		Mı,		
	成员名称	描述		
	pAddr	输入码流地址		
Mille Little Brit	mLen deri	输入码流数据长度	deni	
A STATE OF THE PARTY OF THE PAR	mPTS	时间戳		
THE PARTY OF THE P	${\it mbEndOfFrame}$	是否是一帧数据的结束		
	mbEndOfStream	是否是码流的结尾		

【注意事项】

无。

【相关数据类型及接口】

无。

# 7.5.3 VDEC\_DECODE\_ERROR\_S

【说明】

# 解码器错误类型

# 【定义】

typedef struct VDEC\_DECODE\_ERROR\_S.{

int mFormatErr; /\* format error. eg: do not support filed \*/

int mPicSizeErrSet; /\* picture width or height is larger than chnnel width or height\*/

int mStreamUnsprt; /\* unsupport the stream specification \*/

int mPackErr; /\* stream package error \*/

int mPrtclNumErrSet; /\* protocol num is not enough. eg: slice, pps, sps \*/

int mRefErrSet; /\* refrence num is not enough \*/

int mPicBufSizeErrSet; /\* the buffer size of picture is not enough \*/

} VDEC\_DECODE\_ERROR\_S;

【成员】

成员名称

描过

mFormatErr
mPicSizeErrSet
mStreamUnsprt
mPackErr
mPrtclNumErrSet
mRefErrSet
mPicBufSizeErrSet

【注意事项】

无。

【相关数据类型及接口】

无。

# 7.5.4 VDEC\_CHN\_STAT\_S

【说明】

解码通道属性



# 【定义】

typedef struct VDEC\_CHN\_STAT\_S {

PAYLOAD\_TYPE\_P mType; /\* video type to be decoded \*/

unsigned int mLeftStreamBytes; /\* left stream bytes waiting for decode \*/
unsigned int mLeftStreamFrames;

/\* left frames waiting for decode,only valid for H264D\_MODE\_FRAME \*/
unsigned int mLeftPics; /\* pics waiting for output \*/

BOOL mbStartRecvStream; /\* had started recv stream? \*/

unsigned int mRecvStreamFrames; /\* how many frames of stream has been received. valid when send by frame. \*/

unsigned int mDecodeStreamFrames;

/\* how many frames of stream has been decoded. valid when send by frame. \*/

VDEC\_DECODE\_ERROR\_S mVdecDecErr; /\* information about decode error \*/

} VDEC CHN STAT S;

# 【成员】

描述
解码类型
解码器输入有多少数据待解码
解码器输入待解码数据有多少帧
解码器输出剩余多少图片没有取
是否开始接收待解码数据
接收到的待解码数据有多少帧(按帧传送方式时),未使用。
解码出来的数据有多少帧(按帧传送方式时),未使用。
解码器错误类型,未使用。

#### 【注意事项】

无。

【相关数据类型及接口】

无。



# 7.5.5 VDEC\_CHN\_PARAM\_S

# 【说明】

解码通道参数

# 【定义】

typedef struct VDEC CHN PARAM S {

int mChanErrThr;

int mChanStrmOFThr;

int mDecMode;

int mDecOrderOutput;

VIDEO\_FORMAT\_E mVideoFormat;

COMPRESS\_MODE\_E mCompressMode;

} VDEC\_CHN\_PARAM\_S;

# 【成员】

, menansumor mi;				
mDecMode;	18 17 dillinderi	A REAL TO LINE	Beli	All Merillinderi
mDecOrderOutput;	A TOP TO SERVICE STATE OF THE	A A A A A A A A A A A A A A A A A A A	.⊀.	ETHIRLID .
DEO_FORMAT_E mVi	deoFormat;	Little Reserved	© Harriston	»*
OMPRESS_MODE_E n	nCompressMo	ode;		
DEC_CHN_PARAM_S	;			
<b>戊</b> 员】		11/1/2	J.	
成员名称	描述	I'A		
mChanErrThr	未使用			
mChanStrmOFThr	未使用			
mDecMode	解码模式。0:	普通解码; 1: I 帧和 P	, 帧解码; 2: I 帧解	码。 感
<b>Blind</b>	目前只支持 0	和 1。动态属性。	<i>&gt;</i>	- Allino
mDecOrderOutput	未使用	THE IV		A TOP TO SERVICE OF THE PARTY O
mVideoFormat	未使用	A STATE OF THE STA		\$ ***
mCompressMode	未使用			
-/ %		-/ %	-/45	

# 【注意事项】

无。

【相关数据类型及接口】

无。

# 7.5.6 VDEC\_PRTCL\_PARAM\_S

【说明】



# 【定义】

typedef struct VDEC\_PRTCL\_PARAM\_S {

int mMaxSliceNum; /\* max slice num support \*/

int mMaxSpsNum; /\* max sps num support \*/

int mMaxPpsNum; /\* max pps num support \*/

int mDisplayFrameNum; /\* display frame num \*/

} VDEC\_PRTCL\_PARAM\_S;

# 【成员】

- Piir		nin.	
成员名称	描述	WIN.	
mMaxSliceNum	最大 slig	e 数量,未使	—— !用。
mMaxSpsNum	最大 sp	数量,未使用	∄。
mMaxPpsNum	最大 pp	s 数量,未使原	∄。 【
mDisplayFrameNur	n、显示帧数	7量,未使用。	¥
	M	Wa.	

【注意事项】

无。

【相关数据类型及接口】

无。

# 7.6 错误码

错误代码	宏定义	A. A	1	描述	**************************************
0xA0058002	ERR_VDEC_INV	ALID_CHNII	)	通道 ID 超出合法范围	
0xA0058003	ERR_VDEC_ILL	EGAL_PARAN	М	参数超出合法范围	
0xA0058004	ERR_VDEC_EXI	ST		试图申请或者创建已经存在	生的设备、
				通道或者资源	
0xA0058006	ERR_VDEC_NU	LL_PTR		函数参数中有空指针	
0xA0058007	ERR_VDEC_NO	Γ_CONFIG		使用前未配置	
0xA0058008	ERR_VDEC_NO	Γ_SUPPORT		不支持的参数或者功能。	
0xA0058009	ERR_VDEC_NO	Γ_PERM	ninden	该操作不允许,如试图修改	<b></b>
41/2	Z/1/2		SERVER STATE OF THE SERVER	配置参数	
0xA0058005	ERR_VDEC_UN	EXIST	AN TO SERVICE STATE OF THE SER	试图使用或者销毁不存在的	り设备、
A CONTRACTOR OF THE PARTY OF TH				通道或者资源	
· ·			A CONTRACTOR OF THE PARTY OF TH	A STATE OF THE STA	, t



William Committee of the Committee of th

Will Little White Head of the late of the

V11/.		A11/
错误代码	宏定义	描述
0xA005800C	ERR_VDEC_NOMEM	
0xA005800D	ERR_VDEC_NOBUF	分配缓存失败,如申请的数据
	AN THE STATE OF TH	缓冲区太大
0xA005800E	ERR_VDEC_BUF_EMPTY	缓冲区中无数据
0xA005800F	ERR_VDEC_BUF_FULL	缓冲区中数据满
0xA0058010	ERR_VDEC_SYS_NOTREADY	系统没有初始化或没有加载相应模块
0xA0058012	ERR_VDEC_BUSY	VENC 系统忙
0xA0058014	ERR_VDEC_SAMESTATE	状态相同
0xA0058015	ERR_VDEC_INVALIDSTATE	无效的状态
0xA0058016	ERR_VDEC_INCORRECT_STATE_TRANSITION	状态转换出错
0xA0058011	ERR_VDEC_BADADDR	非法地址

THE REPORT OF THE PARTY OF THE

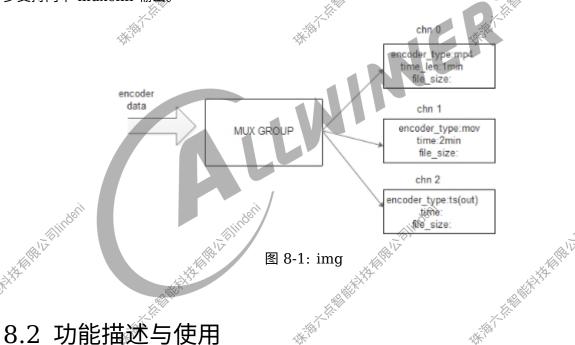
版权所有 © 珠海全志科技股份有限公司。保留一切权利



# 8.1 概述

mux 模块,即文件封装模块。本模块以 muxGroup 为单位,一个 muxGroup 包含一个或多个 mux 通道,属于同一 muxGroup 的 mux 通道对输入的同一视频、音频编码数据流进行封装。

使用 muxGroup 的原因是:有时不仅需要把视频、音频数据流封装为 mp4 文件本地存储,同时 还需封装为 raw 码流或 ts 码流通过网络传输,这时需要对接收的同一视频、音频流数据同时进行 2 路封装处理(1 路处理对应一个 mux 通道),故设计 muxGroup。目前一个 muxGroup 最 多支持两个 muxchn 输出。



mux 通道内部线程完成封装和写入(写卡)的功能。

mux 模块只支持绑定方式的数据输入。主控模块必须将 muxGroup 组件和编码组件绑定。mux 模块的数据输入对象是 muxGroup。muxChannel 在 group 内部接收处理数据。

一个 muxGroup 输入端最多支持 2 个输入绑定(video enc/audio enc)。





本地存储操作时,应用 app 仅需要设置录像文件 fd 与 mux 进行互动。录像时,mux 会在当前录像文件结束前 10 秒向 app 应用发消息 MPP\_EVENT\_NEXT\_FD,获取下一个录像文件的句柄 fd(mux 内部会对此 fd 进行 dup 操作,应用 app 应自己 close 该 fd);mux 在封装完一个文件后会向 app 应用发送 MPP\_EVENT\_RECORD\_DONE 消息,表示一个录像文件已经完成。

对于非本地存储操作(如 ts 网络传输),设置 chn 属性时可设置 mCallbackOutflag 标志(数据操作由外部回调完成),mux 会向应用发出消息 MPP\_EVENT\_BSFRAME\_AVAILABLE 消息,由应用自行处理数据。

mux 模块本身不对数据流进行任何的拷贝操作,所有的数据 buffer 均由输入端绑定通道(venc 或者 aenc 组件实例)提供。因此在销毁 muxGroup 和 muxGroup 的输入绑定端的时候,mux-Group 必须先于输入绑定端组件销毁(销毁 muxGroup 时会处理所有关联数据 buffer),将相关的编码数据 packet buffer 还给 yenc 或者 aenc 组件。

# 8.2.1 muxGroup和 muxChannel



如上图所示: muxGroup 和 VEnc,AEnc 组件交互数据,Group 下的各 muxChannel 处理同样的音视频数据,同时满足不同的封装要求。

Group 参数是配置给 Group 下所有 channel 的参数。

channel 参数是单独设置给指定的 channel 的。

Group 参数和 channel 参数的优先级为: channel > group。即如果某个 channel 的 channel 参数和 group 参数有冲突,那么以 channel 参数为准。

muxGroup、属性和 muxChannel 属性的关系也是: channel > group。

muxChannel 必须归属于某个 muxGroup 之下,muxGroup 同一操作其下的 channel 如 start, stop, reset 等。channel 和 Group 的隶属关系,由调用者在创建 muxChannel 时指定。必须先创建 muxGroup,再创建 muxChannel。



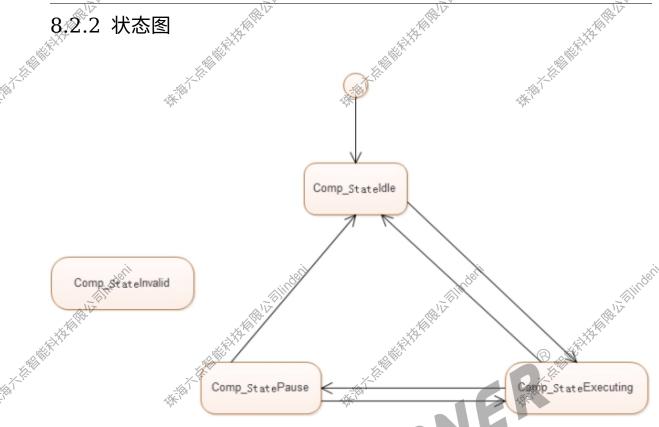


图 8-3: img

demux 组件内部状态设定为:

1 COMP StateLoaded:组件初始创建状态。

1 COMP\_StateIdle:组件完成初始化,参数设置、资源配置完毕,随时可以运行的状态。

1 COMP StateExecuting: 运行状态。

I COMP StatePause: 暂停状态。

l COMP StateInvalid: 异常状态。

AW\_MPI\_MUX\_CreateChn() 的实现过程会经过 COMP\_StateLoaded 状态,到达COMP\_StateIdle。

组件内部状态转换的函数是: SendCommand(..., COMP\_CommandStateSet, 目标COMP\_State, ...);

# 8.2.3 API 和状态

能够引起状态变化的 API, 见状态转换图。

每个 API 只能在允许的状态下调用,如果不在允许的状态下调用 API,则无效。



API 列表如下: (允许被调用的状态栏填写 Y)

. <u>/v=X</u> *	λ-X*		√-X *		λ-X '		
W.			COMP	_StateIdle	COMP_Pause	COMP	_StateExec
AV	V_MPI_MUX_CreateGrp	163,1	1				161
ΑV	V_MPI_MUX_DestroyGrp	×y1.			×,1,		ху1,
ΑV	V_MPI_MUX_StartGrp		Y				
ΑV	V_MPI_MUX_StopGrp				Y	Y	
ΑV	V_MPI_MUX_GetGrpAttr		Y			Y	
ΑV	V_MPI_MUX_SetGrpAttr		Y			Y	
ΑV	V_MPI_MUX_SetH264SpsPpsInfo		Y			Y	
ΑV	V_MPI_MUX_SetH265SpsPpsInfo		Y			Y	
ΑV	V_MPI_MUX_CreateChn		Y	ight		Y	
ΑV	V_MPI_MUX_DestroyChn		Y	Minde		Yinde	
ΑV	WMPI_MUX_GetChnAttr		Y	RIV		ĮνY	
AV	MPI_MUX_SetChnAttr		Y			Y	
ķÂV	V_MPI_MUX_SwitchFd		HELD THE THE PERSON NAMED IN COLUMN TO PERSO		O LENGTON	Y	
ΑV	V_MPI_MUX_RegisterCallback	7-	Y				7-12
ΑV	V_MPI_MUX_GetCacheStatus	""(1515·1	Y		Y	Y	1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1
ΑV	V_MPI_MUX_SetMuxCacheDuration		Y	NE			
ΑV	${ m V\_MPI\_MUX\_SetSwitchFileDurationFull}$	Policy	Y	As .			
AV	${ m V\_MPI\_MUX\_GetSwitchFileDurationPosition}$	Policy	Y				

# 8.3 API 参考

mux 模块主要提供 muxGroup, mux 通道 (在本文档中通道等同于组件实例) 的创建和销毁、通道的复位、开启和停止等功能。

# 8.3.1 AW\_MPI\_MUX\_CreateGrp

# 【描述】

创建 muxGroup。

# 【语法】

ERRORTYPE AW\_MPI\_MUX\_CreateGrp(MUX\_GRP muxGrp, MUX\_GRP\_ATTR\_S \*pGrpAttr);

【参数】



参数名称	描述		输入/输出
muxGrp	MUX GROUP 号。取值范	围:[0, MUX_MAX_GRP_NUM)	。輸入
pGrpAttr	MUX GROUP 属性指针	-165 -165 -165 -165 -165 -165 -165 -165	输入

【返回值】

返回值	描述	
0	成功	
非 0	失败,	参见错误码。

8.3.2 AW\_MPI\_MUX\_DestroyGrp

【描述】

消毀 muxGroup

【语法】

RRORT\*\*

ERRORTYPE AW MPI\_MUX\_DestroyGrp(MUX\_GRP muxGrp);

【参数】

参数名称	描述《	N. T. C.		输入/输出
muxGrp	MUX GROUP 号。	取值范围: [0, MUX_MAX_GRF	_NUM)。	输入

【返回值】

描述 返回值 成功 失败, 参见错误码。

(注意】 一



# 8.3.3 AW\_MPI\_MUX\_StartGrp

【描述】

无。

启动 muxGroup 接收数据。

【语法】

ERRORTYPE AW\_MPI\_MUX\_StartGrp(MUX\_GRP muxGrp);

【参数】

参数名称 描述 输入/输出 muxGrp MUX GROUP 号。取值范围: [0, MUX\_MAX\_GRP\_NUM)。 输入

【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,参见错误码。

【注意】

无。

【举例】

无。

# $8.3.4~AW\_MPI\_MUX\_StopGrp$

【描述】

停止 muxGroup 接收数据。

【语法】



ERRORTYPE AW\_MPI\_MUX\_StopGrp(MUX\_GRP muxGrp);

# 【参数】

参数名称	描述	THE THE PERSON NAMED IN COLUMN TO TH	洪湖	输入/输出
muxGrp	MUX GROUP 号。	取值范围: [0, MUX_MAX_	GRP_NUM)。	输入

【返回值】

【注意】

无。

【举例】

无。

# 8.3.5 AW\_MPI\_MUX\_GetGrpAttr

【描述】

获取 muxGroup 属性。

【语法】

ERRORTYPE AW\_MPI\_MUX\_GetGrpAttr(MUX\_GRP muxGrp, MUX\_GRP\_ATTR\_S \*pGrpAttr);

# 【参数】

参数名称	描述	输入/输出
muxGrp	MUX GROUP 号。取值范围: [0, MUX_MAX_GRP_NUM)。	输入
pGrpAttr	muxGroup 属性指针	输出

【返回值】

版权所有 © 珠海全志科技股份有限公司。保留一切权利

275



 返回值
 描述

 0
 成功

 非 0
 失败,参见错误码。

【注意】

无。

【举例】

无。

# 8.3.6 AW\_MPI\_MUX\_SetGrpAttr

【描述】

设置 muxGroup 属性。

【语法】

ERRORTYPE AW\_MPI\_MUX\_SetGrpAttr(MUX\_GRP muxGrp, MUX\_GRP\_ATTR\_S \*pGrpAttr);

# 【参数】

参数名称	描述	à	输入/输出
dmxChn	MUX GROUP 号。取值范围:	[0, MUX_MAX_GRP_NUM)。	 输入
pGrpAttr	muxGroup 属性指针	A LAND	输入

# 【返回值】

返回值	描述
0	成功
非 0	失败,参见错误码。

# 【注意】

- 1 如果通道未创建,则返回失败。
- 1 此接口并不判断当前是否已经暂停解析,如果已经暂停,调用此接口也返回成功。

# 【举例】



无。

# 8.3.7 AW\_MPI\_MUX\_SetH264SpsPpsInfo

【描述】

设置 muxgroup 的 H264spspps 信息

【语法】

ERRORTYPE AW\_MPI\_MUX\_SetH264SpsPpsInfo(MUX\_GRP muxGrp, VencHeader-Data \*pH264SpsPpsInfo);

### 【参数】

参数名称	描述		A A A A A A A A A A A A A A A A A A A	输入/输出
muxGrp	MUX GROUP 号。耳	X值范围: [0, MUX_MAX	(_GRP_NUM)。	输入
pH264SpsPpsIn	fo《H264 spspps 信息头			输入

# $8.3.8\ AW\_MPI\_MUX\_SetH265SpsPpsInfo$

【描述】

设置 muxgroup 的 spspps 信息

【语法】

ERRORTYPE AW\_MPI\_MUX\_SetH265SpsPpsInfo(MUX\_GRP muxGrp, VencHeader-Data \*pH264SpsPpsInfo);



【参数】

参数名称	描述			输入/输出
muxGrp	MUX GROUP 号。	。取值范围: [0, MUX_MAX	(_GRP_NUM)。	输入
pH265SpsPpsInfo	H265 spspps 信息	急头	-XIV	输入

【返回值】

【注意】

无。

【举例】

无。

8.3.9 AW\_MPI\_MUX\_CreateChn

【描述】

创建 mux 通道,必须指定归属的 mux Group。

【语法】

ERRORTYPE AW\_MPI\_MUX\_CreateChn(MUX\_GRP muxGrp, MUX\_CHN muxChn, MUX\_CHN\_ATTR\_S \*pChnAttr, int nFd);

【参数】

参数名称	描述	输入/输出
muxGrp	MUX GROUP 号。取值范围: [0, MUX_MAX_GRP_NUM)。	 输入
muxChn	MUX chn 号。取值范围:[0, MUX_MAX_CHN_NUM)。	输入
pChnAttr	mux 通道属性指针	输入
nFd	文件句柄	输入

【返回值】



描述 返回值 0 成功 失败,参见错误码。 非 0

【注意】

无。

【举例】

无。

8.3.10 AW\_MPI\_MUX\_DestroyChn

【描述】

销毁 mux 通道,从 muxGroup 中撤出。

【语法】

E Republish the state of the st ERRORTYPE AW\_MPI\_MUX\_DestroyChn(MUX\_GRP muxGrp, MUX\_CHN muxChn);

# 【参数】

参数名称	描述	输入/输出
muxGrp	MUX GROUP 号。取值范围: [0, MUX_MAX_GRP_NUM)。	输入
muxChn	MUX chn 号。取值范围: [0, MUX_MAX_CHN_NUM)。	输入

【返回值】

返回值 描述 0 成功 非 0 失败,参见错误码。

【注意】

无。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

279



# 8.3.11 AW\_MPI\_MUX\_GetChnAttr

# 【描述】

获取 mux 通道属性。

# 【语法】

ERRORTYPE AW\_MPI\_MUX\_GetChnAttr(MUX\_GRP muxGrp, MUX\_CHN muxChn, MUX CHN ATTR S \*pChnAttr);

# 【参数】

参数名称	描述	inderii		inderii		输入/输出
muxGrp	MUX GI	ROUP 号。取值范	围: [0, MUX	_MAX_GRP_N	IUM)。	输入
muxChn	MUX ch	n 号。取值范围:	[0, MUX_M	X_CHN_NUM	1)。	输入
pChnAttr	mux 通道	道属性指针	<b>建</b> 树花	<i>§</i> 1	8	输出

# 【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,参见错误码。

### 【注意】

无。

【举例】

无。

# $8.3.12~AW\_MPI\_MUX\_SetChnAttr$

# 【描述】

设置 mux 通道属性。

### 【语法】

ERRORTYPE AW\_MPI\_MUX\_SetChnAttr(MUX\_GRP muxGrp, MUX\_CHN muxChn, MUX\_CHN\_ATTR\_S \*pChnAttr);

# 【参数】



参数名称	描述		输入/输出
muxGrp	MUX GROUP 号。取值	直范围: [0, MUX_MAX_GRP_NUM)。	输入
muxChn	MUX chn 号。取值范围	🗉: [0, MUX_MAX_CHN_NUM)。	输入
pChnAttr	mux 通道属性指针	AKE THE SHEET	输入

# 【返回值】

描述 返回值 0 成功 非 0 失败,参见错误码。

# 8.3.13 AW\_MPI\_MUX\_SwitchFd 【描述】 7.换(录像)文件句柄 【语法】

ERRORTYPE AW MPI\_MUX\_SwitchFd(MUX\_GRP muxGrp, MUX\_CHN muxChn, int fd, int nFallocateLen);

# 【参数】

参数名称	描述	输入/输出
muxGrp	MUX GROUP 号。取值范围: [0, MUX_MAX_GRP_NUM)。	输入
muxChn	MUX chn 号。取值范围:[0, MUX_MAX_CHN_NUM)。	输入
pChnAttr	mux 通道属性指针	输入
nFallocateLen	文件长度	输入



描述 返回值 0 成功 非 0 失败,参见错误码。

【注意】

无。

【举例】

无。

8.3.14 AW\_MPI\_MUX\_RegisterCallback

【描述】

设置 muxgroup 的回调函数。

【语法】

E REAL HEATH HEAT HEAT THE REAL PROPERTY AND A STATE OF THE PROPERTY AND A STATE OF THE REAL PROPERTY AND A STATE OF THE PROPERTY AND A STATE OF ERRORTYPE AW\_MPI\_MUX\_RegisterCallback(MUX\_GRP muxGrp, MPPCallbackInfo \*pCallback);

# 【参数】

参数名称	描述	Ä	输入/输出
muxGrp	MUX GROUP 号。取值范围:	[0, MUX_MAX_GRP_NUM)。	输入
pCallback	回调参数指针	The last of the la	输入

【返回值】

描述 返回值 成功 0 非 0 失败,参见错误码。

【注意】

无。



# 8.3.15 AW\_MPI\_MUX\_SetMuxCacheDuration

# 【描述】

设置 mux group 的缓冲时间。用于缓冲音视频编解码数据。

# 【语法】

ERRORTYPE AW MPI MUX SetMuxCacheDuration(MUX GRP muxGrp, int nCacheMs);

# 【参数】

参数名称	描述		输入/输出
muxGrp nCacheMs	MUX GROUP 号。取值范围: 缓冲时间,单位 ms	[0, MUX_MAX_GRP_NUM)。	输入输入
ncachems	坂州町町,半世 IDS	W.LV	和八人

# 【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,参见错误码。

【注意】

无。

【举例】

无。

# 8.3.16 AW\_MPI\_MUX\_SetSwitchFileDurationPolicy

# 【描述】

设置文件切换策略。

### 【语法】

ERRORTYPE AW\_MPI\_MUX\_SetSwitchFileDurationPolicy(MUX\_GRP muxGrp, MUX\_CHN muxChn, RecordFileDurationPolicy ePolicy)

【参数】



参数名称	描述		输入/输出
muxGrp	MUX GROUP 号。	取值范围: [0, MUX_MAX_GRP_NUM)。	输入
ePolicy	文件切换策略		输入

【返回值】

返回值	描述	
0 非 0	成功	参见错误码。
∃F U	大火火,	<b>梦</b> 见垣 庆吟。

【注意】

【举例】

无。

# Carthelite Har a state of the s $8.3.17\ AW\_MPI\_MUX\_GetSwitchFileDurationPolicy$

【描述】

获取文件切换策略。

【语法】

AW\_MPI\_MUX\_GetSwitchFileDurationPolicy(MUX\_GRP ERRORTYPE muxGrp, MUX\_CHN muxChn, RecordFileDurationPolicy \*pPolicy)

【参数】

参数名称	描述	**************************************	输入/输出
muxGrp	MUX GROUP 号。	取值范围: [0, MUX_MAX_GRP_NUM	()。 输入
pPolicy	文件切换策略		输出

【返回值】

返回值	描述	
0 Blinds	成功	
《罪 0	失败,	参见错误码。



【注意】

【举例】

无。

# 8.4 数据类型

# 8.4.1 MUX\_GRP\_ATTR\_S With the state of the state of

【说明】

mux group 属性数据结构体

【定义】

typedef struct MUX\_GRP\_ATTR\_S

{

// video

int mHeight;

int mWidth;

int mVideoFrmRate; // \*1000

int mCreateTime;

int mMaxKeyInterval;

PAYLOAD TYPE mVideoEncodeType; //VENC CODEC H264

int mRotateDegree; //0, 90, 180, 270

// audio

int mChannels;

int mBitsPerSample;

int mSamplesPerFrame; //sample\_cnt\_per\_frame

int mSampleRate;

PAYLOAD\_TYPE\_E mAudioEncodeType; //AUDIO\_ENCODER\_AAC\_TYPE



# }MUX\_GRP\_ATTR\_S;

# 【成员】

成员名称	描述	THE THE PERSON NAMED IN COLUMN TWO IS NOT THE PERSON NAMED IN COLUMN TO THE PERSON NAMED IN COLU	A	XX.	
mHeight	视频高度				
mWidth	视频宽度				
mVideoFrmRate	视频帧率				
mCreateTime	创建时间				
mMaxKeyInterval	关键帧间隔	嗝			
mVideo Encode Type	视频编码	类型			
mRotateDegree	旋转角转,	,取值范围:[0 <i>,</i>	90, 180,270]		•.
mChannels	音频通路	数,取值范围:[	1,2] <sub>iii</sub> de <sup>ni</sup>		inder
mBitsPerSample	音频采样》	深度,取值范围:	[16]		771.
mSamplesPerFrame	1.	采样数,取值范围	νX1. <sup>1</sup>	A TOP OF THE PERSON NAMED IN COLUMN TO THE PERSON NAMED IN COLUMN	
mSampleRate			3000, 16000, 44	100, 48000。	
mAudioEncodeType	音频编码	类型			
茶梅		***************************************	16	KERT	
意事项】		4	M		
关数据类型及接口】	1	The.			
	le ni	./			787

# 【注意事项】

无。

【相关数据类型及接口】

无。

# 8.4.2 MUX\_CHN\_ATTR\_S

# 【说明】

muxChannel 属性参数结构体。

# 【定义】

typedef struct MUX\_CHN\_ATTR\_S

{

int mMuxerId;

MEDIA FILE FORMAT E mMediaFileFormat;

int64 tmMaxFileDuration; //unit:ms

int64\_t mMaxFileSizeBytes; //unit:byte



int mFallocateLen;

BOOL mCallbackOutFlag; //send data through callback.

FSWRITEMODE mFsWriteMode;

int mSimpleCacheSize;

}MUX\_CHN\_ATTR\_S;

# 【成员】

成员名称	描述
mMuxerId	muxer 实例的 id 号,由调用者指定,同一 muxerGroup 下的 muxerChannel 的 mi
mMediaFileFormat	文件封装类型
mMaxFileDuration	文件最大持续时间(ms)
mMaxFileSizeBytes	文件最大长度
mFallocateLen	文件预分配长度
mCallbackOutFlag	是否回调输出(用于网络传输),如果为 TRUE,则不写文件系统,通过 callback 输出
mFsWriteMode	写卡操作方式类型,只支持 FSWRITEMODE_SIMPLECACHE 和 FSWRITEMODE
mSimple Cache Size	simpleCahce 写卡模式下的 cache 大小。
【注意事项】	
本地写卡时支持 2 种写-	卡方式
【相关数据类型及接口】	

# 【注意事项】

无。

# 8.4.3 CdxFdT

【说明】

CdxFdT 参数结构体。

【定义】

typedef struct CdxFdT

{

int mFd;

int mIsImpact; int mnFallocateLen;



int mMuxerId;

CdxFdT;

【成员】

成员名称	描述
mFd	文件句柄
mnFallocateLen	文件预分配长度,单位:字节。
mIsImpact	是否是碰撞文件
mMuxerId	muxid

【相关数据类型及接口

# 8.4.4 RecordFileDurationPolicy

【说明】

录制文件的切换策略的枚举类型。

【定义】

typedef enum {

RecordFileDurationPolicy MinDuration = 0,

RecordFileDurationPolicy AverageDuration,

RecordFileDurationPolicy AccurateDuration,

} RecordFileDurationPolicy;

### 【成员】

成员名称 描述

RecordFileDurationPolicy MinDur最io时长策略,录制时间必须要大于等于用户指定的时

RecordFileDurationPolicy Average 平均时长策略,录制的文件的平均时长等于用户指定的 时长,单个文件时长可以大于或小于用户指定时长。

RecordFileDurationPolicy Accurat精确贴长筛略,当前文件达到用户指定的时长后,立即

Will Think the think the transfer of the state of the sta



【注意事项】

无。

### The State of t

【相关数据类型及接回】

# 8.5 错误码

错误代码	宏定义	描述
0xA0658002	ERR_MUX_INVALID_CHNID	通道 ID 超出合法范围
0xA0658003	ERR_MUX_ILLEGAL_PARAM	参数超出合法范围
0xA0658004	ERR_MUX_EXIST	试图申请或者创建已经存在的设备、通道或者资源
0xA0658006	ERR_MUX_NULL_PTR	函数参数中有空指针
0xA0658007	ERR_MUX_NOT_CONFIG	使用前未配置
0xA0658008	ERR_MUX_NOT_SUPPORT	不支持的参数或者功能
0xA0658009	ERR_MUX_NOT_PERM	该操作不允许,如试图修改静态配置参数
0xA0658005	ERR_MUX_UNEXIST	试图使用或者销毁不存在的设备、通道或者资源
0xA065800C	ERR_MUX_NOMEM	分配内存失败,如系统内存不足
0xA065800D	ERR_MUX_NOBUF	分配缓存失败,如申请的数据缓冲区太大
0xA0658010	ERR_MUX_SYS_NOTREADY	系统没有初始化或没有加载相应模块
0xA0658012	ERR_MUX_BUSY	MUX 系统忙

版权所有 © 珠海全志科技股份有限公司。保留一切权利



9 DEMUX 模块

# 9.1 概述

demux 模块,即文件解封装模块。本模块支持创建多个 demux 通道,每路通道的解封装过程独立。

# 9.2 功能描述与使用

demux 通道内部线程完成读媒体文件,分包功能。

demux 模块输入、输出端口均支持绑定与绑定非方式。输入输出端口连接支持情况如下图所示。输入端口最多支持 1 路 clock 输入绑定,输出端口最多支持 2 路输出绑定(vdec/adec)。

demux 通道支持只解析视频包或者只解析音频包; 支持 seek 解析。

注意事项: 输出绑定方式时,demux 模块本身不分配存放解析包所需要的内存,相应内存由 demux

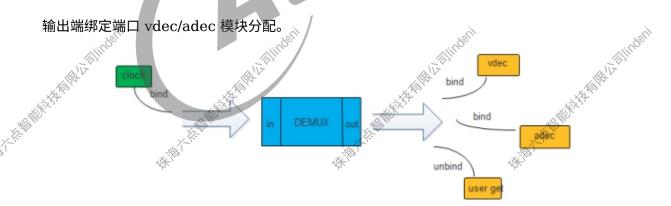


图 9-1: img



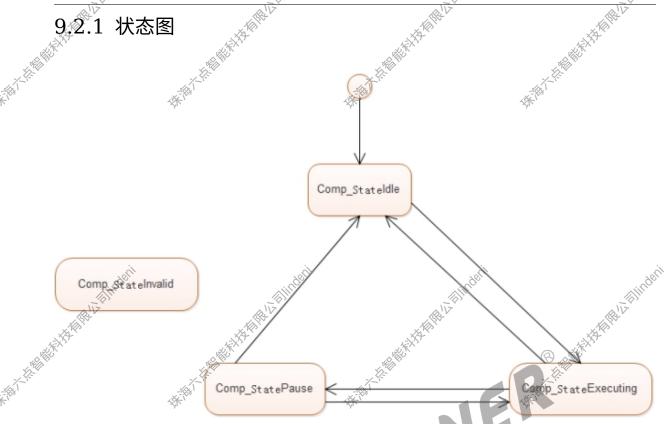


图 9-2: img

demux 组件内部状态设定为:

1 COMP StateLoaded: 组件初始创建状态。

1 COMP\_StateIdle:组件完成初始化,参数设置、资源配置完毕,随时可以运行的状态。

1 COMP StateExecuting: 运行状态。

I COMP StatePause: 暂停状态。

l COMP StateInvalid: 异常状态。

AW\_MPI\_DEMUX\_CreateChn() 的实现过程会经过 COMP\_StateLoaded 状态,到达 COMP\_StateIdle。

组件内部状态转换的函数是: SendCommand(..., COMP\_CommandStateSet, 目标COMP\_State, ...);

# 9.2.2 API 和状态

能够引起状态变化的 API, 见状态转换图。

每个 API 只能在允许的状态下调用,如果不在允许的状态下调用 API,则无效。



API 列表如下: (允许被调用的状态栏填写 Y)

.5c-X		~~X,			
ALC N		COMP_StateIdle	COMP_	Pause COMI	P_StateExecuting
AW	/_MPI_DEMUX_CreateChn	THE PARTY OF THE P	A)	- 12 T	
AW	/_MPI_DEMUX_DestroyChn	Y	Y	Y	3/1
AW	/_MPI_DEMUX_RegisterCallback				
AW	V_MPI_DEMUX_SetChnAttr	Y		Y	
AW	V_MPI_DEMUX_GetChnAttr	Y		Y	
AW	$V_{ m MPI\_DEMUX\_GetMediaInfo}$	Y	Y	Y	
AW	V_MPI_DEMUX_Start	Y	Y		
AW	V_MPI_DEMUX_Stop		Y	Y	
AW	V_MPI_DEMUX_Pause		ino.	Y	inci
AW	V_MPI_DEMUX_ResetChn	Y	illigo		Minde
AW	MPI_DEMUX_Seek	WIV.	Y	Y	7
ΑW	MPI_DEMUX_getDmxOutPutBuf			Y	
ÂW	V_MPI_DEMUX_releaseDmxBuf			(Constitution of the constitution of the const	
AW	V_MPI_DEMUX_DisableTrack	Y			The state of the s
AW	V_MPI_DEMUX_DisableMediaTrack	Y			-183, '
9.	.3 API 参考	MIN	16.		

# 9.3 API 参考

demux 模块主要提供 demux 通道 (在本文档中通道等同于组件实例) 的创建和销毁、通道的复 位、开启和停止等功能。

# 9.3.1 AW\_MPI\_DEMUX\_CreateChn

# 【描述】

创建 demux 通道

# 【语法】

ERRORTYPE AW\_MPI\_DEMUX\_CreateChn(PARAM\_IN DEMUX\_CHN dmxChn, PARAM\_IN const DEMUX\_CHN\_ATTR\_S \*pstAttr);

# 【参数】

参数名称	描述	in	输入/输出
dmxChn	demux 通道号。取值范围:	[0, DEMUX_MAX_CHN_NUM)。	输入 🧥
pstAttr	demux 通道属性指针	W. V.	输入





【返回值】

(s.		Y=X,2,
	返回值	描述
	0	成功
	非 0	失败,参见错误码。

【需求】

l 头文件: mm\_comm\_demux.h、mm\_common.h

l 库文件: libmedia mpp.so

# 

ERRORTYPE AW\_MPI\_DEMUX\_DestroyChn(PARAM\_IN DEMUX\_CHN dmxChn);

【参数】

参数名称	描述				输入/输出
dmxChn	demux 通道号。	取值范围:	[0, DEMUX_	MAX_CHN_NUM)。	输入

【返回值】

返回值	描述	
0	成功	
非 0 .	失败,	参见错误码。



1头文件: mm\_comm\_demux.h、mm\_common.h

l 库文件: libmedia\_mpp.so

【注意】

无。

【举例】

无。

# 9.3.3 AW\_MPI\_DEMUX\_RegisterCallback

【描述】

设置 demux 通道回调。

【语法】

ERRORTYPE AW\_MPI\_DEMUX\_RegisterCallback(PARAM\_IN DEMUX\_CHN dmxChn, PARAM\_IN MPPCallbackInfo \*pCallback);

# 【参数】

参数名称	描述		输入/输出
dmxChn	demux 通道号。取值范围:	[0, DEMUX_MAX_CHN_NUM)。	输入
pCallback	回调参数指针	in the second second	输入

# 【返回值】

返回值	描述
0	成功。
非 0	失败,参见错误码。

# 【需求】

l 头文件: mm comm demux.h、mm common.h

l 库文件: libmedia\_mpp.so

【注意】

无。



【举例】

※ 无。

# 9.3.4 AW\_MPI\_DEMUX\_SetChnAttr

【描述】

设置 demux 通道属性。

【语法】

ERRORTYPE AW\_MPI\_DEMUX\_SetChnAttr(PARAM\_IN\_DEMUX\_CHN\_dmxChn, PARAM\_IN\_DEMUX\_CHN\_ATTR\_S \*pAttr);

# 【参数】

参数名称	描述	12.5	输入/输出
dmxChn	demux 通道号。	取值范围: [0, DEMUX_MAX_CHN_	NUM)。 输入
pAttr	属性批针		输入

【返回值】

返回值	描述	
0	成功	
非0%	失败,	参见错误码。

【需求】

1头文件: mm\_comm\_demux.h、mm\_common.h

l 库文件: libmedia\_mpp.so

【注意】

无。

【举例】

无。



# 9.3.5 AW\_MPI\_DEMUX\_GetChnAttr

# 【描述】

获取 demux 通道属性。

# 【语法】

ERRORTYPE AW\_MPI\_DEMUX\_GetChnAttr(PARAM\_IN DEMUX\_CHN dmxChn, PARAM OUT DEMUX CHN ATTR S \*pstAttr);

# 【参数】

参数名称	描述	::ndeni		inberi	输入/输出
dmxChn	demux 通道号。	取值范围	: [0, DEMUX_N	MAX_CHN_NUM)。	输入
pstAttr	属性指针	No.	×4 <sup>1</sup>		输出
【返回值】	A THE LAND OF THE PARTY.		W. A. The House of the Control of th		
		返回值	描述	MG.	
		0 非 0	成功 失败,参见错误6	<u> </u>	

# 【返回值】

返回值	描述	
0	成功	
非 0	失败,参	》见错误码。

# 【需求】

l 头文件: mm\_comm\_demux.h、mm\_common.h

l 库文件: libmedia\_mpp.so

【注意】

无。

【举例】

无。

# 9.3.6 AW\_MPI\_DEMUX\_GetMediaInfo

# 【描述】

获取 demux 通道的媒体文件

【语法】



ERRORTYPE AW\_MPI\_DEMUX\_GetMediaInfo(PARAM\_IN DEMUX\_CHN dmxChn, PARAM\_OUT DEMUX\_MEDIA\_INFO\_S \*pMediaInfo);

### 【参数】

参数名称	描述		N.		7)	输入/输出
dmxChn pMediaInfo	demux 通道号。 媒体信息指针	取值范围:	[0, DEMUX_	MAX_CHN	_NUM)。	· 输入 输出

# 【返回值】

返回值	描述	Beril	
0. William	成功	Alline Land	
0 非 0	失败,参见错误码。	/ XAMELY	
	描述 成功 失败,参见错误码。	NER	
	/		

# 【需求】

l 头文件: mm\_comm\_demux.h、mm\_common.h

l 库文件: libmedia\_mpp.so

【注意】

无。

【举例】

无。

9.3.7 AW\_MPI\_DEMUX\_Start

【描述】

开启 demux 通道,解析文件。

【语法】

ERRORTYPE AW MPI DEMUX Start(PARAM IN DEMUX CHN dmxChn);

# 【参数】

参数名称	描述	allindent	输入/输出
dmxChn	demux 通道号。取值范围:	: [0, DEMUX_MAX_CHN	_NUM)。 输入。



# 【返回值】

L-X-		4-X
O.K.	返回值	描述
	0 非 0	成功 失败,参见错误码。

【需求】

l 头文件: mm comm demux.h、mm common.h

l 库文件: libmedia\_mpp.so

【注意】

1 如果通道未创建,则返回失败。

了此接口并不判断当前是否已经开始解析,如果已经开始解析,

# 9.3.8 AW\_MPI\_DEMUX\_Stop 【描述】

停止 demux 通道解析文件。

【语法】

ERRORTYPE AW MPI\_DEMUX\_Stop(PARAM\_IN DEMUX\_CHN dmxChn);

### 【参数】

参数名称	描述		ŚŁ,		%,	输入/输出
dmxChn	demux 通道号。	取值范围:	[0, DEMUX]	_MAX_CHN_NU	JM)。	输入

# 【返回值】

返回值	描述	
0 6	成功	
15000	失败,	参见错误码。
A 117		- 11

版权所有 © 珠海全志科技股份有限公司。保留一切权利



# 【需求】

1头文件: mm comm demux.h、mm common.h

l 库文件: libmedia mpp.so

### 【注意】

1 如果通道未创建,则返回失败。

1 此接口并不判断当前是否已经停止解析,如果已经停止,调用此接口也返回成功。

1 此接口用于 demux 通道停止解析文件,在解码通道销毁或复位前必须停止解析文件。

### 【举例】

# 9.3.9 AW\_MPI\_DEMUX\_Pause

### 【描述】

暂停 demux 通道解析文件。

### 【语法】

A. I. ERRORTYPE AW MPI DEMUX Pause(PARAM IN DEMUX CHN dmxChn);

### 【参数】

\O;\		\0;\	\Q\\\	
参数名称	描述	Allino	allino de la companya della companya della companya de la companya de la companya della companya	输入/输出
dmxChn	demux 通道号	。取值范围:	[0, DEMUX_MAX_CHN_NUM)。	输入

# 【返回值】

返回值	描述	
0	成功	
非 0	失败,参	<b>》见错误码</b> 。

# 【需求】

l 头文件: mm comm demux.h、mm common.h

l 库文件: libmedia mpp.so 注意】



1 如果通道未创建,则返回失败。

l 此接口并不判断当前是否已经暂停解析,如果已经暂停,调用此接口也返回成功。

### 【举例】

无。

# 9.3.10 AW\_MPI\_DEMUX\_ResetChn

# 【描述】

复位 demux 通道,但不会重置已设置的通道参数。通道复位后,缓冲数据都被清空,随时可以从文件开始处重新解析数据。

# 【语法】

ERRORTYPE AW\_MPI\_DEMUX\_ResetChn(PARAM\_IN DEMUX\_CHN dmxChn);

# 【参数】

参数名称	描述	1111	输入/输出
dmxChn	demux 通道号。取值范围:	[0, DEMUX_MAX_CHN_NUM)。	输入

# 【返回值】

返回值	描述	
0.4	成功	117
非 0	失败,	参见错误码。

# 【需求】

l 头文件: mm comm demux.h、mm common.h

l 库文件: libmedia mpp.so

# 【注意】

l Reset 并不存在的通道,返回失败 AW ERR VENC UNEXIST。

l 如果通道没有停止而 reset 通道,则返回失败。

### 【举例】

**太**。



# 9.3.11 AW\_MPI\_DEMUX\_Seek

# 【描述】

跳转到媒体文件某个时间点解析

# 【语法】

ERRORTYPE AW MPI DEMUX Seek(PARAM IN DEMUX CHN dmxChn, PARAM IN int msec);

# 【参数】

描述	inderil		inderi	输入/输出
demux 通道号。	取值范围	: [0, ]	DEMUX_MAX_CHN_NUM)。	输入 输入
时间点(ms)	No.		A TOTAL CONTRACTOR OF THE PARTY	输入
H. H		×,		
	返回值	描述	ANG.	
	0	成功	<b>会</b> 用供识和	
	demux 通道号。	demux 通道号。取值范围时间点(ms)	demux 通道号。取值范围: [0, 时间点 (ms) 返回值 描述 0 成功	demux 通道号。取值范围: [0, DEMUX_MAX_CHN_NUM)。时间点 (ms) 返回值 描述 0 成功

# 【返回值】

返回值	描述	
0	成功	
非 0	失败,参见错误码。	

# 【需求】

l 头文件: mm\_comm\_demux.h、mm\_common.h

l 库文件: libmedia\_mpp.so

【注意】

会清空已解析 buffer list

【举例】

无。

# 9.3.12 AW\_MPI\_DEMUX\_getDmxOutPutBuf

# 【描述】

获取 demux 解析出来的数据包

【語法】



ERRORTYPE AW\_MPI\_DEMUX\_getDmxOutPutBuf(PARAM\_IN DEMUX\_CHN dmx-Chn, PARAM\_OUT DemuxCompOutputBuffer \*pDmxOutBuf, PARAM\_IN int nMilliSec);

### 【参数】

参数名称 描述
dmxChn demux 通道号。取值范围: [0, DEMUX\_MAX\_CHN\_NUM)。
pDmxOutBuf 数据包指针
nMilliSec 获取数据的超时时间。-1 表示阻塞模式; 0 表示非阻塞模式; >0 表示阻塞 s32MilliSec 毫秒,

### 【返回值】

### 【需求】

l 头文件: mm\_comm\_demux.h、mm\_common.h

l 库文件: libmedia\_mpp.so

【注意】

仅适用于非绑定模式

【举例】

无。

# 9.3.13 AW\_MPI\_DEMUX\_releaseDmxBuf

# 【描述】

释放 demux 解析出来的数据包 buffer

### 【语法】

ERRORTYPE AW\_MPI\_DEMUX\_releaseDmxBuf(PARAM\_IN DEMUX\_CHN dmxChn, PARAM\_QUT DemuxCompOutputBuffer \*pDmxOutBuf);

### 【参数】



输入

ALLWIMER		The state of the s		文档密级、秘密
2/12		117	2/12	4/17
参数名称	描述		NA TOP OF THE PARTY OF THE PART	输入/输出
dmxChn	demux 通道号	。取值范围:[(	D, DEMUX_MAX_	_CHN_NUM)。 输入

【返回值】

pDmxOutBuf

返回值 描述 0 成功 非 0 失败,参见错误码。

【需求】

1头文件: mm comm demux h、mm common.h

数据包指针

l 库文件: libmedia mpp.so

【注意】

仅适用于非绑定模式,与 AW\_MPI\_DEMUX\_getDmxOutPutBuf 成对使用

【举例】

无。

# 9.4 数据类型

# 9.4.1 STREAMTYPE I

【说明】

流媒体类型

【定义】

typedef enum STREAMTYPE E{

STREAMTYPE\_NETWORK,

STREAMTYPE LOCALFILE,

}STREAMTYPE\_E;

【成员】



成员名称《

STREAMTYPE\_NETWORK 网络 stream。 STREAMTYPE LOCALFILE 本地文件 stream。

描述

【注意事项】

无。

【相关数据类型及接口】

无。

9.4.2 SOURCETYPE E

【说明】

源类型

【定义】

typedef enum SOURCETYPE E{

SOURCETYPE FD,

SOURCETYPE FILEPATH,

SOURCETYPE\_WRITER\_CALLBACK = 6, //for recoder writer

}SOURCETYPE E;

【成员】

成员名称		描述	
SOURCETYPE_FD	茶	文件来源是 fd。	**************************************
SOURCETYPE_FILEPATH		文件来源是字符	串路径。
SOURCETYPE WRITER CAI	LLBACK	不支持。	

【注意事项】

无。

【相关数据类型及接口】

版权所有 © 珠海全志科技股份有限公司。保留一切权利



# 9.4.3 CEDARX\_MEDIA\_TYPE

【说明】

媒体类别

【定义】

typedef enum CEDARX MEDIA TYPE{

CEDARX MEDIATYPE NORMAL = 0,

CEDARX MEDIATYPE RAWMUSIC,

CEDARX\_MEDIATYPE\_3D\_VIDEO,

CEDARX\_MEDIATYPE\_BD,

CEDARX\_SOURCE\_MULTI\_LIPT

CEDARY

}CEDARX\_MEDIA\_TYPE;

【成员】

成员名称 描述

CEDARX MEDIATYPE NORMAL

CEDARX MEDIATYPE RAWMUSIC

CEDARX MEDIATYPE 3D VIDEO /-

CEDARX MEDIATYPE DRM VIDEO

CEDARX MEDIATYPE DRM WVM VIDEO

CEDARX MEDIATYPE DRM ES BASED VIDEO

CEDARX MEDIATYPE DRM CONTAINNER BASED VIDEO

CEDARX MEDIATYPE BD

CEDARX SOURCE MULTI URL

【相关数据类型及接口】



无。

# 9.4.4 DEMUX\_DISABLE\_TRACKINFO

【说明】

轨道类型

【定义】

typedef enum DEMUX DISABLE TRACKINFO {

DEMUX DISABLE AUDIO TRACK = 0x01,

 $DEMUX_DISABLE_VIDEO_TRACK = 0x02$ ,

DEMUX DISABLE SUBTITLE TRACK = 0x04

} DEMUX\_DISABLE\_TRACKINFO;

【成员】

成员名称

描述

DEMUX\_DISABLE\_AUDIO\_TRACK
DEMUX\_DISABLE\_VIDEO\_TRACK
DEMUX\_DISABLE\_SUBTITLE\_TRACK

音频轨道 视频轨道

视频轨道 subtitle

【注意事项】

【相关数据类型及接口】

无。

# 9.4.5 DEMUX\_CHN\_ATTR\_S

【说明】

通道属性【定义】

typedef struct DEMUX CHN ATTR S

版权所有 © 珠海全志科技股份有限公司。保留一切权利

306



```
STREAMTYPE E mStreamType;
```

SOURCETYPE\_E mSourceType;

char\* mSourceUrl;

int mFd;

int mDemuxDisableTrack; //DEMUX DISABLE AUDIO TRACK

}DEMUX CHN ATTR S;

# 【成员】

成员名称	描述	, defi	aderii
mStreamType;	stream 类型	A TOP IN THE PERSON NAMED	
mSourceType	文件来源类别	rikila.	
mSourceUrl	url 地址		NA THE STATE OF TH
mFd	文件句柄		
mDemuxDisableTrack	禁止解析的轨道	道类型,DEMUX_DISABLE_TRACKI	NFO 枚举类型的组合。
【注意事项】		ME	- Tyl
无。		1 Ma	
【相关数据类型及接口】	AL		
ль <sub>о</sub>			•

# 【注意事项】

# 9.4.6 DEMUX\_VIDEO\_STREAM\_INFO\_S

# 【说明】

通道属性【定义】∜

typedef struct DEMUX\_VIDEO\_STREAM\_INFO\_S

{

PAYLOAD TYPE E mCodecType;

int mWidth; //display width

int mHeight;

int mFrameRate; // x1000



int mAvgBitsRate;

int mMaxBitsRate;

int nCodecSpecificDataLen;

char\* pCodecSpecificData;

}DEMUX\_VIDEO\_STREAM\_INFO\_S;

### 【成员】

成员名称	描述			
mCodecType	视频编码类型			
mCodecType mWidth mHeight	视频编码类型视频图像宽度			
mHeight	视频图像高度			
mFrameRate	帧率,单位 x1000。			
mAvgBitsRate	平均码率(			
mMaxBitsRate	最大码率			
nCodecSpecificDataLen	Metadata size			
pCodecSpecificData	Buffer addr that store metadata			
· 工				

### 【注意事项】

无。

【相关数据类型及接口】

无。

# 9.4.7 DEMUX\_AUDIO\_STREAM\_INFO\_S

### 【说明】

通道属性【定义】

typedef struct DEMUX\_AUDIO\_STREAM\_INFO\_S

{

PAYLOAD\_TYPE\_E mCodecType;

int mChannelNum;

int mBitsPerSample;

int mSampleRate;



unsigned char strLang[MAX\_LANG\_CHAR\_SIZE];

}DEMUX\_AUDIO\_STREAM\_INFO\_S;

【成员】

成员名称	 描述
mCodecType	音频编码类型
mChannelNum	音轨的声道数
mBitsPerSample	sample 采样位数
mSampleRate	采样率
strLang	音轨名称

# 9.4.8 DEMUX\_MEDIA\_INFO\_S 【说明】 @道属性【定义】 /pedef str

typedef struct DEMUX\_MEDIA INFO\_S

unsigned int mDuration; //unit: ms

int mAudioNum, mAudioIndex;

int mVideoNum, mVideoIndex;

int mSubtitleNum, mSubtitleIndex;

DEMUX AUDIO STREAM INFO S mAudioStreamInfo[DEMUX MAX AUDIO STREAM NUM];

 $DEMUX\_VIDEO\_STREAM\_INFO\_S\ mVideoStreamInfo[DEMUX\_MAX\_VIDEO\_STREAM\_NUM];$ 

DEMUX\_SUBTITLE\_STREAM\_INFO\_S mSubtitleStreamInfo[DEMUX\_MAX\_SUBTITLE\_STREAM\_N

}DEMUX MEDIA INFO



-\	3,1	-13	(-)/3/	-13
K.	成员名称	描述	H. H	
	mDuration	文件时长。	单位 ms。	
	mAudioNum	文件包含的	音频轨道数量。	
	mAudioIndex	选定播放的	音频轨道 index,范围 [(	), mAudioNum)。
	mVideoNum	文件包含的	视频轨道数量。	
	mVideoIndex	选定播放的	视频轨道 index,范围 [(	), mVideoNum)。
	mSubtitleNum	文件包含的	字幕轨道数量,不支持。	
	mSubtitleIndex	选定播放的	字幕轨道 index,范围 [(	), mSubtitleNum),不支持。
	mAudioStreamInfo	audioTrac	k 的流信息数组	
	mVideoStreamInfo	videoTrac	k 的流信息数组	i)
	mSubtitleStreamInfo	subtitleTr	ack 的流信息数组,未使	用。 <sup>de</sup>
×	【注意事项】	A THE WAY	A STATE OF THE PARTY OF THE PAR	O LAND
J	Ē.		A CHILD	
	【相关数据类型及接口】			
J	Ē.		MIN	
S	9.4.9 DemuxCon	npOutp	utBuffer	

### 【注意事项】

# 9.4.9 DemuxCompOutputBuffer

### 【说明】

Demux 组件的输出 buffer 数据结构

### 【定义】

{

typedef struct DemuxCompOutputBuffer

int nTobeFillLen; // [in] indicate data size.

unsigned int video\_stream\_type; //[in] indicate VIDEO\_TYPE\_MAJOR, CDX\_VIDEO\_STREAM\_MAJO

unsigned char\* pBuffer; // [out]

unsigned char\* pBufferExtra; // [out]

unsigned int nBufferLen; // [out] size of the buffer , in bytes

unsigned int nBufferExtraLen; // [out] size of the buffer extra, in bytes



unsigned int nCtrlBits; // [in]

unsigned int nFilledLen; // [in]

int64\_t nTimeStamp; // [in]

int64 t duration; // [in] for subtitle

//unsigned int subDispEndTime;

int infoVersion; // [in]

void \*pChangedStreamsInfo; // [in] VideoInfo/AudioInfo/SubtitleInfo

int media type; //add by andy for video/audio/subtitle

}DemuxCompOutputBuffer;

[in][out] 相对于绑定模式下进行隧道链接的 vdec 而言。如果 demux 组件处于非绑定模式,无意义。

### 【成员】

成员名称	描述
nTobeFillLen	申请的 buffer 长度。
video_stream_type	标记码流编号,针对 3D 双流格式。取值:
	VIDEO_TYPE_MAJOR。
pBuffer	装填待解码数据的第一段 buffer 起始虚拟地址。
pBufferExtra	装填待解码数据的第二段 buffer 起始虚拟地址。
nBufferLen	第一段 buffer 的长度。
nBufferLenExtraLen	第二段 buffer 的长度
nCtrlBits	第一段 buffer 的长度。 第二段 buffer 的长度 指示 buffer 数据的特征,为以下宏的组合。#define
XXXXX	CEDARV_FLAG_PTS_VALID 0x2#define
· ·	CEDARV_FLAG_FIRST_PART 0x8#define
	CEDARV_FLAG_LAST_PART
AK.	0x10CEDARV_FLAG_PTS_VALID:是否携带有效的 PTS。
	CEDARV_FLAG_FIRST_PART: buffer 数据是否包含一帧的起
	始。CEDARV_FLAG_LAST_PART:buffer 数据是否包含一帧的
	结束。
nFilledLen	buffer 中有效数据的长度。
nTimeStamp	时间戳,单位微秒。
duration	仅用于表示 subtitle 字幕的持续时间,单位微秒。
infoVersion	码流信息版本,取值范围: $[0, +\infty)$ ,当码流改变时,版本号加 $1$ 。
pChangedStreamsIn	fo改变后的码流信息。
media type	码流类型,取值范围:enum CdxMediaTypeE。



【注意事项】

无。

【相关数据类型及接回】

无。

# 9.5 错误码

	错误代码	宏定义	
	0xA0648002	ERR_DEMUX_INVALID_CHNID	通道 ID 超出合法范围
	0xA0648003	ERR_DEMUX_ILLEGAL_PARAM	参数超出合法范围
	0xA0648004	ERR_DEMUX_EXIST	试图申请或者创建已经存在的设备、通道或者资源
HE NEW	0xA0648006	ERR_DEMUX_NULL_PTR	函数参数中有空指针
	0xA0648007	ERR_DEMUX_NOT_CONFIG	使用前未配置
XX.	0xA0648008	ERR_DEMUX_NOT_SUPPORT	不支持的参数或者功能
	0xA0648009	ERR_DEMUX_NOT_PERM	该操作不允许,如试图修改静态配置参数
	0xA0648005	ERR_DEMUX_UNEXIST	试图使用或者销毁不存在的设备、通道或者资源
	0xA064800C	ERR_DEMUX_NOMEM	分配内存失败,如系统内存不足
	0xA064800D	ERR_DEMUX_NOBUF	分配缓存失败,如申请的数据缓冲区太大
	0xA0648010	ERR_DEMUX_SYS_NOTREADY	系统没有初始化或没有加载相应模块
	0xA0648012	ERR_DEMUX_BUSY	VENC 系统忙
	A THE WALL WINDOWS TO SERVE THE SERV	The state of the s	Her Her Lillingtoni
W. T. C.			

版权所有 ⑥ 珠海全志科技股份有限公司。保留一切权利



# 10 音频

# 10.1 概述

### Ø \* 音频输入输出 \*

音频输入输出接口简称为 AIO(Audio Input/Output)接口,用于向下对接 alsa driver 和 Audio Codec,向上提供 api 对接应用程序,完成声音的录制和播放。AIO 接口分为两种类型:输入模式、输出模式。当为输入类型时,称为 AIP,当为输出类型时,称为 AOP。软件中负责抽象音频接口输入功能的单元,称之为 AI 设备;负责抽象输出功能的单元,称之为 AO 设备。一个 AI 设备下可以挂多个 AI 通道 (组件),实现音频数据复用。

AIO 设备向下通过 alsa-lib 与 alsa 内核驱动对接,来进行 pcm 数据交互、播放/采集控制、音量大小/静音状态调整等操作。AI 设备支持板卡 mic 的声音采集,AO 设备支持 lineout 和 hdmi 两种方式输出。

AIO 设备向上对接 AIO 通道,AIO 通道通过各自的输入/输出端接口对接 app。AI、AO 通道为单端口通道,AI 通道提供 outport 用于向 app 输出采集的 pcm 数据,AO 通道提供 input 通道用于 app 向其送 pcm 数据进行音频播放。例如,对于 AI 组件,当 pcm 和 mixer 控件初始化完毕并且开始采集 pcm 数据后,app 可以从 AI 组件 outport 拿 pcm 数据,其 inport 在内部实现固定绑定到音频采集的环型 buffer;对于 AO 组件,app 可以向其 inport 不停地送 pcm数据,其 outport 内部实现了固定绑定到音频输出设备的环形 buffer。

### Ø \* 音频编码解码 \*

音频编解码组件简称为 AEnc、ADec,用于压缩/解压缩 pcm 数据。App 可创建多个 AEnc、ADec 通道,各通道可设置不同编解码格式等参数,并且各通道独立运行,互不干扰。

AEnc 编码通道为双端口通道,其 inport 接收外部送来的 pcm 数据,而不关心数据来自哪个模块,例如可来自 AI 通道,也可来自本地磁盘文件; 其 outport 用于输出编码码流,可送到 muxer 模块、ADec 模块或本地磁盘文件。

ADec 解码通道也为双端口通道,其 inport 接收压缩的音频码流,也不关心数据来自哪个模块,例如可来自 demuxer 模块,也可来自网络,也可来自 AEnc 模块;其 outport 用于输出 pcm数据,可送到 AO 通道、AI 通道或本地磁盘文件。

从 AI 组件获取了 pcm 数据后,通常送编码器进行数据压缩,AEnc 组件即为音频编码的设备抽象。通过编码器类型和码率的设置,可以进行多种格式的音频压缩,目前包括以下格式:aac、mp3、adpcm、pcm、g711a、g711u、g726。

AEnc 组件为双端口 (inport、outport) 组件,即 app 可以操作输入端口和输出端口的数据,如



向 AEnc 输入端口送 pcm 数据,编码完成后,app 再从 AEnc 输出端口取压缩后的音频码流数

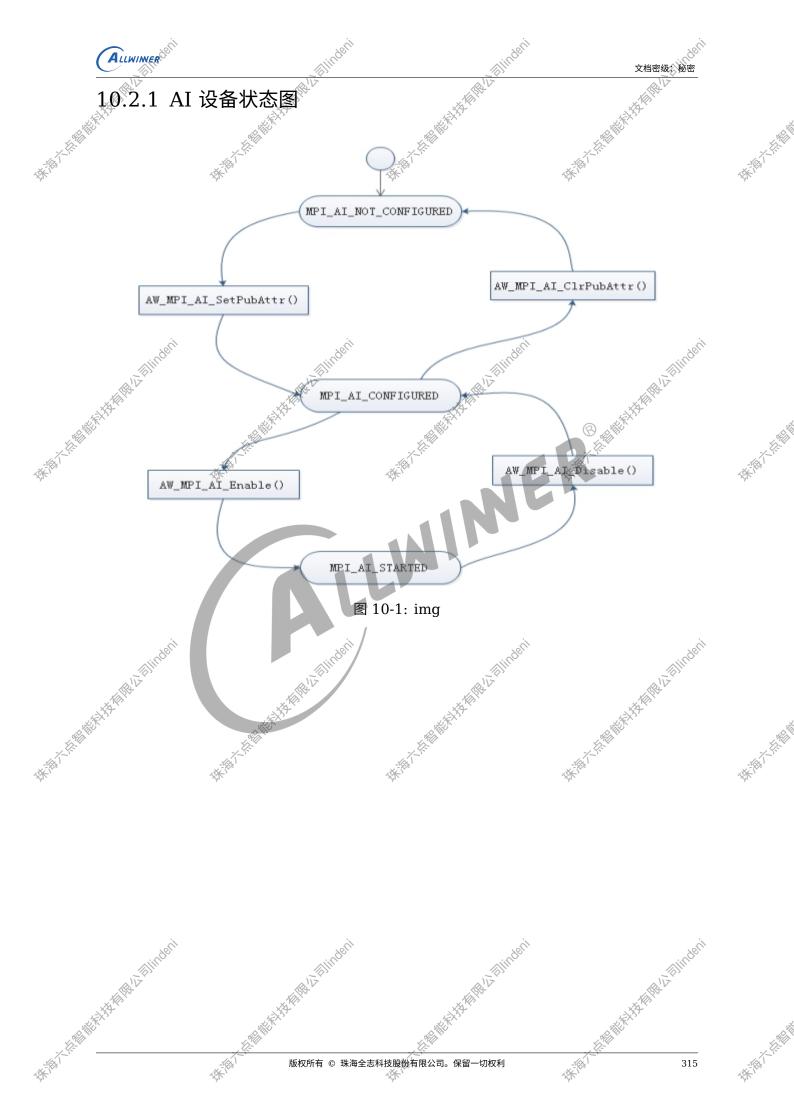
从 AI 通道到 AEnc 通道进行 pcm 数据传递可以分为 tunnel 方式 (又称绑定方式) 和 non-tunnel 方式 (又称非绑定方式), tunnel 方式为组件之间内部自动传递数据方式, app 不用参与数据传递 过程,而只需专注于应用层面的业务开发; non-tunnel 方式为 app 主动管理 pcm 数据,即从 AI 组件拿数据,然后将其送给 AEnc 组件,再从 AEnc 组件输出端口拿编码后的数据,之后将其 送 muxer 组件或网传后,最后再利用该包数据向 AEnc 组件还帧。

通过网络或文件解封装后得到的音频码流数据,需要送解码器进行解码以恢复为原始音频数据, ADec 组件即是对该功能的抽象。目前其支持的解码类型与编码类型相同。与 AEnc 组件类似, 其也为双端口组件,数据传递也包括 tunnel 方式和 non-tunnel 方式。其实现的功能与 AEnc 正 好相反,不再累述。

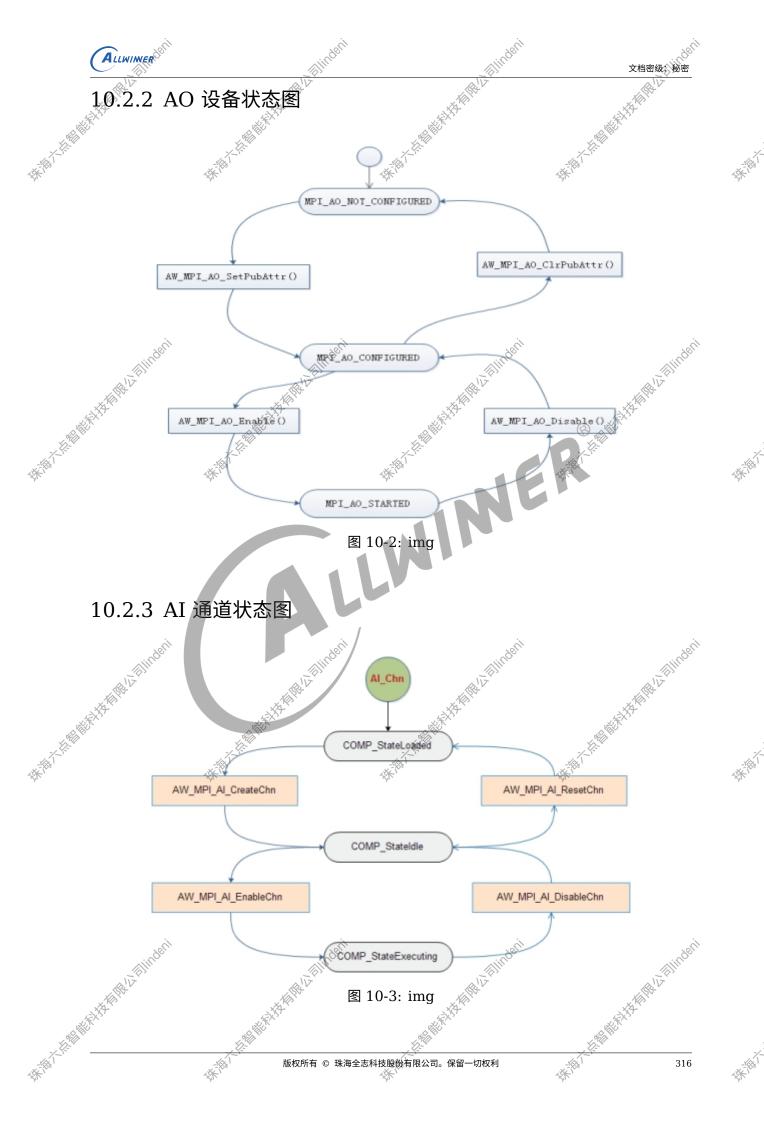
# 10.2 功能描述

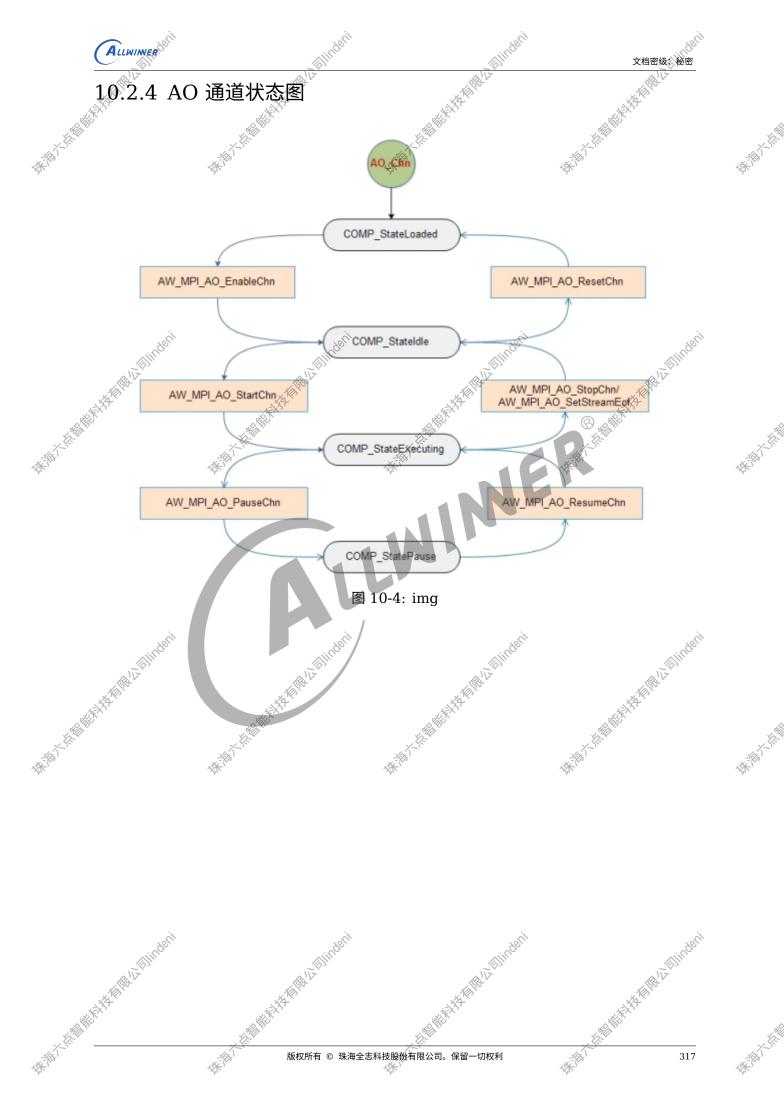
AIO 设备、AI 通道、AO 通道、AEnc 通道、ADec 通道内部都有 工作,其状态转换图如下:

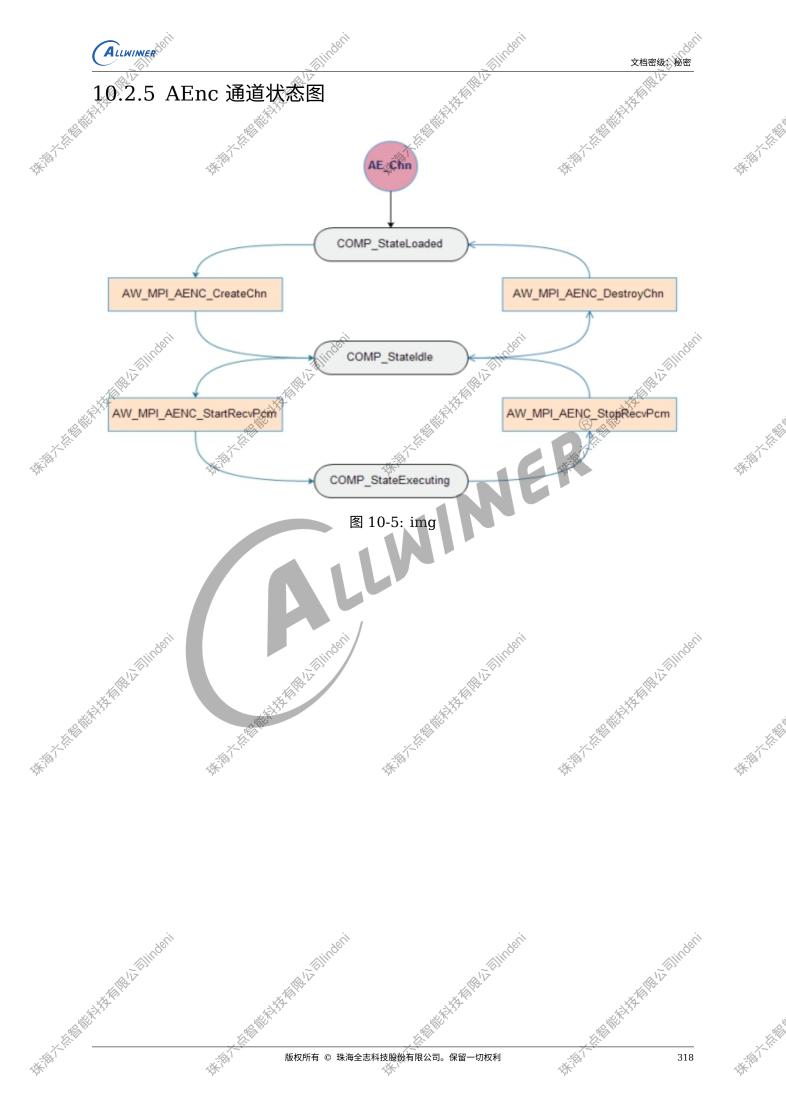




版权所有 © 珠海全志科技股份有限公司。保留一切权利

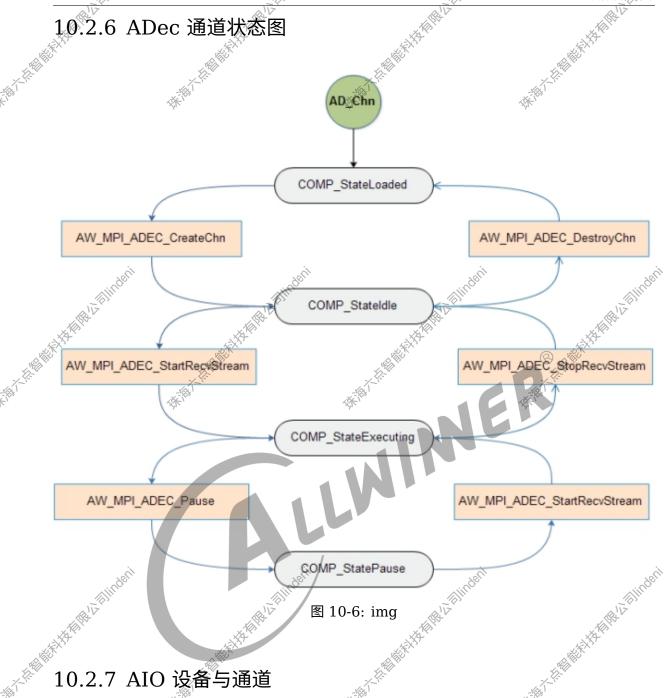






版权所有 © 珠海全志科技股份有限公司。保留一切权利





设备:从软硬件上划分,其属于硬件的范畴,由 IC 设计决定其数量。

通道:属于软件上的虚拟范畴,可存在多个,即一个设备下可挂多个通道实例。例如,在多路录制时,一个 AI 设备下挂多个 AI 虚拟通道,则每个通道的输入数据都相同,因为其是由同一个音频硬件采集得到的;每个 AI 通道后面又绑定其对应的 AEnc 通道,通过对 AEnc 设置不同的编码格式,可得到多路输入音频相同但编码格式不同的输出码流。如下为其拓扑图:

图 10-7: img

注意,目前 mpp 在音频输出上只支持 -个 AO 设备下挂一个 AO 通道,暂不支持多路 AO 通道 同时存在的混音功能。

mpi 通过提供 ai/ao 通道的控制接口,间接地操作 ai/ao 设备。在应用程序开发时,通过操作 ai/ao 通道提供的接口,可完成 ai/ao 设备使能、去使能等操作。 NE

# 10.2.8 音频回声消除

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

# 10.2.9 音频降噪

AI 通道将采集的音频数据,利用降噪库去除白噪声。

# 10.3 音频接口调用流程介绍

使用 mpp 进行音频的应用程序开发时,需特别注意接口调用流程及顺序,否则容易出现调用失 败、段错误、帧节点回收出错等问题。

通常情况下,AI/AO 通道调用流程为:设备属性设置、设备使能、通道创建、通道使能、送 pcm 数据、停止通道、复位通道、销毁通道、设备去使能。

AEnc/ADec 通道调用流程类似:通道创建、启动通道、送/还 pcm 数据或 stream 数据、停止 接收 pcm 或 stream、复位通道、销毁通道。



# 10.3.1 AI 通道使用流程

进行音频采集,需打开 aǐ 设备、创建 ai 通道后才能进行音频 pcm 的采集,参考例子为 sam-ple\_ai.c,其流程如下:

Step1:AW MPI AI SetPubAttr() //设置 ai 设备的 pcm 采集参数

Step2:AW MPI AI Enable() //启动 ai 设备,目前可不调用

Step3:AW MPI AI CreateChn() //创建 ai 通道,并将其挂到 ai 设备下

Step4:AW MPI AI EnableChn() //运行 ai 通道

//AI 设备往。AI 通道源源不断送 pcm, App 拿数据/还帧

loop

Step5:AW\_MPI\_AI\_GetFrame() //app 去拿 pcm 数据

Step6:AW\_MPI\_AI\_ReleaseFrame() //app 还帧给 ai 通道

}

Step7:AW\_MPI\_AI\_DisableChn() //停止 ai 通道接收数据

Step8:AW MPI AI ResetChn() //ai 通道复位

Step9:AW MPI AI DestroyChn() //销毁 ai 通道,并从 ai 设备下退出

Step10:AW MPI AI Disable() //停止 ai 设备,目前可不调用

其中,step2 的接口目前可不调用,其已经做到了 step3 的接口中,其作用为在创建通道时根据需要来打开 AI 设备。当 AI 通道在第一次调用 AW\_MPI\_AI\_CreateChn() 时,其实现中会主动调用 AW\_MPI\_AI\_Enable() 打开 AI 设备,第二个以及后面的通道在调用 AW\_MPI\_AI\_CreateChn() 创建通道时,其内部实现则不需再次打开 AI 设备,而是直接单纯的创建组件。

其中,step9 的 AW\_MPI\_AI\_DestroyChn() 会将该通道从 AI 设备所管理的多个通道列表中删除(即使 ai 设备又采集到 pcm 数据,但不会再往本通道传递了,而会送往其它 AI 通道),接下来释放本通道的全部资源。

其中,step10 的 AW\_MPI\_AI\_Disable() 的接口可不调用,因其已经做到了 step9 的接口实现中。当销毁通道 (AW\_MPI\_AI\_DestroyChn) 时,AI 通道从 AI 设备下删除,若 AI 设备下只包含一个 AI 通道,则在销毁通道的同时会停止 AI 设备,即最后一个通道在销毁通道时会连带关闭 AI 设备。那么,当最后一个通道调用 AW\_MPI\_AI\_DestroyChn() 时,其内部实现会自动关闭 AI 设备,用户无需手动调用 Step10 的 AW MPI AI Disable()。

板权所有 © 珠海全志科技股份有限公司。保留一切权利



# 10.3.2 AO 通道使用流程

进行音频播放,需打开 ao 设备、创建 ao 通道后才能进行音频 pcm 数据的播放,参考例子为 sample\_ao.c,其流程如下:

Step1:AW MPI AO SetPubAttr() //设置 ao 设备的 pcm 参数

Step2:AW MPI AO Enable() //启动 ao 设备,可不调用

Step3:AW MPI AO EnableChn() //创建 ao 通道,并将其挂到 ao 设备下

Step4:AW MPI AO StartChn() //运行 ao 通道

//用户往 AQ 通道源源不断送 pcm

loop

Step5:AW\_MPI\_AO\_SendFrame() //app 手动送 pcm 数据到 ao 通道

}

Step7:AW MPI AO StopChn() //停止 ao 通道

Step8:AW MPI AO DisableChn() //销毁 ao 通道,并关闭 ao 设备

Step9:AW MPI AO Disable() //停止 ao 设备,可不调用

其中,step2的AW\_MPI\_AO\_Enable()目前可不调用,其已经在 step3的AW\_MPI\_AO\_EnableChn()接口实现中被调用了。

其中,step8 的 AW\_MPI\_AO\_DisableChn() 会将本通道从 AO 设备下删除,并关闭 AO 设备,接下来释放本通道的全部资源。即该接口内部实现中已经调用了 step9 的 AW MPI AO Disable()接口,用户在使用 AO 做应用开发时可不调用该接口。

# 10.3.3 AEnc 通道调用流程

进行音频编码,需按照通道创建、启动通道、送 pcm 数据、取 stream 数据、还帧、停止通道、复位通道、销毁通道的顺序使用,参考例子为 sample\_aenc.c,其流程如下:

Step1:AW MPI AEnc CreateChn() //创建通道

Step2:AW MPI AEnc StartRecvPcm() //启动通道

//non-tunnel 方式下用户往 AEnc 通道源源不断送 pcm、取 stream、还 stream。。。

loop

版权所有 © 珠海全志科技股份有限公司。保留一切权利



Step3:AW MPI AEnc SendFrame() //app 手动送 pcm 数据到 aenc 通道

Step4:AW\_MPI\_AEnc\_GetStream() //app 手动拿编码数据

Step4:AW MPI AEnc ReleaseStream() //app 手动还帧到编码库

}

Step5:AW\_MPI\_AEnc\_StopRecvPcm() //停止通道

Step6:AW MPI AEnc ResetChn() //复位通道

Step7:AW MPI AEnc DestroyChn()//销毁通道

注意,在 non-tunnel 方式下,应用程序需手动方式送 pcm 数据到 aenc 通道,组件内部线程 对 pcm 数据进行编码,生成的 stream 在输出队列中进行管理,等待用户取走;当用户取走 stream 后,用户还需将该帧码流还给编码通道,以释放其占用的 buffer。在 tunnel 方式下,内部数据通路进行了动态绑定 (AI-AEnc),应用无需手动调用送帧、取帧、还帧的接口,因其在内部实现中自动被调用。

# 10.3.4 ADec 通道调用流程

进行音频解码,需按照通道创建、启动通道、送 stream 数据、取 pcm 数据、还帧、停止通道、复位通道、销毁通道的顺序使用,参考例子为 sample adec.c,其流程如下:

Step1:AW MPI ADec CreateChn() //创建通道

Step2:AW MPI ADec StartRecvStream() //启动通道

//non-tunnel 方式下用户往 ADec 通道源源不断送 stream、取 pcm、还 pcm。。。

loop

{

Step3:AW\_MPI\_ADec\_SendStream() //app 手动送 stream 数据到 adec 通道

Step4:AW\_MPI\_ADec\_GetFrame() //app 手动拿 pcm 数据

Step4:AW\_MPI\_ADec\_ReleaseFrame() //app 手动还帧到解码库

}

Step5:AW MPI ADec StopRecvStream() //停止通道

Step6:AW MPI ADec ResetChn() //复位通道



# Step7:AW MPI ADec DestroyChn() //销毁通道

注意,在 non-tunnel 方式下,应用程序需手动方式送 stream 数据到 adec 通道,组件内部线 程对 stream 数据进行解码,生成的 pcm 在输出队列中进行管理,等待用户取走; 当用户取走 pcm 后,用户还需将该帧 pcm 数据还给解码通道,以释放其占用的 buffer。在 tunnel 方式 下,内部数据通路进行了动态绑定 (ADec-AO),应用无需手动调用送帧、取帧、还帧的接口,因 其在内部实现中自动被调用。

# 10.4 API 接口

# 10.4.1。音频输入

# 10.4.1.1 AW\_MPI\_AI\_SetPubAttr

【目的】

设置 AI 设备属性。

【语法】

Au -ERRORTYPE AW MPI AI SetPubAttr(AUDIO DEV AudioDevId, const AIO ATTR S \*pstAttr);

### 【参数】

参数 <sub>nin</sub> de <sup>ni</sup>	描述	Jindeni
AudioDevId	音频设备号。	输入
pstAttr	AI 设备属性指针。	输入

【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

【需求】

1头文件: mm comm aio.h、mm common.h

【注意】



无必

【举例】

无。

# ${\bf 10.4.1.2 \quad AW\_MPI\_AI\_GetPubAttr}$

【目的】

获取 AI 设备属性。

【语法】

ERRORTYPE AW\_MPI\_AI\_GetPubAttr(AUDIO\_DEV\_AudioDevId, AIO\_ATTR\_S \*pstAttr);

【参数】

参数 描述

AudioDevId 音频设备号。 输入 pstAttr AI 设备属性指针。 输出

【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,其值见错误码。

【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

无。

【举例】

无。



### AW\_MPI\_AI Enable 10.4.1.3

【目的】

启用 AI(音频采集) 设备。

【语法】

ERRORTYPE AW\_MPI\_AI\_Enable(AUDIO\_DEV AudioDevId);

【参数】

参数 描述 E Rent Helphar Hall Berner AudioDevId 音频设备号。

【返回值】

返回值 描述 0 成功 非 0 失败,其值见错误码。

【需求】

l 头文件: mm comm aio.h、mm common.h

【注意】

1 必须在启用前配置音频设备属性,否则返回属性未配置错误。

1 如果音频设备已经处于启用状态,则直接返回成功。此种场景常见于多路音视频同时录制时,最 早一路已经开始采集声音数据后,又创建一个 recorder 进行音视频录制。多个 AI 组件实例同时 运行时,音频数据复用,即硬件采集到的每帧 pcm 数据分别送往不同 AI 组件实例,又各自送往 其对应的 AEnc 组件实例。

【举例】

无。

### 10.4.1.4 AW\_MPI\_AI\_Disable

【目的】

禁用 AI(音频采集) 设备。



### 【语法】

ERRORTYPE AW\_MPI\_AI\_Disable(AUDIO\_DEV\_AudioDevId);

### 【参数】

 参数	描述	
AudioDevId	音频设备号。	输入

### 【返回值】

返回值	描述		is.		is.
0 Minde	成功		Allinderi		Winder
# 0	失败,	其值见错误	码。		RIV
mm_cor	nmon.	1	N		B. V. Blinderi
		Ma			

### 【需求】

l 头文件: mm comm aio.h、mm common.h

### 【注意】

- 1 如果音频采集设备已经处于禁用状态,则直接返回成功。
- 1 禁用音频设备前必须先禁用该设备下已启用的所有 AI 通道。
- l 要求在禁用 AI 设备之前,先禁用与之关联、使用 AI 的音频数据的 AENC 通道和 AO 设备,否则可能导致该接口调用失败。

### 【举例】

多路音视频录制时,最先结束录制的 recorder 在禁用音频设备时,会遍历使用该设备的 AI 通道。如果还有其它 AI 通道在使用该音频设备,则只关闭该 AI 通道而不关闭该音频设备就直接返回;最后一个 recorder 退出时,由于音频采集设备只被该 recorder 的 AI 通道占用,因此直接关闭 AI 通道和音频设备。

综上有,音频采集设备在被共享时,只有等到最后一次被禁用时才能真正释放。

## 10.4.1.5 AW\_MPI\_AI\_CreateChn

【目的】

创建 AI 通道。

【语法】



# ERRORTYPE AW\_MPI\_AI\_CreateChn(AUDIO\_DEV AudioDevId, AI\_CHN AiChn);

### 【参数】

参数 描述 AudioDevId 音频设备号。 输入 AiChn AI 通道号 输入

### 【返回值】

返回值	描述	
0 deri	成功	
非0111	失败,	其值见错误码。

【需求】

1头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

E Republikation of the state of l 创建 AI 通道前,必须保证先启用其所属的 AI 设备,否则返回设备未启动的错误码。

l 创建 AI 通道后才能调用启用通道 (AW\_MPI\_AI\_EnableChn) 接口。

【举例】

无。

### 10.4.1.6 AW MPI AI DestroyChn

【目的】

销毁 AI 通道。

【语法】

ERRORTYPE AW\_MPI\_AI\_DestroyChn(AUDIO\_DEV AudioDevId, AI\_CHN AiChn);

【参数】

参数	描述	
AudioDevId	音频设备号。	——— 输入
AiChn	AI 通道号	输入



# 【返回值】

<i>λ</i> − <i>λ</i> *		Z-X ·
A STATE OF THE STA	返回值	描述
THE THE	0	成功
,	非 0	失败,其值见错误码。

【需求】

l 头文件: mm comm aio.h、mm common.h

【注意】

l 销毁 AI 通道前,必须保证该 AI 通道已被复位。

The state of the s 1录音结束时的操作步骤: DisableChn->ResetChn->DestroyChn。

【举例】

无。

### AW MPI AI ResetChn 10.4.1.7

【目的】

复位 AI 通道。

【语法】

ERRORTYPE AW\_MPI\_AI\_ResetChn(AUDIO\_DEV AudioDevId, AI\_CHN AiChn);

【参数】

参数	描述	
AudioDevId	音频设备号。	输入
AiChn	AI 通道号	输入

【返回值】

返回值	描述	
0	成功	
非 0 %	失败,	其值见错误码。

需求】



1头文件: mm comm aio.h、mm common.h

【注意】

l 录音结束时的操作步骤: DisableChn->ResetChn->DestroyChn。

【举例】

无。

### 10.4.1.8 AW\_MPI\_AI\_PauseChn

【目的】

暂停 AI 通道。

【语法】

ERRORTYPE AW\_MPI\_AI\_PauseChn(AUDIO\_DEV AudioDevId, AI\_CHN AiChn);

【参数】

参数	描述	La.
AudioDevId	音频设备号。	输入
AiChn	AI 通道号	输入

【返回值】

	//	
返回值	描述	ALIV MAN
0	成功	
非 0	失败,	其值见错误码。

【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

l 用于录音过程中暂停本通道的声音采集,若同时存在其它 AI 通道,并不影响其它通道的声音采集。

【平例】

无。



### AW\_MPI\_AI\_ResumeChn 10.4.1.9

【目的】

恢复 AI 通道状态至运行态。

【语法】

ERRORTYPE AW\_MPI\_AI\_ResumeChn(AUDIO\_DEV AudioDevId, AI\_CHN AiChn);

【参数】

参数	描述	
AudioDevIc AiChn	l 音频设备号。 AI 通道号	输入 <sup>86</sup> 。 输入
		输入。
返回值	描述	
	成功 E败,其值见错误	码。

【返回值】

返回值	描述《
0	成功
非0	失败,其值见错误码。

【需求】

l 头文件: mm comm aio.h、mm common.h

【注意】

l将 AI 通道从暂停态转换为运行态,需配合 AW\_MPI\_AI\_PauseChn 接口使用。

【举例】

无。

### 10.4.1.10 AW\_MPI\_AI\_EnableChn

【目的】

启用 AI 通道。

【语法】

ERRORTYPE AW\_MPI\_AI\_EnableChn(AUDIO\_DEV AudioDevId, AI\_CHN AiChn);

【参数】



描述 参数 AudioDevId 音频设备号。 输入 AI 通道号 AiChn 输入

【返回值】

返回值	描述	
0 非 0	成功 失败,	其值见错误码。

【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

l 启用 AI 通道前,必须先启用其所属的 AI 设备并且该 AI 通道已被创建, 否则返回设备未启动的 错误码。

【举例】

无。

### AW\_MPI\_AI\_DisableChn 10.4.1.11

【目的】

禁用 AI 通道。

【语法】

ERRORTYPE AW\_MPI\_AI\_DisableChn(AUDIO\_DEV AudioDevId, AI\_CHN AiChn);

【参数】

参数	描述	
AudioDevId	音频设备号。	输入
AiChn	AI 通道号	输入



返回值	描述	
0	成功	A STATE OF THE STA
非 0	失败,	其值见错误码。

l 头文件: mm comm aio.h、mm common.h

【注意】

1禁用 AI 通道前,必须保证 AI 设备已启用并且该 AI 通道已处于运行状态,否则返回错误码。

10.4.1.12 AW MPI\_AI\_GetFrame

【目的】

获取音频帧。

【语法】

DIO T ERRORTYPE AW MPI AI GetFrame(AUDIO DEV AudioDevId, AI CHN AiChn, AU-DIO\_FRAME\_S pstFrm, AEC\_FRAME\_S pstAecFrm, int s32MilliSec);

### 【参数】

参数	描述			7
AudioDevId	音频设备号。		有	俞入
AiChn	AI 通道号		斩	俞入
pstFrm	音频帧结构体指针。	K. Comment of the com	茶	俞出
pstAecFrm	回声抵消参考帧结构体	指针。	斩	俞出
s32MilliSec	获取数据的超时时间-1	表示阻塞模式,无数据时	一直等待;0 表示非   输	俞入
	阻塞模式,无数据时则 超时则报错返回。	报错返回;>0 表示阻塞 s	32MilliSec 毫秒,	

### 【返回值】

返回值	描述	- California
O L	成功	The last of the la
非 0	失败,	其值见错误码。



l 头文件: mm\_comm\_aio.h、mm\_common.h

### 【注意】

1 如果 AI 的回声抵消功能已使能, pstAecFrm 不能是空指针; 如果 AI 的回声抵消功能没有使 能,pstAecFrm 可以置为空。

l AI 模块会缓存音频帧数据,用于用户态获取。缓存的深度通过 AW MPI AI SetChnParam 接 口设定,默认为 0。

ls32MilliSec的值必须大于等于-1,等于-1时采用阻塞模式获取数据,等于0时采用非阻塞模 式获取数据、大于 0 时,阻塞 s32MilliSec 毫秒后,没有数据则返回超时并报错。

THE REPORT OF THE PARTY OF THE 1 获取音频帧数据前,必须先使能对应的 AI 设备和 AI 通道

### 【举例】

无。

### 10.4.1.13 AW MPI AI ReleaseFrame

【目的】

释放音频帧。

### 【语法】

ERRORTYPE AW MPI AI ReleaseFrame(AUDIO DEV AudioDevId, AI CHN AiChn, AUDIO FRAME S pstFrm, AEC FRAME S pstAecFrm);

### 【参数】

参数	描述	
AudioDevId	音频设备号。	输入
AiChn	AI 通道号	输入
pstFrm	音频帧结构体指针。	输入
pstAecFrm	回声抵消参考帧结构体指针。	输入

### 【返回值】

返回值	描述	;
Q	成功	ALIV
非 0	失败,	其值见错误码。



l 头文件: mm\_comm\_aio.h、mm\_common.h

### 【注意】

1 如果不需要释放回声抵消参考帧,pstAecFrm 置为 NULL 即可。

l 当 app 取走 pcm 数据后,调用该 api 可以释放 AI 组件的 pcmBufferManager 缓冲队列中的对应的数据帧,释放出空间以便于存储采集到的最新的 pcm 数据。

### 【举例】

无。

### 10.4.1.14 AW\_MPI\_AI\_SetChnParam

### 【目的】

设置 AI 通道属性。

### 【语法】

ERRORTYPE AW\_MPI\_AI\_SetChnParam(AUDIO\_DEV AudioDevId, AI\_CHN AiChn, AI\_CHN\_PARAM\_S \*pstChnParam);

### 【参数】

参数	描述	deril	
AudioDevId	音频设备号。	Elling	输入
AiChn	AI 通道号	7	输入
pstChnParam	音频通道参数结构体指	計。	输入

### 【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

### 【需求】

l 头文件: mm comm aio.h、mm common.h

### 【注意】



1 通道参数目前只有一个成员变量,用于设置用户获取音频帧的缓存深度,默认深度为 0。该成员变量的值不能大于 30。

l 建议先调用 AW\_MPI\_AI\_GetChnParam 接口获取默认配置,再调用本接口修改配置,以便于后续扩展。

【举例】

无。

# 10.4.1.15 AW\_MPI\_AI\_GetChnParam

【目的】

获取 AI 通道属性。

【语法】

ERRORTYPE AW\_MPI\_AI\_GetChnParam(AUDIO\_DEV AudioDevId, AI\_CHN AiChn, AI\_CHN\_PARAM\_S \*pstChnParam);

### 【参数】

参数	描述	
AudioDevId	音频设备号。	输入
AiChn	AI 通道号	输入
pstChnParam	音频通道参数结构体指针。	输出

【返回值】

返回值	描述
0	成功
非 0	失败,其值见错误码。

【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

无。

【举例】

池。



# 10.4.1.16 AW\_MPI\_AI\_EnableReSmp

【目的】

启用 AI 重采样。

【语法】

ERRORTYPE AW\_MPI\_AI\_EnableReSmp(AUDIO\_DEV AudioDevId, AI\_CHN AiChn, AUDIO SAMPLE RATE E enOutSampleRate);

### 【参数】

参数	描述	inderi		inderi linderi
AudioDevId	音频设备号。	ALIV TO THE PERSON OF THE PERS	输入	ALIV TO THE PERSON OF THE PERS
AiChn	AI 通道号	XA KINST	输入	XA KIND
en Out Sample Rate	音频重采样的	输出采样率。	输入	(C) AL HELD Y
A CONTRACTOR OF THE PARTY OF TH	A. T.	-1		
返回值	描述	Ma		
0	成功			

【返回值】

返回值	描述		N
0	成功		
非0	失败,	其值见错i	误码。

### 【需求】

l 头文件:mm\_comm\_aio.h、mm\_common.h

### 【注意】

- l 在启用 AI 通道之后,调用此接口启用重采样功能。
- 1允许重复启用重采样功能,但必须保证后配置的属性与之前配置的属性-
- 1 在禁用 AI 通道之后,如果重新启用 AI 通道,并使用重采样功能,需调用此接口重新启用重采 样。

【举例】

无。

# AW\_MPI\_AI\_DisableReSmp

(目的)



禁用 AI 重采样。

### 【语法】

ERRORTYPE AW MPI\_AI\_DisableReSmp(AUDIO\_DEV AudioDevId, AI\_CHN AiChn);

### 【参数】

参数	描述	
AudioDevId	音频设备号。	输入
AiChn	AI 通道号	输入

### 【返回值】

返回值 描述

0 成功
非 0 失败,其值见错误码。

### 【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

### 【注意】

l 不再使用 AI 重采样功能的话,应该调用此接口将其禁用。

1要求在调用此接口之前,先禁用使用该 AI 设备相应通道音频数据的 AENC 通道和 AO 通道,否则可能导致该接口调用失败。

### 【举例】

无。

### 10.4.1.18 AW\_MPI\_AI\_SetVqeAttr

### 【目的】

设置 AI 的声音质量增强功能相关属性。

### 【语法】

ERRORTYPE AW\_MPI\_AI\_SetVqeAttr(AUDIO\_DEV AiDevId, AI\_CHN AiChn,AI\_VQE\_CONFIG\_S \*pstVqeConfig);

# 【参数】



	.11/2"	∆ 1\Z	
参数	描述		
AudioDevId	音频设备号。		输入 🔊
AiChn	AI 通道号	JAN TE	输入
pstVqeConfig	音频输入声音	质量增强配置结构体指针	输入

【返回值】

返回值 描述

0 成功
非 0 失败,其值见错误码。

mm\_common.h

Representation of the common of the common

【需求】

】头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

无。

【举例】

无。

# 10.4.1.19 AW\_MPI\_AI\_GetVqeAttr

【目的】

获取 AI 的声音质量增强功能相关属性。

【语法】

ERRORTYPE AW\_MPI\_AI\_GetVqeAttr(AUDIO\_DEV AiDevId, AI\_CHN AiChn, AI VQE CONFIG S \*pstVqeConfig);

【参数】

参数	描述	
AudioDevId	音频设备号。	输入
AiChn	AI 通道号	输入
pstVqeConfig	音频输入声音质量增强配置结构体指针	输出

[返回值]



返回值	描述	NA TOPING
0	成功	A STATE OF THE STA
非 0	失败,	其值见错误码。

l 头文件: mm comm aio.h、mm common.h

【注意】

1 获取声音质量增强功能相关属性前必须先设置相对应 AI 通道的声音质量增强功能相关属性。

AW\_MPI\_AI\_EnableVqe 10.4.1.20

【目的】

启用 AI 的声音质量增强功能。

【语法】

TO T ERRORTYPE AW\_MPI\_AI\_EnableVqe(AUDIO\_DEV AiDevId, AI\_CHN AiChn);

【参数】

- inde		200
参数。	描述	\$1/2
AudioDevId	音频设备号。	输入
AiChn	AI 通道号	输入

【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

注意】



- l 启用声音质量增强功能前必须先启用相对应的 AI 通道。
- l 多次使能相同 AI 通道的声音质量增强功能时,返回成功。
- l 禁用 AI 通道后,如果重新启用 AI 通道,并使用声音质量增强功能,需调用此接口重新启用声音质量增强功能。

【举例】

无。

# 10.4.1.21 AW\_MPI\_AI\_DisableVqe

【目的》

禁用 AI 的声音质量增强功能。

【语法】

ERRORTYPE AW MPI\_AI\_DisableVqe(AUDIO\_DEV AiDevId, AI\_CHN AiChn);

### 【参数】

参数	描述	
AudioDevId	音频设备号。	输入
AiChn	AI 通道号	输入

【仮回值】

返回值	描述	A TOP TO SERVICE AND A SERVICE
0	成功	HE WALLEY X
非 0	失败,	其值见错误码。

【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

- l 不再使用 AI 声音质量增强功能时,应该调用此接口将其禁用。
- 1. 多次禁用相同 AI 通道的声音质量增强功能,返回成功。

【举例】



无。

# 10.4.1.22 AW\_MPI\_AI\_SetTrackMode

【目的】

设置 AI 声道模式。

【语法】

ERRORTYPE AW\_MPI\_AI\_SetTrackMode(AUDIO\_DEV AudioDevId, AUDIO\_TRACK\_MODE\_E enTrackMode);

【参数】

- Religinden		Iz Blindeni		Alle Vallinderi
参数	描述		XXX	A THINK
AudioDevId enTrackMode	音频设备号。 AI 声道模式。	输入输入	O THE REAL PROPERTY.	
返回值描述		NE		

【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,其值见错误码。

【需求】

l头文件: mm comm aio.h、mm common.h

【注意】

l在 AI 设备成功启用后再调用此接口。

1 该接口功能目前不支持。

l AI 设备工作在 I2S 模式时,支持设置声道模式,PCM 模式下不支持。

【举例】

无。



# AW\_MPI\_AI\_GetTrackMode

【目的】

获取 AI 声道模式。

【语法】

 $ERRORTYPE\ AW\_MPI\_AI\_GetTrackMode(AUDIO\_DEV\ AudioDevId,\ AUDIO\_TRACK\_MODE\_E$ \*penTrackMode);

### 【参数】

参数		描述	indeni	indeni
AudioDevI	d	音频设备号。	输入	RELIVE TO SERVICE TO S
penTrackN	/lode	AI 声道模式。	输出	144 th 181
		A CHANGE CONTRACTOR		C. E.
返回值	描述		NE	
0	成功			
非 0	失败	,其值见错误码。	)	

【返回值】

返回值	描述
0	成功
非 0	失败,其值见错误码。

### 【需求】

l 头文件: mm comm aio.h、mm common.h

【注意】

T在 AI 设备成功启用后再调用此接口。

l AI 设备工作在 I2S 模式时,支持设置声道模式,PCM 模式下不支持。

【举例】

无。

# 10.4.1.24 AW\_MPI\_AI\_ClrPubAttr

【目的】

清空 pub 属性。

语法】



ERRORTYPE AW\_MPI\_AI\_CIrPubAttr(AUDIO\_DEV AudioDevId);

### 【参数】

参数 描述 AudioDevId 音频设备号。 输入

### 【返回值】

返回值	描述	
0	成功	
非0.%	失败,	其值见错误码。

l 头文件: mm comm aio.h、mm common.h

### 【注意】

1清除设备属性前,需要先停止设备。

【举例】

无。

### AW\_MPI\_AI\_SaveFile 10.4.1.25

【目的】

开启音频输入保存文件功能。

【语法】

ERRORTYPE AW\_MPI\_AI\_SaveFile(AUDIO\_DEV AudioDevId, AI\_CHN AiChn, AU-DIO SAVE FILE INFO S \*pstSaveFileInfo);

### 【参数】

参数	描述	
AudioDevId	音频设备号。	输入
AiChn	AI 通道号	输入
pstSaveFileInfo	音频保存文件属性结构体指针。	输入

版权所有 © 珠海全志科技股份有限公司。保留一切权利

Report of the state of the stat



【返回值】

V-V.		
THE THE PERSON NAMED IN COLUMN TO PERSON NAM	返回值	描述
THE TENE	0 非 0	成功 失败,其值见错误码。

【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

无。

10.4.1.26 AW\_MPI\_AI\_QueryFileStatus

【目的】

查询 AI 通道当前 pcm 文件保存状态。

【语法】

ERRORTYPE AW\_MPI\_AI\_QueryFile(AUDIO\_DEV AudioDevId, AI\_CHN AiChn, AU-DIO\_SAVE\_FILE\_INFO\_S \*pstSaveFileInfo);

参数	描述	
AudioDevId	音频设备号。	输入
AiChn	AI 通道号	输入
pstSaveFileInfo	音频保存文件属性结构体指针。	输出

返回值	描述	
0	成功	
非 Qinder	失败,	其值见错误码。



1头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

无。

【举例】

无。

### 10.4.1.27 AW\_MPI\_AI\_SetVqeVolume

【目的】

设置 AI 设备音量大小。

【语法】

ERRORTYPE AW\_MPI\_AI\_SetVqeVolume(AUDIO\_DEV AudioDevId, AI\_CHN AiChn, int s32VolumeDb);

### 【参数】

参数	描述	
AudioDevId	音频设备号。	输入
AiChn	AI 通道号	输入
s32VolumeDb	音频设备音量大小。	输入

【返回值】

返回值	描述
0	成功
非 0	失败,其值见错误码。

【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

【举例】



无必

### 10.4.1.28 AW\_MPI\_AI\_GetVqeVolume

【目的】

获取 AI 设备音量大小。

【语法】

ERRORTYPE AW\_MPI\_AI\_GetVqeVolume(AUDIO\_DEV AudioDevId, AI\_CHN AiChn, int \*ps32VolumeDb);

【参数】

参数	描述	
AudioDevId	音频设备号。	输入
AiChn	AI 通道号	输入
ps32VolumeDb	音频设备音量大小指针。	输出

【返回值】

返回值	描述	
0	成功	
非0	失败,	其值见错误码。

【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

无。

【举例】

无。

### ${\bf 10.4.1.29 \quad AW\_MPI\_AI\_RegisterCallback}$

【目的】

设备回调函数给 AI 通道。



### 【语法】

ERRORTYPE AW\_MPI\_AI\_RegisterCallback(AUDIO\_DEV AudioDevId, AI\_CHN AiChn, MPPCallbackInfo \*pCallback);

### 【参数】

参数	描述	
AudioDevId	音频设备号。	输入
AiChn	AI 通道号	输入
pCallback	来自 app 层的回调信息。	输入

### 【返回值】

a la filinder	描述 成功 失败,其值见错误码。
返回值	描述
0	成功
非 0	失败,其值见错误码。
mm_cor	**INE

### 【需求】

l 头文件: mm comm aio.h、mm common.h

### 【注意】

l 在音视频录制过程中,音频数据从 AI 组件成功送到 AEnc 组件后,通过该回调信息,将音频时间长度送往 recorder 主控模块(组件向框架传递消息),用于统计文件中的音频 duration,以方便进行音视频同步处理。

### 【举例】

》无。

### 10.4.1.30 AW\_MPI\_AI\_SetVolume

### 【目的】

设置 AI 设备声音采集音量大小。

### 【语法】

ERRORTYPE AW\_MPI\_AI\_SetVolume(AUDIO\_DEV AudioDevId, int s32VolumeDb);

### 【参数】



描述 参数 AudioDevId 音频设备号。 输入 待设置的音量值。 s32VolumeDb 输入

【返回值】

返回值 描述 0 成功 非 0 失败,其值见错误码。

【需求】

1头文件: mm comm aio.h, mm common.h

【注意】

无。

【举例】

无。

## e Report of the state of the st 10.4.1.31 AW\_MPI\_AI\_GetVolume

【目的】

获取 AI 设备声音采集音量值。

【语法】

ERRORTYPE AW\_MPI\_AI\_GetVolume(AUDIO\_DEV AudioDevId, int \*ps32VolumeDb);

【参数】

参数	描述	
AudioDevId	音频设备号。	输入
ps32VolumeDb	待设置的音量值指针。	输出



返回值	描述	NA TOP OF THE PERSON NAMED IN COLUMN TO THE PERSON NAMED IN COLUMN
0	成功	A STATE OF THE STA
非 0	失败,	其值见错误码。

l 头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

无。

AW\_MPI\_AI\_SetMute 10.4.1.32

【目的】

设置 AI 设备静音状态。

【语法】

DF. ERRORTYPE AW\_MPI\_AI\_SetMute(AUDIO\_DEV AudioDevId, int bEnableFlag);

【参数】

inou		iilos
参数	描述	
AudioDevId	音频设备号。	输入
bEnableFlag	待设置的静音标志值。	输入

【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

[注意]



l 参数 bEnableFlag 为欲设置的静音状态值。当该值设置为 1 时则表示设置 AI 设备为静音状态,为 0 时为取消静音状态。

【举例】

无。

### 10.4.1.33 AW\_MPI\_AI\_GetMute

【目的】

获取 AI 设备静音状态值。

【语法】

ERRORTYPE AW\_MPI\_AI\_GetMute(AUDIO\_DEV AudioDevId, int \*pbEnableFlag);

【参数】

参数	描述《	
AudioDevId	音频设备号。	输入
pbEnableFlag	待设置的静音状态值指针。	输出

【返回值】

返回值	描述	
0 linder	成功	
非0	失败,	其值见错误码。

【需求】

1头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

无。

【举例】

无。



### 10.4.2 音频输出

### 10.4.2.1 AW\_MPI\_AO\_SetPubAttr

【目的】

设置 AO 设备属性。

【语法】

ERRORTYPE AW\_MPI\_AO\_SetPubAttr(AUDIO\_DEV AudioDevId, const AIO\_ATTR\_S \*pstAttr);

•	$\sim$	MAI	•
•	7	不丌	٠,
	~~	77 V	-4

a lizalind	ALIZ ALIZ	·····································
参数	描述	
AudioDevId	AO 设备号。	<u> </u>
pstAttr	音频输出设备属性指针。	输入
返回值	描述	

【返回值】

 返回值 描述

 0
 成功

 非 0
 失败,其值见错误码。

### 【需求】

1头文件: mm comm aio.h、mm common.h

### 【注意】

1在设置属性之前需要保证 AO 处于禁用状态,如果处于启用状态则需要首先禁用 AO 设备。

l AO 必须和 DA 配合起来才能正常工作,用户必须清楚 DA 发送的数据分布和通道的关系才能从 正确的通道发送数据。

1 对接外置 Codec 时,由于时序的问题,在 AO 设备从模式下,建议用户先配置好对接的 Codec,再配置 AO 设备;而在 AO 设备主模式下,建议用户先配置好 AO 设备,再配置对接的 Codec。对接内置 Codec 时,都需要先配置内置 Codec,再配置 AO 设备。

l 对接内置 Codec 时,AI 设备 0 和 AO 设备 0 的帧同步时钟与位流时钟不能共用,u32ClkSel 需要配置为 0。

l AO 设备主模式时,决定 AO 设备输出时钟的关键配置项是采样率、采样精度以及通道数目,采



样精度乘以通道数目即为 AO 设备时序一次采样的位宽。

- l 扩展标志对 AO 设备无效。
- 1 AO 设备属性结构体中其他项请参见 AI 模块中相关接口的描述。

【举例】

无。

### 10.4.2.2 AW\_MPI\_AO\_GetPubAttr

【目的】

获取 AO 设备属性。

【语法】

ERRORTYPE AW\_MPI\_AO\_GetPubAttr(AUDIO\_DEV AudioDevId, AIO\_ATTR\_S

【参数】

参数	描述	
AudioDevId	AO 设备号。	输入
pstAttr	音频输出设备属性指针。	输出

【返回值】

返回值	描述	
0	成功	HE WALL
非 0	失败,	其值见错误码。

### 【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

- 1 获取的属性为前一次配置的属性。
- 1 如果从未配置过属性,则返回属性未配置的错误。

【举例】



### AW\_MPI\_AO\_ClrPubAttr

【目的】

清除 AO 设备属性。

【语法】

ERRORTYPE AW MPI AO ClrPubAttr(AUDIO DEV AudioDevId);

【参数】

描述 AudioDevId AO 设备号。 输入

【返回值】

返回值 描述 0 成功 失败, 其值见错误码。

【需求】

l 头文件: mm comm aio.h、mm common.h

【注意】

1清除设备属性前,需要先停止设备。

【举例】

无。

### 10.4.2.4 AW\_MPI\_AO\_Enable

【目的】

启用 AO 设备。

【语法】



ERRORTYPE AW\_MPI\_AO\_Enable(AUDIO\_DEV AudioDevId);

【参数】

参数 描述

AudioDevId AO 设备号。 输入

【返回值】

返回值 描述 成功 0 失败,其值见错误码

【需求】

l 头文件: mm comm aio.h、mm common.h

【注意】

E Real Hartham Berger Landing l要求在启用前配置 AO 设备属性,否则会返回属性未配置的错误。

l 如果 AO 设备已经启用,则直接返回成功

【举例】

无。

AW MPI AO Disable 10.4.2.5

【目的】

禁用 AO 设备。

【语法】

ERRORTYPE AW MPI AO Disable(AUDIO DEV AudioDevId);

【参数】

描述 参数 AudioDevId AO 设备号。



返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

l 头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

l 如果 AO 设备已经禁用,则直接返回成功。

1禁用 AO 设备前必须先禁用设备下所有 AO 通道。

10.4.2.6 AW\_MPI\_AO\_EnableChn

【目的】

创建 AO 通道。

【语法】

C. Char ERRORTYRE AW\_MPI\_AO\_EnableChn(AUDIO\_DEV AudioDevId, AO\_CHN AoChn);

参数	描述	-
AudioDevId	AO 设备号。	输入
AoChn	AO 通道号。	输入

【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

【需求】

1头文件: mm\_comm\_aio.h、mm\_common.h



### 【注意】

l 启用 AO 通道前,必须先启用其所属的 AO 设备,否则返回设备未启动的错误码。

【举例】

无。

### 10.4.2.7 AW\_MPI\_AO\_DisableChn

【目的】

销毁 AO 通道。

【语法】

ERRORTYPE AW\_MPI\_AO\_DisableChn(AUDIO\_DEV AudioDevId, AO\_CHN AoChn);

【参数】

参数描述AudioDevIdAO 设备号。 输入AoChnAO 通道号。 输入

【返回值】

返回值。	描述	
0 - 1	成功	
非 0	失败,	其值见错误码。

### 【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

### 【注意】

l 销毁 AO 通道时,该通道会从其所属的 AO 设备的播放管理列表中退出,当只有该 AO 通道占用 该 AO 设备时,会主动关闭 AO 设备;如果还有其 AO 通道占用 AO 设备,则等到最后一个 AO 设备退出时,才会主动关闭 AO 设备。

l 释放 AO 通道和 AO 设备流程: AO\_StopChn->AO\_DisableChn->AO\_Disable

【举例】

丸。



### AW\_MPI\_AO\_StartChn 10.4.2.8

【目的】

启用 AO 通道。

【语法】

ERRORTYPE AW\_MPI\_AO\_StartChn(AUDIO\_DEV AudioDevId, AO\_CHN AoChn);

【参数】

参数	描述	
AudioDev		输入。Binderi
AoChn	AO 通道号。	输入
**************************************		输入。
返回值	描述	
0 非 0	成功 失败,其值见错说	吴码。

【返回值】

返回值	描述《
0	成功
非 0	失败,其值见错误码。

【需求】

l 头文件: mm comm aio.h、mm common.h

【注意】

l 启用 AO 通道前,必须先启用其所属的 AO 设备,否则返回设备未启动的错误码。

【举例】

无。

### 10.4.2.9 AW\_MPI\_AO\_StopChn

【目的】

停止 AO 通道。

【语法】

ERRORTYPE AW\_MPI\_AO\_StopChn(AUDIO\_DEV AudioDevId, AO\_CHN AoChn);

【参数】



参数	描述	A STORY
AudioDevId	AO 设备号。	输入
AoChn	AO 通道号。	输入

【返回值】

返回值 描述 0 成功 非 0 失败,其值见错误码。

【需求】

l 类文件: mm\_comm\_aio.h, mm\_common.h

【注意】

【举例】

无。

### k 10.4.2.10 AW\_MPI\_AO\_RegisterCallback

【目的】

停止 AO 通道。

【语法】

ERRORTYPE AW MPI AO RegisterCallback(AUDIO DEV AudioDevId, AO CHN AoChn, MPPCallbackInfo \*pCallback);

【参数】

参数	描述	
AudioDevId	AO 设备号。	输入
AoChn	AO 通道号。	输入
pCallback	来自 app 层的回调信息。	输入

版权所有 © 珠海全志科技股份有限公司。保留一切权利

359



述	
次 3. 3. 其值见错误码	0
	XX.

l 头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

【举例】

无。

10.4.2.11 AW\_MPI\_AO\_SendFrame

【目的】

发送 AO 音频帧。

【语法】

EV ERRORTYPE AW\_MPI\_AO\_SendFrame(AUDIO\_DEV AudioDevId, AO\_CHN AoChn, const AUDIO\_FRAME\_S \*pstData, int s32MilliSec);

### 【参数】

6/1			
参数高流	描述	ind <sup>®</sup>	La lind
AudioDevId	AO 设备号。	A STATE OF THE STA	输入
AoChn	AO 通道号。		输入
pstData	音频帧结构体指针。	tight.	输入
s32MilliSec	发送数据的超时时间。	-1 表示阻塞模式; 0 表示非阻塞模式; >0 表	输入
	示阻塞 s32MilliSec	毫秒,超时则报错返回。	

返回值	描述	
0	成功	
非 0	、失败,	其值见错误码。



1头文件: mm comm aio.h、mm common.h

### 【注意】

l 该接口用于 app 主动发送音频帧至 AO 输出,如果 AO 通道已经通过系统绑定(AW\_MPI\_SYS\_Bind)。 接口与 AI 或 ADEC 绑定,不需要也不建议调此接口。

ls32MilliSec的值必须大于等于-1,等于-1时采用阻塞模式发送数据,等于0时采用非阻塞模 式发送数据,大于 0 时,阻塞 s32MilliSec 毫秒后,则返回超时并报错。

1调用该接口发送音频帧到 AO 输出时,必须先使能对应的 AO 通道。

### 【举例】

无。

### 10.4.2.12 AW\_MPI\_AO\_EnableReSmp

【目的】

启用 AO 重采样。

### 【语法】

Ev. ERRORTYPE AW MPI AO EnableReSmp(AUDIO DEV AudioDevId, AO CHN AoChn, AUDIO SAMPLE RATE E enInSampleRate);

### 【参数】

		is
参数	描述	inger.
AudioDevId	AO 设备号。	输入
AoChn	AO 通道号。	输入
enInSampleRate	音频重采样的输入采样率	ጆ。 输入

### 【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

### 【需求】

l 头文件: mm comm\_aio.h、mm\_common.h

### 【注意】



l 应该在启用 AO 通道之后,绑定 AO 通道之前,调用此接口启用重采样功能。

1 允许重复启用重采样功能,但必须保证后配置的重采样输入采样率与之前配置的重采样输入采样率一样。

1 在禁用 AO 通道后,如果重新启用 AO 通道,并使用重采样功能,需调用此接口重新启用重采样。

l AO 重采样的输入采样率必须与 AO 设备属性配置的采样率不相同。

### 【举例】

无。

### 10.4.2.13 AW\_MPI\_AO\_DisableReSmp

【目的】

禁用 AO 重采样。

【语法】

ERRORTYPE AW\_MPI\_AO\_DisableReSmp(AUDIO\_DEV AudioDevId, AO\_CHN AoChn);

【参数】

参数	描述	
AudioDevId	AO 设备号。	——— 输入
AoChn	AO 通道号。	输入

【返回值】

返回值	描述《
0	成功
非 0	失败,其值见错误码。

### 【需求】

l 头文件: mm comm aio.h、mm common.h

【注意】

1 不再使用 AO 重采样功能的话,应该调用此接口将其禁用。

C PRINTER AND THE PROPERTY OF THE PROPERTY OF



【举例】

》无。

### 10.4.2.14 AW\_MPI\_AO\_PauseChn

【目的】

暂停 AO 通道。

【语法】

ERRORTYPE AW\_MPI\_AO\_PauseChn(AUDIO\_DEV AudioDevId, AO\_CHN AoChn);

【参数】

参数 描述
AudioDevId AO 设备号。 输入
AoChn AO 通道号。 输入

【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,其值见错误码。

【需求】

l头文件: mm comm aio.h、mm common.h

【注意】

l AO 通道暂停后,如果绑定的 ADEC 通道继续向此通道发送音频帧数据,发送的音频帧数据将会被阻塞;而如果绑定的 AI 通道继续向此通道发送音频帧数据,在通道缓冲未满的情况下则将音频帧放入缓冲区,在满的情况下则将音频帧丢弃。

1 AO 通道为禁用状态时,不允许调用此接口暂停 AO 通道。

【举例】

无。



### AW\_MPI\_AO\_ResumeChn 10.4.2.15

【目的】

恢复 AO 通道。

【语法】

ERRORTYPE AW\_MPI\_AO\_ResumeChn(AUDIO\_DEV AudioDevId, AO\_CHN AoChn);

【参数】

参数	描述	
AudioDevI		输入。
AoChn	AO 通道号。	输入
<b>X</b> <sup>+</sup>		S AND THE STATE OF
返回值	描述等	
	成功	M
非 0	失败,其值见错词 ————————————————————————————————————	码。

【返回值】

返回值	描述
0	成功
非0	失败,其值见错误码。

【需求】

l 头文件: mm comm aio.h、mm common.h

【注意】

1AO 通道暂停后可以通过调用此接口重新恢复。

l AO 通道为暂停状态或使能状态下,调用此接口返回成功;否则调用将返回错误。

【举例】

无。

### 10.4.2.16 AW\_MPI\_AO\_Seek

【目的】

Player 跳播播放。

【语法】

ERRORTYPE AW\_MPI\_AO\_Seek(AUDIO\_DEV AudioDevId, AO\_CHN AoChn);



### 【参数】

相關於	~ 参数	描述	
	AudioDevId AoChn	AO 设备号。 AO 通道号。	输入

【返回值】

		•	
返回值	描述		
0	 成功	•	
非0	失败,其值见错误码。	, sii	ing
AR VERLINGE	成功 失败,其值见错误码。 mmon.h	Allinderii	Alle I I I I I I I I I I I I I I I I I I
mm_con	nmon.h	© HALL	
	Willey Live		
退到某一时		A D	
11	L		
O_Clea	rChnBuf	deri'	deri

【需求】

l 头文件:mm\_comm\_aio.h、mm\_common.h

【注意】

1 用于播放视频文件时快进或快退到某一时刻。

【举例】

无。

### 10.4.2.17 AW\_MPI\_AO\_ClearChnBuf

【目的】

清除 AO 通道中当前的音频数据缓存。

【语法】

ERRORTYPE AW\_MPI\_AO\_ClearChnBuf(AUDIO\_DEV AudioDevId ,AO\_CHN AoChn);

【参数】

参数	描述	
AudioDevId	AO 设备号。	输入
AoChn	AO 通道号。	输入



返回值	描述	
0	成功	W. W
非 0	失败,	其值见错误码。

l 头文件: mm comm aio.h、mm common.h

【注意】

l 在 AO 通道成功启用后再调用此接口。

l 为完全清除解码回放通路上所有缓存数据,此接口还应该与 AW\_MPI\_ADEC\_ClearChnBuf 接 口配合使用。

# 10.4.2.18 AW\_MPI\_AO\_QueryChnStat 【目的】 查询 AO 通道中当前的音频数据缓存状态。 【语法】

【语法】

ERRORTYPE AW MPI AO QueryChnStat(AUDIO DEV AudioDevId ,AO CHN AoChn, AO CHN\_STATE\_S \*pstStatus);

【参数】

参数	描述	
AudioDevId	AO 设备号。	输入
AoChn	AO 通道号。	输入
pstStatus	缓存状态结构体指针。	输出

返回值	描述	
0 Minder	成功	
非0	失败,	其值见错误码。



1头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

l 在 AO 通道成功启用后再调用此接口。

【举例】

无。

### 10.4.2.19 AW\_MPI\_AO\_SetTrackMode

【目的】

设置 AO 设备声道模式。

【语法】

ERRORTYPE AW\_MPI\_AO\_SetTrackMode(AUDIO\_DEV AudioDevId, AUDIO\_TRACK\_MODE\_E enTrackMode);

### 【参数】

参数	描述	
AudioDevId	AO 设备号。	输入
enTrackMode /	音频设备声道模式。	输入

### 【返回值】

返回值	描述
0	成功
非 0	失败,其值见错误码。

### 【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

l在 AO 设备成功启用后再调用此接口。

l AO 设备工作在 I2S 模式时,支持设置声道模式,PCM 模式下不支持。



【举例】

》无。

### 10.4.2.20 AW\_MPI\_AO\_GetTrackMode

【目的】

获取 AO 设备声道模式。

【语法】

ERRORTYPE AW\_MPI\_AO\_GetTrackMode(AUDIO\_DEV AudioDevId, AUDIO\_TRACK\_MODE\_E \*penTrackMode);

【参数】

参数描述AudioDevIdAO 设备号。输入penTrackMode音频设备声道模式指针。输出

【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,其值见错误码。

【需求】

1头文件: mm\_comm\_alo.h、mm\_common.h

【注意】

l 在 AO 设备成功启用后再调用此接口。

1 AO 设备工作在 I2S 模式时,支持获取声道模式,PCM 模式下不支持。

【举例】

无。



### AW\_MPI\_AO\_SetVolume 10.4.2.21

【目的】

设置 AO 设备音量大小。

【语法】

ERRORTYPE AW\_MPI\_AO\_SetVolume(AUDIO\_DEV AudioDevId, int s32VolumeDb);

【参数】

参数	描述	<del>_</del>
AudioDevId	AO 设备号。	i A
s32VolumeDb	AO 设备音量大小。	WINT.
	A STATE OF THE STA	Six Control of the state of the
返回值	描述等	
0 非 0	成功 失败,其值见错误码。	

【返回值】

返回值	描述※
0	成功
非0	失败,其值见错误码。

【需求】

l 头文件: mm comm aio.h、mm common.h

【注意】

l在AO设备成功启用后再调用此接口。

l 音量大小 s32VolumeDb 参数的范围为 0~100 内的整数。

【举例】

无。

### 10.4.2.22 AW\_MPI\_AO\_GetVolume

【目的】

获取 AO 设备音量大小。

【语法】

ERRORTYPE AW\_MPI\_AO\_GetVolume(AUDIO\_DEV AudioDevId, int\*ps32VolumeDb);



### 【参数】

4X 3	A=X 3	
参数	描述	
AudioDevId ps32VolumeDb	AO 设备号。 AO 设备音量大小指针。	 输入 输出

【返回值】

返回值	描述		
0	 成功		
非0	失败,其值见错误码。	أأه	ing
ARIVA III CO	成功 失败,其值见错误码。 mmon.h	IE RALEBERT BERT STEEL S	illinde
mm_cor	nmon.h	© Halling The Control of the Control	
	Willey Live		
比接口。		An.	
	-14		
1	L		
O_SetI	Mute	oberii .	deril
-90	/	-9 <sub>6</sub> .	-90.

【需求】

l 头文件:mm\_comm\_aio.h、mm\_common.h

【注意】

l 在 AO 设备成功启用后再调用此接口。

【举例】

无。

### 10.4.2.23 AW\_MPI\_AO\_SetMute

【目的】

设置 AO 设备静音状态

【语法】

ERRORTYPE AW\_MPI\_AO\_SetMute(AUDIO\_DEV AudioDevId, BOOL bEnable, AUDIO FADE S \*pstFade);

【参数】

参数	描述	
AudioDevId	AO 设备号。	输入
bEnable	音频设备是否启用静音。	输入
pstFade	淡入淡出结构体指针。	输入

版权所有 © 珠海全志科技股份有限公司。保留一切权利

370



### 【返回值】

∠-X *		4-X
The Market of the Control of the Con	返回值	描述
E TO	0	成功
	非 0	失败,其值见错误码。

### 【需求】

l 头文件: mm comm aio.h、mm common.h

### 【注意】

1 在 AO 设备成功启用后再调用此接口。

1 调用此接口时,用户可以选择是否使用淡入淡出功能,如果不使用淡入淡出则将结构体指针赋为 空即可。(暂不支持 pstFade 参数设置)

l 静音标志 bEnable 为 l 时,设置音频主通道静音。bEnable 为 0 时,取消主通道静音,即恢 MINE 复声音,音量大小为设置静音前的值。

### 【举例】

无。

### 10.4.2.24 AW\_MPI\_AO\_GetMute

【目的】

获取 AO 设备静音状态。

### 【语法】

ERRORTYPE AW MPI AO GetMute(AUDIO DEV AudioDevId, BOOL pbEnable, AU-DIO\_FADE\_S pstFade);

### 【参数】

参数	描述	
AudioDevId	AO 设备号。	输入
pbEnable	音频设备静音状态指针。	输出
pstFade	淡入淡出结构体指针。	输出



返回值	描述	TA THE STATE OF TH
0	成功	HE NATIONAL PROPERTY OF THE PARTY OF THE PAR
非 0	失败,	其值见错误码。

l 头文件: mm comm aio.h、mm common.h

【注意】

l 在 AO 设备成功启用后再调用此接口。

Right High and the state of the 1 静音状态值为 1 时指示 AO 设备目前处于静音状态; 反之为正常工作状态。

AW\_MPI\_AO\_SetVqeAttr

【目的】

设置 AO 的声音质量增强功能相关属性。

【语法】

ERRORTYPE AW\_MPI\_AO\_SetVqeAttr(AUDIO\_DEV AudioDevId, AO\_CHN AoChn, AO VQE CONFIG S \*pstVqeConfig);

### 【参数】

参数	描述	
AudioDevId	AO 设备号。	输入
AoChn	AO 通道号。	输入
pstVqeConfig	音频输出声音质量增强配置结构体指针。	输入

返回值	描述	
0	成功	
# Qinde	失败,	其值见错误码。



l 头文件: mm comm aio.h、mm common.h

### 【注意】

1 启用声音质量增强功能前必须先设置相对应 AO 通道的声音质量增强功能相关属性。

1 设置 AO 的声音质量增强功能相关属性前,必须先使能对应的 AO 通道。

l 相同 AO 通道的声音质量增强功能不支持动态设置属性,重新设置 AO 通道的声音质量增强功能 相关属性时,需要先关闭 AO 通道的声音质量功能,再设置 AO 通道的声音质量增强功能相关属

1 AO 声音质量增强功能包括了环境噪声抑制功能,自动增益控制功能,高通滤波功能,均衡器功 能。在设置声音质量增强功能属性时,可通过配置相应的声音质量增强功能属性来选择使能其中 的部分功能。 

### 【举例】

无。

### 10.4.2.26 AW\_MPI\_AO\_GetVqeAttr

### 【目的】

获取 AO 的声音质量增强功能相关属性

### 【语法】

ERRORTYPE AW MPI AO GetVqeAttr(AUDIO DEV AudioDevId, AO CHN AoChn, AO VQE CONFIG S \*pstVqeConfig);

### 【参数】

参数	描述	**************************************
AudioDevId	AO 设备号。	输入
AoChn	AO 通道号。	
pstVqeConfig	音频输出声音质量增强配置结构体指针	

返回值	描述	ي.
Q	成功	ALIV
非 0	失败,	其值见错误码。



1头文件: mm\_comm\_aio.h、mm\_common.h

### 【注意】

1 获取声音质量增强功能相关属性前必须先设置相对应 AO 通道的声音质量增强功能相关属性。

### 【举例】

无。

### 10.4.2.27 AW\_MPI\_AO\_EnableVqe

### 【目的】

使能 AO 的声音质量增强功能。

### 【语法】

ERRORTYPE AW\_MPI\_AO\_EnableVqe(AUDIO\_DEV AudioDevId, AO\_CHN AoChn);

### 【参数】

参数	描述	
AudioDevId	AO 设备号。	输入
AoChn	AO 通道号。	输入

### 【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

### 【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

### 【注意】

- 1 启用声音质量增强功能前必须先启用相对应的 AO 通道。
- 1 多次使能相同 AO 通道的声音质量增强功能时,返回成功。
- 1 禁用 AO 通道后,如果重新启用 AO 通道,并使用声音质量增强功能,需调用此接口重新启用声



音质量增强功能。

【举例】

无。

### 10.4.2.28 AW\_MPI\_AO\_DisableVqe

【目的】

禁用 AO 的声音质量增强功能。

【语法】

ERRORTYPE AW\_MPI\_AO\_DisableVqe(AUDIO\_DEV AudioDevId, AO\_CHN AoChn);

【参数】

参数 描述 AudioDevId AO 设备号。 输入 AoChn AO 通道号。 输入

【返回值】

返回值 描述	
0 成功	
非 0、 失败,	其值见错误码。

### 【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

### 【注意】

- 1 不再使用 AO 声音质量增强功能时,应该调用此接口将其禁用。
- l 多次禁用相同 AO 通道的声音质量增强功能,返回成功。

### 【举例】

无。



### AW\_MPI\_AO\_SetStreamEof

### 【目的】

通知 AO 组件码流传递完毕标志。

### 【语法】

 $ERRORTYPE\ AW\_MPI\_AO\_SetStreamEof(AUDIO\_DEV\ AudioDevId,\ AO\_CHN\ AoChn,$ BOOL bEofFlag);

### 【参数】

参	数 inderi	描述	inderi iinderi
	udioDevIo Chn	d AO 设备号。 AO 通道号。	输入输入
bl	EofFlag	码流结束标志	EXO (S) # 11917 (
		A Line	
	返回值	描述	N
	0	成功	

【返回值】

返回值	描述	.111
0	成功	
非0	失败,	其值见错误码。

### 【需求】

l 头文件:mm\_comm\_aio.h、mm\_common.h

### 【注意】

1播放完毕后,app 必须设置码流结束标志到 AO 组件中

### 【举例】

无。

### 10.4.2.30 AW\_MPI\_AO\_SaveFile

### 【目的】

设置 AO 组件 pcm 数据的文件保存信息。

### 【语法】



ERRORTYPE AW\_MPI\_AO\_SaveFile(AUDIO\_DEV AudioDevId, AO\_CHN AoChn, AUDIO\_SAVE\_FILE\_INFO\_S \*pstSaveFileInfo);

### 【参数】

参数	描述	
AudioDevId	AO 设备号。	输入
AoChn	AO 通道号。	输入
pst Save File Info	文件保存信息指针。	

### 【返回值】

返回值 0 非 0	描述 成功 失败,其值见错误码。	
0	成功	
非 0	失败,其值见错误码。	
mm_con	nmon.h	
deri		

### 【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

【举例】

无。

### 10.4.2.31 AW\_MPI\_AO\_QueryFileStatus

【目的】

查询 AO 组件文件保存状态。

【语法】

ERRORTYPE AW\_MPI\_AO\_QueryFileStatus(AUDIO\_DEV AudioDevId, AO\_CHN AoChn, AUDIO SAVE FILE INFO S \*pstSaveFileInfo);

### 【参数】

参数	描述
AudioDevId	AO 设备号。 输入
AoChn	AO 通道号。 输入
pstSaveFileInfo	文件保存信息指针。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

377



【返回值】

17-11 H	返回值	描述
THE PARTY OF THE P	0	成功
	非 0	失败,其值见错误码。

【需求】

l 头文件: mm\_comm\_aio.h、mm\_common.h

【注意】

【举例】

10.4.3 音频编码

## an Reference of the state of th ${\bf 10.4.3.1 \quad AW\_MPI\_AENC\_CreateChn}$

【目的】

创建音频编码通道。

【语法】

ERRORTYPE AW MPI AENC CreateChn(AENC CHN AeChn, const AENC CHN ATTR S \*pstAttr);

【参数】

参数 描述 AeChn 音频编码通道号。 输入 音频编码属性指针。 输入 pstAttr

【返回值】

返回值 描述 成功 0 失败,其值见错误码

版权所有 © 珠海全志科技股份有限公司。保留一切权利

378

Real Property of the State of t



### 【需求】

1头文件: mm\_comm\_aenc.h、mm\_common.h

【注意】

无。

【举例】

无。

### 10.4.3.2 AW\_MPI\_AENC\_DestroyChn

【目的】

销毁音频编码通道。

【语法】

ERRORTYPE AW\_MPI\_AENC\_DestroyChn(AENC\_CHN AeChn);

【参数】

参数 描述 AeChn 音频编码通道号。 输入

【返回值】

112	/	112
返回值	描述	A STATE OF THE STA
·		X
0	成功	PACKETY'S
非 0	失败,	其值见错误码。
		/

【需求】

l 头文件: mm comm aenc.h、mm common.h

【注意】

- 1 通道未创建的情况下调用此接口会返回成功。
- 1 如果正在获取/释放码流或者发送帧时销毁该通道,则会返回失败,用户同步处理时需要注意。

【举例】

过。



### AW\_MPI\_AENC\_SendFrame 10.4.3.3

### 【目的】

发送音频编码音频帧。

### 【语法】

 $ERRORTYPE\ AW\_MPI\_AENC\_SendFrame (AENC\_CHN\ AeChn, const\ AUDIO\_FRAME\_INFO\_S$ \*pFrameInfo);

### 【参数】

 参数	6 <del>\$</del> 1	 描述	derii	nder.
AeChn	F F	音频编码通道号。	——— 输入	ARIVE IN
pFrameInf	o E	音频 pcm 帧结构体指针。	输入	-7.14 Mary
The state of the s		THE THE WAY		Religion of the state of the st
返回	回值	描述	6	
0	0	成功		

### 【返回值】

返回值	描述
0	成功
非 0	失败,其值见错误码。

### 【需求】

1头文件: mm comm aenc.h、mm common.h

### 【注意】

T发送 pcm 信息接口是非阻塞接口,如果音频 pcm 缓存区满,则直接返回失败。

1该接口用于用户主动发送音频帧进行编码,如果 AENC 通道已经通过系统绑定(AW MPI SYS Bind) 接口与 AI 绑定,不需要也不建议调此接口。

1 调用该接口发送音频编码音频帧时,必须先创建对应的编码通道。

### 【举例】

无。

### 10.4.3.4 AW MPI AENC GetStream

【目的】



获取编码后码流。

### 【语法】

ERRORTYPE AW MPI\_AENC\_GetStream(AENC\_CHN\_AeChn, AUDIO\_STREAM\_S \*pStream, int nMilliSec);

### 【参数】

 参数	描述			
AeChn	音频编码通道号。			 输入
pStream	音频编码属性指	計。		输出
nMilliSec	获取数据的超时间	时间:-1	表示阻塞模式,无数据时一直等待;0 表示非	输入
Me la linderi		据时则报	错返回;>0 表示阻塞 nMilliSec 毫秒,超时	lin
WIV.	则报错返回。	WILL STATE	ALV.	WILL.
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		<i>S.</i> .		
【返回值】				
	茶海	 返回值	描述《	
		0	成功	
		非 0	失败,其值见错误码。	

### 【返回值】

返回值	描述《
0	成功
非 0	失败,其值见错误码。

### 【需求】

l 头文件: mm comm aenc.h, mm common.h

### 【注意】

1 必须创建通道后才可能获取码流,否则直接返回失败,如果在获取码流过程中销毁通道则会立刻 返回失败。

l nMilliSec 的值必须大于等于-1,等于-1 时采用阻塞模式获取数据,等于 0 时采用非阻塞模式 获取数据,大于 0 时,阻塞 nMilliSec 毫秒后,没有数据则返回超时并报错。

### 【举例】

无。

### $10.4.3.5 \quad AW\_MPI\_AENC\_ReleaseStream$

【目的】

释放用户占用的编码码流。



### 【语法】

ERRORTYPE AW\_MPI\_AENC\_ReleaseStream(AENC\_CHN AeChn, AUDIO\_STREAM\_S \*pStream);

### 【参数】

参数	描述	
AeChn	音频编码通道号。	输入
pStream	音频编码属性指针。	输出

### 【返回值】

865	描述 描述	C. Hall Har Har Har Land Control of the Control of
返回值	描述	
0	成功	
非 0	失败,其值见错误码。	(S) Helphilit
n、mm_c	ommon.h	

### 【需求】

l 头文件: mm comm aenc.h、mm common.h

### 【注意】

- l 用户通过 AW\_MPI\_AENC\_GetStream 接口获得编码数据后,必须尽快将该编码数据再还给编码组件,以方便释放 stream buffer 中占用的空间。
- l 该接口配合 AW\_MPI\_AENC\_GetStream 一起使用,用于 none-tunnel 方式来保存编码后的数据,通常用于 nvr 模式,将采集到的 pcm 数据送编码,然后应用去拿编码数据,再去做文件封装。
- l AI 组件与 AEnc 组件通常用 tunnel 方式进行数据传递(需进行 bind 的操作)。AEnc 组件与 Muexer 组件通过 tunnel 方式数据传递时,mpp 组件内部做封装处理,保存为本地文件;通过 none-tunnel 方式数据传递时,即应用主动拿编码数据,然后自行处理,或通过网络传走,或自 行封装写卡。
- 1 码流最好能够在使用完之后立即释放,如果不及时释放,会导致编码过程阻塞。
- 1 释放的码流必须是从该通道获取的码流,不得对码流信息结构体进行任何修改,否则会导致码流 不能释放,使此码流 buffer 丢失,甚至导致程序异常。
- 1 释放码流时必须保证通道已经被创建,否则直接返回失败,如果在释放码流过程中销毁通道则会立刻返回失败。

### 【举例】

**光**。



### AW\_MPI\_AENC\_StartRecvPcm

【目的】

启动音频编码组件

【语法】

ERRORTYPE AW\_MPI\_AENC\_StartRecvPcm(AENC\_CHN AeChn);

【参数】

描述 参数 E Real House Hall the later of AeChn 音频编码通道号。

【返回值】

返回值 描述 成功 0 非 0 失败,其值见错误码。

【需求】

l 头文件: mm comm aenc.h、mm common.h

【注意】

1 该接口常用于绑定方式(AI、AENC 组件绑定)下,控制音频编码器的启动。如果 APP 主动发 送音频帧进行编码,建议使用 AW MPI AENC SendFrame 接口,如果 APP 本身不进行音频 数据管理,而希望系统主动处理,建议使用本接口。

【举例】

无。

### 10.4.3.7 AW\_MPI\_AENC\_StopRecvPcm

【目的】

关闭音频编码组件。

【语法】

ERRORTYPE AW\_MPI\_AENC\_StopRecvPcm(AENC\_CHN AeChn);



描述 参数 音频编码通道号。 AeChn 输入

【返回值】

返回值 描述 成功 0 非 0 失败,其值见错误码。

【需求】

l头文件: mm\_comm\_aenc.h、mm\_common.h

【注意】

无。

【举例】

无。

Chr 10.4.3.8 AW\_MPI\_AENC\_ResetChn

【目的】

复位 AEnc 通道。

【语法】

ERRORTYPE AW MPI\_AENC\_ResetChn(AENC\_CHN AeChn);

【参数】

描述 参数 音频编码通道号。 AeChn 输入

【返回值】

返回值 描述 成功



返回值 描述

失败,其值见错误码。 非 0

【需求】

l 头文件: mm comm aenc.h、mm common.h

【注意】

【举例】

无。

AW\_MPI\_AENC\_Query

【目的】

查询 AEnc 通道内部数据状态。

【语法】

AeC. AW MPI\_AENC\_Query(AENC\_CHN AeChn, AENC\_CHN\_STAT\_S ERRORTYPE \*pStat);

【参数】

参数 描述

AeChn 音频编码通道号。

pStat 来自 app 的状态信息指针。

【返回值】

返回值 描述 0 成功 非 0 失败,其值见错误码。

【需求】

l 头文件: mm comm aenc.h、mm common.h

【注意】

举例】



### 10.4.3.10 AW\_MPI\_AENC\_RegisterCallback

【目的】

向 AEnc 通道注册回调信息。

【语法】

ERRORTYPE AW MPI AENC RegisterCallback(AENC CHN AeChn, MPPCallback-Info \*pCallback);

	jinderi	indeni	C. H.
参数人	描述		A THE STATE OF THE
AeChn	音频编码通道号。	——— 输入	© Harry
pCallback	来自 app 的回调信息指针。		A STATE OF THE STA
返回	値 描述	1E	

【返回值】

返回值 描述 成功 0 非 0 失败,其值见错误码。

【需求】

l头文件: mm comm aench、mm common.h

【注意】

【举例】

无。

### ${\bf 10.4.3.11 \ AW\_MPI\_AENC\_SetChnAttr}$

【目的】

设置 AEnc 通道属性信息。

【语法】



ERRORTYPE AW\_MPI\_AENC\_SetChnAttr(AENC\_CHN AeChn, const AENC\_CHN\_ATTR\_S \*pAttr);

【参数】

参数 描述
AeChn 音频编码通道号。 输入
pAttr 来自 app 的通道属性信息指针。

【返回值】

返回值。描述

成功
非 0 失败,其值见错误码。

mm\_common.h

【需求】

l 头文件: mm\_comm\_aenc.h、mm\_common.h

【注意】

【举例】

无。

10.4.3.12 AW\_MPI\_AENC\_GetChnAttr

《目的】

获取 AEnc 通道属性信息。

【语法】

ERRORTYPE AW\_MPI\_AENC\_GetChnAttr(AENC\_CHN AeChn, const AENC\_CHN\_ATTR\_S \*pAttr);

【参数】

参数描述AeChn音频编码通道号。输入pAttr通道属性信息指针。

Am	NIMER

【返回值】

返回值 描述 0 成功 非 0 失败,其值见错误码。

【需求】

l 头文件: mm\_comm\_aenc.h、mm\_common.h

【注意】

无。

NC-10.4.3.13 AW\_MPI\_AENC\_GetHandle

【目的】

获取 AEnc 通道句柄。

【语法】

int AW\_MPI\_AENC\_GetHandle(AENC\_CHN AeChn);

描述

AeChn 音频编码通道号。 输入

【返回值】

返回值 描述 编码句柄号。 int

【需求】

1头文件: mm comm aenc.h、mm common.h

【注意】



无‰

【举例】

无。

### 10.4.4 音频解码

### ${\bf 10.4.4.1 \quad AW\_MPI\_ADEC\_CreateChn}$

【目的】

创建音频解码通道。

【语法】

ERRORTYPE AW\_MPI\_ADEC\_CreateChn(ADEC\_CHN AdChn, ADEC\_CHN\_ATTR\_S \*pstAttr);

【参数】

参数描述AdChn音频解码通道号。输入pstStream音频解码通道属性指针。输入

【返回值】》

 返回值 描述

 0 成功

 非 0 失败,其值见错误码。

【需求】

l 头文件: mm\_comm\_adec.h、mm\_common.h

【注意】

无。

【举例】

无。



### ${\bf AW\_MPI\_ADEC\_DestroyChn}$ 10.4.4.2

【目的】

销毁音频解码通道。

【语法】

ERRORTYPE AW\_MPI\_ADEC\_DestroyChn(ADEC\_CHN AdChn);

【参数】

描述 参数 C Report to the state of the st AdChn 音频解码通道号。

【返回值】

描述 返回值 0 成功 非 0 失败,其值见错误码。

【需求】

l 头文件: mm comm adec.h、mm common.h

【注意】

无。

【举例】

无。

10.4.4.3 AW\_MPI\_ADEC\_ResetChn

【目的】

复位 ADec 通道。

【语法】

ERRORTYPE AW\_MPI\_ADEC\_ResetChn(ADEC\_CHN AdChn);

【参数】



描述 参数

AdChn 音频解码通道号。 输入

【返回值】

返回值 描述 成功 0 非 0 失败,其值见错误码。

【需求】

l 头文件: mm\_comm\_adec.h、mm\_common.h

注意】

无。

【举例】

无。

# Record of the state of the stat 10.4.4.4 AW\_MPI\_ADEC\_RegisterCallback

【目的】

向 ADec 通道注册回调信息。

【语法】

ERRORTYPE AW\_MPI\_ADEC\_RegisterCallback(ADEC\_CHN ADecChn, MPPCallback-Info \*pCallback);

【参数】

参数	描述	
AdChn	音频解码通道号。	输入
pCallback	来自 app 层的回调信息指针。	

版权所有 © 珠海全志科技股份有限公司。保留一切权利



A 11/		A 11/
返回值	描述	
0	成功	A STATE OF THE STA
非 0	失败,	其值见错误码。

【需求】

l 头文件: mm\_comm\_adec.h、mm\_common.h

【注意】

无。

AW\_MPI\_ADEC\_SendStream

【目的】

向音频解码通道发送码流。

【语法】

(ADT ERRORTYPE AW\_MPI\_ADEC\_SendStream(ADEC\_CHN AdChn, const AUDIO\_STREAM\_S \*pstStream, BOOL bBlock);

参数	描述		-174		
AdChn	音频解码通道	道号。			输入
pstStream	音频码流指针	t			输入
bBlock	阻塞标识。T	RUE、阻塞。	FALSE:	非阻塞。	※ 输入

【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。



1头文件: mm\_comm\_adec.h。mm\_common.h

【注意】

无。

【举例】

无。

### 10.4.4.6 AW\_MPI\_ADEC\_ClearChnBuf

【目的】

清除 ADEC 通道中当前的音频数据缓存。

【语法】

ERRORTYPE AW\_MPI\_ADEC\_ClearChnBuf(ADEC\_CHN AdChn);

【参数】

参数 描述
AdChn 音频解码通道号。 输入

【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,其值见错误码。

【需求】

l 头文件: mm\_comm\_adec.h、mm\_common.h

【注意】

无。

【举例】

七。



### 10.4.4.7 AW\_MPI\_ADEC\_GetFrame

【目的】

获取解码后音频帧。

【语法】

ERRORTYPE AW\_MPI\_ADEC\_GetFrame(ADEC\_CHN AdChn, AUDIO\_FRAME\_INFO\_S \*pstFrmInfo, BOOL bBlock);

### 【参数】

)	参数	描述	ii			deril		indeni
	AdChn	音频解码通			A LIVE		输入	ik la lindeni
	pstFrmInfo	音频帧指针		×A	Ø.		输出	XA XA
	bBlock	阻塞标识。	TRUE:	阻塞。FA	LSE:	非阻塞。	输入	EX3.
			<b>₩</b>					
		返回值	描述		N			

【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,其值见错误码。

【需求】

l 头文件: mm\_comm\_adec.h、mm\_common.h

【注意】

无。

【举例】

无。

### ${\bf 10.4.4.8\ AW\_MPI\_ADEC\_ReleaseFrame}$

【目的】

释放从音频解码通道获取的音频帧。

【语法】



ERRORTYPE AW\_MPI\_ADEC\_ReleaseFrame(ADEC\_CHN AdChn, AUDIO\_FRAME\_INFO\_S \*pstFrmInfo);

【参数】

参数	描述	
AdChn	音频解码通道号。	输入
pstFrmInfo	获取的音频帧指针。	输入

【返回值】

			-	
返回值	描述		- Beril	deril
O Bliff	成功	112	alline Tallin	
10 非 0	失败,	其值见错误码。	X HIRLY	
x mm_co	√,×	其值见错误码。	All Report of the State of the	
der			deri	deni

【需求】

l 头文件: mm\_comm\_adec.h、mm\_common.h

【注意】

无。

【举例】

无。

10.4.4.9 AW\_MPI\_ADEC\_SetStreamEof

【目的】

向解码器发送码流结束标识符,并清除码流 buffer。

【语法】

ERRORTYPE AW\_MPI\_ADEC\_SetStreamEof(ADEC\_CHN AdChn, BOOL bEofFlag);

【参数】

 参数	描述		
AdChn	音频解码通道号。	dindeni	输入
ALL VI		WHO INTO	
A THE STATE OF THE	A STATE OF THE STA		
-		+51+1000 + 70 \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \ \	205



参数描述

b E of Flag

是否立即清除解码器内部的缓存数据。取值范围: FALSE: 延时清除。 输入不会立即清除解码器内部的缓存数据,解码会继续进行,直到剩余 buffer 不足一帧数据时进行清除操作。TRUE: 立即清除解码器内部缓存数据。

【返回值】

【需求】

l 头文件: mm\_comm\_adec.h、mm\_common.h

【注意】

无。

【举例】

无。

10.4.4.10 AW\_MPI\_ADEC\_StartRecvStream

【目的】

启动 ADEC 通道。

【语法】

ERRORTYPE AW MPI ADEC StartRecvStream(ADEC CHN AdChn);

【参数】

参数 描述 AdChn 音频解码通道号。 输入

【返回值】

版权所有 © 珠海全志科技股份有限公司。保留一切权利

396



返回值	描述	
0	成功	A STATE OF THE STA
非 0	失败,	其值见错误码。

【需求】

l 头文件: mm comm adec.h、mm common.h

【注意】

l 在启动该通道前,该 ADec 通道已经创建。否则返回错误码。

aream Re l 该 api 的调用,其内部实现中,该组件由 idle 状态过渡到 executing 状态。

### 10.4.4.11 AW\_MPI\_ADEC\_StopRecvStream

【目的】

停止 ADEC 通道。

【语法】

ERRORTYRE AW\_MPI\_ADEC\_StopRecvStream(ADEC\_CHN\_AdChn);

描述 AdChn 音频解码通道号。 输入

【返回值】

返回值 描述 0 成功 非 0 失败,其值见错误码。

【需求】

1头文件: mm comm adec.h. mm common.h



### 【注意】

1 在停止该通道前,需确保该 ADec 通道已经创建。否则返回错误码。

l 该 api 的调用,其内部实现中,该组件由 executing 状态过渡到 idle 状态。

【举例】

无。

### 10.4.4.12 AW\_MPI\_ADEC\_SetChnAttr

【目的】

设置 ADEC 通道属性。

【语法】

ERRORTYPE AW\_MPI\_ADEC\_SetChnAttr(ADEC\_CHN ADecChn, const-ADEC\_CHN\_ATTR\_S \*pAttr);

【参数】

参数	描述	MI	
AdChn	音频解码	通道号。	输入
pAttr	音频帧属	性。	输入

【仮回值】

返回值	描述	A TOP TO SERVICE A SERVICE
0	成功	A. A
非 0	失败,	其值见错误码。

【需求】

l 头文件: mm\_comm\_adec.h、mm\_common.h

【注意】

无。

【举例】

无。



### 10.4.4.13 AW\_MPI\_ADEC\_GetChnAttr

【目的】

获取 ADEC 通道属性。

【语法】

ERRORTYPE AW\_MPI\_ADEC\_GetChnAttr(ADEC\_CHN ADecChn, ADEC\_CHN\_ATTR\_S \*pAttr);

【参数】

Children Walter Hard Hard Committee of the Committee of t 描述 参数 AdChn 音频解码通道号。 pAttr 音频帧属性。 输出

【返回值】

描述 返回值 成功 非 0 其值见错误码。 失败,

【需求】

l 头文件: mm comm adec.h、mm common.h

【注意】

【举例】

无。

### 10.4.4.14 AW\_MPI\_ADEC\_Pause

【目的】

修改 ADEC 通道中组件内部状态。

【语法】

ERRORTYPE AW\_MPI\_ADEC\_Pause(ADEC\_CHN AdChn);



【参数】

参数 描述
AdChn 音频解码通道号。 输入

【返回值】

返回值	描述	
0	成功	
非 0	失败,	其值见错误码。

【需求】

l 头文件: mm\_comm\_adec h、mm\_common.h

【注意】

l 组件只有在 idle 或 executing 状态下才能变换到 pause 状态。常用于 player 在播放时暂停。

【举例】

无。

### 10.4.4.15 AW\_MPI\_ADEC\_Seek

【目的】

Player 跳播播放。

【语法】

ERRORTYPE AW MPI\_ADEC\_Seek(ADEC CHN AdChn);

【参数】

参数 描述		
AdChn	音频解码通道号。	——— 输入

【返回值】

 返回值 描述

 0 成功



返回值 描述

失败,其值见错误码。 非 0

【需求】

l 头文件: mm\_comm\_adec.h、mm common.h

【注意】

l播放器进行 seek 跳转播放时,该接口在内部实现为刷新音频解码器内部的 pcm 和 bitstream 缓冲管理器。

10.5 数据结构

10.5.1 音频输入输出

### 10.5.1.1 AIO\_ATTR\_S

【说明】

定义音频输入输出设备属性结构。

【定义】

typedef struct AIO ATTR S

AUDIO SAMPLE RATE E enSamplerate;

AUDIO\_BIT\_WIDTH\_E enBitwidth;

AIO MODE E enWorkmode;

AUDIO\_SOUND\_MODE\_E enSoundmode;

unsigned int u32EXFlag;

unsigned int u32FrmNum;

unsigned int u32PtNumPerFrm;

珠海全志科技股份有限公司。保留一切权利



unsigned int u32ChnCnt;

unsigned int u32ClkSely

unsigned int mPcmCardId;

int ai aec en;

int aec\_delay\_ms;

int ai ans en;

} AIO ATTR S;

### 【成员】

- In			2011	
成员名称	描述		其它说明	
enSamplerate	音频采样率。		支持。	
enBitwidth	音频采样精度。		支持。	142/10
enWorkmode	音频输入输出工作模式。		暂不支持。	17-17-1
enSoundmode ***	音频声道模式。		暂不支持。 🤻	* 183°
u32EXFlag	扩展标志。	-NE	暂不支持。	
u32FrmNum	缓存帧数目。	41 11	暂不支持。	
u32PtNumPerFrm	每帧采样点个数。		暂不支持。	
u32ChnCnt	支持的通道数目。	M	支持。	
u32ClkSel	配置 AI 设备 0 是否复	用 AO 设备 0 的帧同步时钟及位流时钟。	暂不支持。	
mPcmCardId	配置声卡类型。		支持。	
ai_aec_en	是否激活回声消除		支持	
aec_delay_ms	音频采集端采集的回声	,和回声播出时间的间隔。	如无特殊需求,	默认为
ai_ans_en	是否开启降噪处理	A STATE OF THE STA	支持	ļ
AlV.	ALV.		A!V	

### 【注意事项】

在设置 AIO 设备公共属性 (AW\_MPI\_AI/AO\_SetPubAttr) 时,其设备属性结构中三个 field 需填充正确的参数,这三个 field 分别为 enSamplerate、enBitwidth、u32ChnCnt。

在设置 AO 设备公共属性时,还需再添加输出的声卡类型 mPcmCardId: AudioCodec 和 Snd-Hdmi,其对应的输出接口,一种为 lineout 方式,另一种为 hdmi 输出方式。

### 【相关数据类型及接口】

 $AW\_MPI\_AI\_SetPubAttr_{\circ}$ 

AW\_MPI\_AO\_SetPubAttr。



# 10.5.1.2 AI\_CHN\_PARAM\_S

【说明】

定义通道参数结构体。

【定义】

typedef struct AI\_CHN\_PARAM\_S

{

unsigned int u32UsrFrmDepth;

} AI\_CHN\_PARAM\_S

【成员】

UsrFrmDepth; M_S		inderii inderii
M_S		其它说明
成员名称	描述	其它说明
u32UsrFrmDepth	音频帧缓存深度。	暂不支持。
ED]	LWIN	Ne 'a deri

【注意事项】

无。

【相关数据类型及接口】

无。

# 10.5.1.3 AUDIO\_FRAME\_S

【说明】

定义音频帧结构体。

【定义】

 $typedef\ struct\ AUDIO\_FRAME\_S$ 

{

AUDIO\_BIT\_WIDTH\_E enBitwidth;

AUDIO\_SOUND\_MODE\_E enSoundmode;

void \*mpAddr;

版权所有 © 珠海全志科技股份有限公司。保留一切权利

403



unsigned long long mTimeStamp;

unsigned int mSeq;

unsigned int mLen;

unsigned int mId;

} AUDIO\_FRAME\_S;

### 【成员】

成员名称	描述	其它说明
enBitwidth	音频采样精度。	 支持。
en Sound mode	音频声道模式。	支持。
mpAddr	音频帧数据虚拟地址。	支持。
mTimeStamp	音频帧时间戳。以 µs 为单位。	支持。
mSeq	音频帧序号。	暂不支持。
mLen	音频帧长度。以 byte 为单位。	支持。
mJd	音频帧 ID。《	支持。

### 【注意事项】

 $1\,\mathrm{mLen}$  (音频帧长度) 指  $1024\,\mathrm{\uparrow sample}$  采样的数据长度,在位宽为 16、单通道情况下,其值为 2048,在位宽为 16、双通道情况下,其值为 4096。

### 【相关数据类型及接口】

无。

### 10.5.1.4 AEC\_FRAME\_S

### 【说明】

定义音频回音消除参考帧信息结构体。

### 【定义】

 $typedef\ struct\ AEC\_FRAME\_S$ 

{

AUDIO\_FRAME\_S stRefFrame;

BOOL bValid;

BOOL bSysBind;



### } AEC\_FRAME\_S;

【成员】

成员名称	描述	其它说明
stRefFrame	回音消除参考帧结构体。	支持。
bValid	参考帧有效标志。	支持。
bSysBind	组件间是否采用绑定方式。	支持。

### 【注意事项】

无。

【相关数据类型及接口】

### 10.5.1.5 AUDIO\_AGC\_CONFIG\_S

【说明】

定义音频自动电平控制配置信息结构体。

【定义】

typedef struct AUDIO\_AGC\_CONFIG\_S

BOOL bUsrMode;

signed char s8TargetLevel;

signed char s8NoiseFloor;

signed char s8MaxGain;

signed char s8AdjustSpeed;

signed char s8ImproveSNR;

signed char s8UseHighPassFilt;

signed char s8OutputMode;

short s16NoiseSupSwitch;

int s32Reserved;

The state of the s



### } AUDIO AGC CONFIG S;

### 【成员】

成员名称	描述
------	----

是否采用用户模式: 0 自动模式, 1 用户模式, 默认为 0 bUsrMode

s8TargetLevel 目标电平。 s8NoiseFloor 噪声底线。 s8MaxGain 最大增益。 s8AdjustSpeed 调整速度。

s8ImproveSNR 提高信噪比开关。 s8UseHighPassFilt 打开高通滤波标志。

s8OutputMode 输出模式, 低于 NoiseFloor 的信号输出静音。范围: [0: 关闭, 1: 打开]

噪声抑制开关;范围 {0,1},0 表示关闭,1 表示开启。 s16NoiseSupSwitch

s32Reserved 保留。

### 【注意事项】

无。

【相关数据类型及接口】

无。

# S S 10.5.1.6 AI\_AEC\_CONFIG\_S

【说明】

定义音频回声抵消配置信息结构位

【定义】

typedef struct AFAEC CONFIG S

{

EC\_PARAMS\_T prms;

} AI AEC CONFIG S;

typedef struct

int enable aec;



AEC\_PARAMS\_T aec\_prms;

int enable\_bdc;

BDC\_PARAMS\_T bdc\_prms;

Int enable cdc;

DRC\_PARAMS\_T txdrc\_prms;

int enable rxdrc;

DRC\_PARAMS\_T rxdrc\_prms;

int enable\_txeq;

EQ PARAMS T txeq prms;

int enable\_rxeq;

EQ PARAMS T rxeg prms;

int enable ns;

NS PARAMS T ns prms;

int enable txfade;

} EC\_PARAMS\_T;

【成员】

Real Property of the state of t 其它说明 成员名称

参考《V316 音效库模块说明》。

【注意事项】

无。

【相关数据类型及接口】

无。

### AUDIO\_ANR\_CONFIG\_S 10.5.1.7

【说明】

定义音频环境噪声抑制功能配置信息结构体。

版权所有 © 珠海全志科技股份有限公司。保留一切权利



### 【定义】

```
typedef struct AUDIO ANR_CONFIG_S
{
```

BOOL bUsrMode;

short s16NrIntensity;

short s16NoiseDbThr;

signed char s8SpProSwitch;

int s32Reserved;

} AUDIO\_ANR\_CONFIG\_S;

### 【成员】

			(R) 4 (R)
成员名称	描述		Part His
bUsrMode	是否采用用户模式: (	) 自动模式,1 用户模式	じ,默认为 0
s16NrIntensity	降噪力度配置。	ANU	
s16NoiseDbThr	噪声门限配置。	1119	
s8SpProSwitch	音乐检测开关。		
s32Reserved	保留。		

### 【注意事项】

无。

【相关数据类型及接口】

无。

### 10.5.1.8 AUDIO\_HPF\_CONFIG\_S

### 【说明】

定义音频高通滤波功能配置信息结构体。

### 【定义】

typedef struct AUDIO\_HPF\_CONFIG\_S

A THE VERTILIAN SELECTION OF THE PROPERTY OF T



BOOL bUsrMode;

AUDIO\_HPF\_FREQ\_E\_enHpfFreq;

} AUDIO\_HPF\_CONFIG\_S;

### 【成员】

成员名称	描述	其它说明
bUsrMode	是否采用用户模式: 0 自动模式,1 用户模式,默认为 0	暂不支持。
enHpfFreq	高通滤波截止频率选择。	暂不支持。

【注意事项】

【相关数据类型及接口】

无。

AI\_RNR\_CONFIG\_S 10.5.1.9

【说明】

定义音频输入高保真噪声抑制功能配置信息结构体。

【定义】

typedef struct AI\_RNR\_CONFIG\_S

int sMaxNoiseSuppression;

int sOverlapPercent;

int sNonstat;

} AI\_RNR\_CONFIG\_S;

【成员】

其它说明 成员名称 暂不支持。 sMaxNoiseSuppression 暂不支持。 sOverlapPercent sNonstat 暂不支持。

【注意事项】

无。

【相关数据类型及接口】

无。

### 10.5.1.10 AUDIO\_EQ\_CONFIG\_S

【说明】

定义音频均衡器功能配置信息结构体。

【定义】

typedef struct AUDIO\_EQ\_CONFIG\_S

{

short s16GaindB[VQE EQ BAND NUM]

int s32Reserved;

} AUDIO\_EQ\_CONFIG\_S;

【成员】

成员名称 描述
s8GaindB EQ 频段增益调节。ndering s32Reserved 保留。

【注意事项】

无。

【相关数据类型及接口】

无。

### 10.5.1.11 AI\_VQE\_CONFIG\_S

【说明】

定义音频输入声音质量增强配置信息结构体。

版权所有 © 珠海全志科技股份有限公司。保留一切权利

410



### 【定义】

typedef struct AI\_VQE\_CONFIG\_S

int bHpfOpen;

int bAecOpen;

int bAnrOpen;

int bRnrOpen;

int bAgcOpen;

int bEqOpen;

int bDrcOpen;

int s32WorkSampleRate;

int s32FrameSample;

VQE WORKSTATE E enWorkstate;

AUDIO HPF CONFIG S stHpfCfg;

AI\_AEC\_CONFIG\_S stAecCfg;

AUDIO ANR CONFIG S stAnrCfg;

AI\_RNR\_CONFIG\_S stRnrCfg;

AUDIO AGC CONFIG S stAgcCfg;

AUDIO EQ CONFIG S stEqCfg;

AI\_DRC\_CONFIG\_S stDrcCfg;

} AI VQE CONFIG S;

【成员】

成员名称 描述

bHpfOpen 高通滤波功能是否使能标志。
bAecOpen 回声抵消功能是否使能标志。
bAnrOpen 环境噪声抑制功能是否使能标志。
bRnrOpen 高保真噪声抑制功能是否使能标志。
bAgcOpen 自动电平控制功能是否使能标志
bEqOpen 均衡器功能是否使能标志

THE REPORT OF THE PARTY OF THE

iden'i

放好所有 © 珠海全志科技股份有限公司。保留一切权利



成员名称	描述
bDrcOpen	录音宽动态功能是否使能标志
s32WorkSampleRate	工作采样频率。
s32FrameSample	VQE 的帧长,即采样点数目。
enWorkstate	工作模式。
stHpfCfg	高通滤波功能相关配置信息。
stAecCfg	回声抵消功能相关配置信息。
stAnrCfg	环境噪声抑止功能相关配置信息。
stRnrCfg	高保真噪声抑制功能相关配置信息。
stAgcCfg	自动电平控制相关配置信息。
stEqCfg	均衡器相关配置信息。
stDrcCfg	录音宽动态相关配置信息。

### 【定义】

typedef struct AO\_VQE\_CONFIG\_S

int bHpfOpen;

int bAnrOpen;

int bAgcOpen;

int bEqOpen;

int bGainOpen;

int s32WorkSampleRate;

int s32FrameSample;

10.5.1.12 AO\_VQE\_CONFIG\_S

【说明】

② YEMPHARMAN

③ YEMPHARMAN

② YEMPHARMAN

③ YEMPHARMAN

③ YEMPHARMAN

③ YEMPHARMAN

② YEMPHARMAN

② YEMPHARMAN

③ YEMPHARMAN

② YEMPHARMAN

② YEMPHARMAN

③ YEMPHARMAN

② YEMPHARMAN

② YEMPHARMAN

② YEMPHARMAN

② YEMPHARMAN

② YEMPHARMAN

③ YEMPHARMAN

② YEMPHAR



VQE\_WORKSTATE\_E enWorkstate;

AUDIO\_HPF\_CONFIG S stHpfCfg;

AUDIO\_ANR\_CONFIG\_S stAnrCfg;

AUDIO\_AGC\_CONFIG\_S stAgcCfg;

AUDIO\_EQ\_CONFIG\_S stEqCfg;

AUDIO GAIN CONFIG S stGainCfg;

} AO\_VQE\_CONFIG\_S;

【成员】

成员名称	描述
bHpfOpen	高通滤波功能是否使能标志。
bAnrOpen	降噪功能是否使能标志。
bAgeOpen	自动电平控制功能是否使能标志
bEqOpen	均衡器功能是否使能标志
bGainOpen	增益功能是否使能标志
s 32 Work Sample Rate	工作采样频率。
s32FrameSample	VQE 的帧长,即采样点数目。
enWorkstate	工作模式。
stHpfCfg	高通滤波功能相关配置信息。
stAnrCfg	环境噪声抑止功能相关配置信息。
stAgcCfg	自动电平控制相关配置信息。
stEqCfg	均衡器相关配置信息。
stGainCfg	增益相关配置信息。

【注意事项】

无。

【相关数据类型及接口】

无。

### **10.5.1.13 AUDIO\_STREAM\_S**

【说明】

定义音频码流结构体。

定义】



```
typedef struct AUDIO_STREAM_S

{
   unsigned char *pStream;
   unsigned int mLen;
   unsigned long long mTimeStamp;
   unsigned int mId;
} AUDIO_STREAM_S;
```

【成员】

成员名称 描述
pStream 音频码流数据指针。
mLen 音频码流长度。单位为 byte。
mTimeStamp 音频码流时间戳。
mId 音频码流序号。

【注意事项】

无。

【相关数据类型及接口】

无。

## 10.5.1.14 AO\_CHN\_STATE\_S

【说明】

音频输出通道的数据缓存状态结构体。

【定义】

 $typedef\ struct\ AO\_CHN\_STATE\_S$ 

{

unsigned int u32ChnTotalNum;

unsigned int u32ChnFreeNum;

unsigned int u32ChnBusyNum;

版权所有 © 珠海全志科技股份有限公司。保留一切权利

414



### } AO\_CHN\_STATE\_S;

【成员】

描述
输出通道总的缓存块数。
空闲缓存块数。
被占用缓存块数。

【注意事项】

无。

【相关数据类型及接口】

# 10.5.1.15 AUDIO\_FADE\_S

【说明】

音频输出设备淡入淡出配置结构体。

【定义】

typedef struct AUDIO\_FADE\_S

BOOL bFade;

AUDIO\_FADE\_RATE\_E\_enFadeInRate;

AUDIO\_FADE\_RATE\_E enFadeOutRate;

} AUDIO\_FADE\_S;

【成员】

成员名称	描述
bFade	是否开启淡入淡出功能。
enFadeInRate	音频输出设备音量淡入速度。
enFadeOutRate	音频输出设备音量淡出速度。

【注意事项】

版权所有 © 珠海全志科技股份有限公司。保留一切权利

无必

【相关数据类型及接口】

无。

### 10.5.1.16 AUDIO\_SAMPLE\_RATE\_E

【说明】

定义音频采样率。

【定义】

typedef enum AUDIO SAMPLE RATE E

A.

AUDIO\_SAMPLE\_RATE\_8000 = 8000, /\* 8K samplerate\*/

AUDIO SAMPLE RATE 12000 = 12000, /\* 12K samplerate\*/

AUDIO SAMPLE RATE 11025 = 11025, /\* 11.025K samplerate\*/

AUDIO\_SAMPLE\_RATE\_16000 = 16000, /\* 16K samplerate\*/

AUDIO SAMPLE RATE 22050 = 22050, /\* 22.050K samplerate\*/

AUDIO SAMPLE RATE 24000 = 24000, /\* 24K samplerate\*/

AUDIO SAMPLE RATE 32000 = 32000, /\* 32K samplerate\*/

AUDIO SAMPLE RATE 44100 = 44100, /\* 44.1K samplerate\*/

AUDIO SAMPLE RATE 48000 = 48000, /\* 48K samplerate\*/

} AUDIO SAMPLE RATE E;

【成员】

成员名称	描述
AUDIO_SAMPLE_RATE_8000	8kHz 采样率
AUDIO_SAMPLE_RATE_12000	12kHz 采样率
AUDIO_SAMPLE_RATE_11025	11.025kHz 采样率
AUDIO_SAMPLE_RATE_16000	16kHz 采样率
AUDIO_SAMPLE_RATE_22050	22.05kHz 采样率
AUDIO_SAMPLE_RATE_24000	24kHz 采样率
AUDIO_SAMPLE_RATE_32000	32kHz 采样率
/=X ?	/CY /



成员名称

AUDIO\_SAMPLE\_RATE\_44100 44.1kHz 采样率 AUDIO\_SAMPLE\_RATE\_48000 48kHz 采样率

描述

【注意事项】

无。

【相关数据类型及接口】

无。

### C Branch Little 10.5.1.17 AUDIO\_BIT\_WIDTH\_E

【说明】

定义音频采样精度。

【定义】

typedef enum AUDIO\_BIT\_WIDTH\_E

{

AUDIO BIT WIDTH 8 = 0, /\* 8bit width \*/

AUDIO\_BIT\_WIDTH\_16 = 1, /\* 16bit width\*/

AUDIO\_BIT\_WIDTH\_24 = 2, /\* 24bit width\*/

AUDIO\_BIT\_WIDTH\_32 = 3, /\* 32bit width\*/

} AUDIO\_BIT\_WIDTH\_E;

【成员】

成员名称	描述
AUDIO_BIT_WIDTH_8	采样精度为 8bit 位宽。
AUDIO_BIT_WIDTH_16	采样精度为 16bit 位宽。
AUDIO_BIT_WIDTH_24	采样精度为 24bit 位宽。
AUDIO_BIT_WIDTH_32	采样精度为 32bit 位宽。

【注意事项】



【相关数据类型及接口】

无。

### 10.5.1.18 AIO\_MODE\_E

【说明】

定义音频设备工作模式信息结构体。

【定义】

typedef enum AIO\_MODE\_E

{

AIO MODE I2S MASTER = 0, /\* AIO I2S master mode \*/

AIO\_MODE\_I2S\_SLAVE, /\* AIO I2S slave mode \*/

AIO\_MODE\_PCM\_SLAVE\_STD, /\* AIO PCM slave standard mode \*/

AIO MODE PCM SLAVE NSTD, /\* AIO PCM slave non-standard mode \*/

AIO\_MODE\_PCM\_MASTER\_STD, /\* AIO PCM master standard mode \*/

AIO MODE PCM MASTER NSTD, /\* AIO PCM master non-standard mode \*/

AIO\_MODE\_BUTT

} AIO MODE E;

【成员】

成员名称	描述	其它说明
AIO_MODE_12S_MASTER	I2S 主模式	暂不支持。
AIO_MODE_I2S_SLAVE	I2S 从模式	暂不支持。
AIO_MODE_PCM_SLAVE_STD	PCM 从模式(标准协议)	暂不支持。
AIO_MODE_PCM_SLAVE_NSTD	PCM 从模式(自定义协i	义) 暂不支持。
AIO_MODE_PCM_MASTER_STD	PCM 主模式(标准协议)	暂不支持。
AIO_MODE_PCM_MASTER_NSTD	PCM 主模式(自定义协i	义) 暂不支持。

【注意事项】

无。

【相关数据类型及接口】

### 10.5.1.19 AIQ\_SOUND\_MODE\_E

【说明】

定义音频声道模式结构体。

【定义】

typedef enum AIO SOUND MODE E

{

AUDIO\_SOUND\_MODE\_MONO =0, /mono/

AUDIO SOUND MODE STEREO =1, /stereo/

} AUDIO\_SOUND\_MODE\_E;

【成员】

成员名称 描述 AUDIO SOUND MODE MONO 单声道 AUDIO SOUND MODE STEREO

【注意事项】

【相关数据类型及接口】

无。

### 10.5.1.20 AUDIO\_HPF\_FREQ\_E

【说明】

定义音频 HPE 频率结构体。

【定义】

typedef enum AUDIO\_HPF\_FREQ\_E

双声道

Real Property of the State of t



```
AUDIO_HPF_FREQ_80 = 80, 7* 80Hz */
```

AUDIO\_HPF\_FREQ\_120 = 120, /\* 120Hz \*/

AUDIO\_HPF\_FREQ\_150 = 150, /\* 150Hz \*/

} AUDIO HPF FREQ E;

【成员】

	成员名称	描述	
	AUDIO_HPF_FREQ_80	 截止频率为 80Hz。	
	AUDIO_HPF_FREQ_120		
inderi'	AUDIO_HPF_FREQ_150	0 截止频率为 150Hz。	
《注意事项】	A CANAL TO THE PARTY OF THE PAR	A CARLON CONTRACTOR OF THE PARTY OF THE PART	O. F. Haller
【/工总事项】	all the second	A Military	(Q) ORINGE
无。			
【相关数据类型及接			W. The
无。		11N4	
40 7 4 94 4 96	· · · · · · · · · · · · · · · · · · ·	MI	
10.5.1.21 AQ	E_WORKSTATE_E		
【说明】			
定义工作模式结构体。	a like Valideri	A Maria Caria Cari	
ALIV .	WIN.	ARIV.	
【定义】	A THINK	XA TOPSY	Ž.

typedef enum VQE\_WORKSTATE\_E

VQE WORKSTATE COMMON = 0,

VQE WORKSTATE MUSIC = 1,

 $VQE_WORKSTATE_NOISY = 2$ ,

} VQE\_WORKSTATE\_E;

【成员】



成员名称	描述
VQE_WORKSTATE_COMMON	一般模式。
VQE_WORKSTATE_MUSIC	音乐模式。
VQE_WORKSTATE_NOISY	噪声模式。

【注意事项】

无。

【相关数据类型及接口】

无。

AUDIO TRACK BOTH LEFT = 15,000

 $AUDIO_TRACK_BOTH_RIGHT = 2$ ,

AUDIO TRACK EXCHANGE = 3,

AUDIO TRACK MIX = 4,

 $AUDIO_TRACK_LEFT_MUTE = 5$ ,

AUDIO TRACK RIGHT MUTE = 6,

 $AUDIO_TRACK_BOTH_MUTE = 7$ ,

} AUDIO\_TRACK\_MODE E;

【成员】

に定义】

typedef enum AUDIO\_TRACK\_MODE\_E

{
AUDIO\_TRACK\_NORMAL = 0



	成员名称	描述	A CONTRACTOR OF THE PARTY OF TH		其它说明
VAN A	AUDIO_TRACK_NORMAL	正常模式,不做处理	1		支持。
r	AUDIO_TRACK_BOTH_LEFT	两个声道全部为左声	道声音	Ž.	暂不支持
	AUDIO_TRACK_BOTH_RIGHT	两个声道全部为右声	道声音		暂不支持
	AUDIO_TRACK_EXCHANGE	左右声道数据互换,	左声道为右声道声音,	右声道为左声道声音	暂不支持
	AUDIO_TRACK_MIX	左右两个声道输出为	」左右声道相加(混音)		暂不支持
	AUDIO_TRACK_LEFT_MUTE	左声道静音,右声道	<b>插放原右声道声音</b>		暂不支持
	AUDIO_TRACK_RIGHT_MUTE	右声道静音,左声道	<b>插放原左声道声音</b>		暂不支持
	AUDIO_TRACK_BOTH_MUTE	左右声道均静音			暂不支持

### 【注意事项】

【相关数据类型及接口】

无。

## A Report of the state of the st 10.5.1.23 AUDIO\_FADE\_RATE\_E

【说明】

定义音频输出设备淡入淡出速度类型。

【定义】

typedef enum AUDIO\_FADE\_RATE\_E

 $AUDIO_FADE_RATE_1 = 0$ ,

 $AUDIO_FADE_RATE_2 = 1$ ,

 $AUDIO_FADE_RATE_4 = 2$ ,

 $AUDIO_FADE_RATE_8 = 3$ ,

 $AUDIO_FADE_RATE_16 = 4$ ,

 $AUDIO_FADE_RATE_32 = 5$ ,

AUDIO FADE RATE 64 = 6,

AUDIO\_FADE\_RATE\_128 = 7

AUDIO\_FADE\_RATE\_E



### 【成员】

成员名称	描述	
AUDIO_FADE_RATE_1	1 个采样点改变一次	
AUDIO_FADE_RATE_2	2 个采样点改变一次	XV.
AUDIO_FADE_RATE_4	4 个采样点改变一次	
AUDIO_FADE_RATE_8	8 个采样点改变一次	
AUDIO_FADE_RATE_16	16 个采样点改变一次	
AUDIO_FADE_RATE_32	32 个采样点改变一次	
AUDIO_FADE_RATE_64	64 个采样点改变一次	
AUDIO_FADE_RATE_128	128 个采样点改变一次	
26_BPS_E	INNE IN E	Referrible to the state of the
协议速率。	<u>,</u>	·
deri	deri	deri

### 【注意事项】

无。

【相关数据类型及接口】

无。

### 10.5.1.24 G726\_BPS\_E

### 【说明】

定义 G.726 编解码协议速率。

【定义】

typedef enum G726 BPS E

G726\_16K = 0, /\* G726 16kbps, see RFC3551.txt 4.5.4 G726-16 \*/

G726\_24K, /\* G726 24kbps, see RFC3551.txt 4.5.4 G726-24 \*/

G726 32K, /\* G726 32kbps, see RFC3551.txt 4.5.4 G726-32 \*/

G726\_40K, /\* G726 40kbps, see RFC3551.txt 4.5.4 G726-40 \*/

MEDIA\_G726\_16K, /\* G726 16kbps for ASF ... \*/

MEDIA G726 24K, /\* G726 24kbps for ASF ... \*/

MEDIA\_G726\_32K, /\* G726 32kbps for ASF ... \*/

MEDIA\_G726\_40K, /\* G726 40kbps for ASF ... \*,

版权所有 © 珠海全志科技股份有限公司。保留一切权利

423



### } G726\_BPS\_E;

### 【成员】

	~/ Y~	-/ YE	~/ Y~	
	成员名称	描述	其它说明	
	G726_16K	16kbit/s G.726 <sub>°</sub>	暂不支持。	
	G726_24K	24kbit/s G.726 <sub>°</sub>	暂不支持。	
	G726_32K	32kbit/s G.726。	暂不支持。	
	G726_40K	40kbit/s G.726°	暂不支持。	
	MEDIA_G726_16K	G726 16kbit/s for ASF $_{\circ}$	暂不支持。	
	MEDIA_G726_24K	G726 24kbit/s for ASF $_{\circ}$	暂不支持。	
•	MEDIA_G726_32K	G726 32kbit/s for ASF <sub>o</sub>	暂不支持。	
Hinden	MEDIA_G726_40K	G726 40kbit/s for ASF。	暂不支持。	Hinden
WILL.	SE IV	A LITTER OF THE PARTY OF THE PA		ALIVE TO SERVICE TO SE
【注意事项】	OR FEET HOLD	A STATE OF THE STA	Q , W	A CONTRACTOR OF THE PARTY OF TH
无。				
【相关数据类型	╗┸╬╙ ╗┸╬╙	**************************************	供源	
【作人奴/6天日	E/XI女口』			
无。		41 N		
		MIN		
10.5.1.25	ADPCM_TYPE_I			
【说明】				
Luc-911			i i	deri
定义 ADPCM	编解码协议类型。	a lind		<b>Alling</b>
【定义】	THE LIVE	E IV		RAIV
DEX.	EXPARAMENT OF THE PROPERTY OF			X
1200/1	120/3	122/1	17.3/	-

typedef enum ADPCM TYPE E

ADPCM TYPE DVI4 = 0,

ADPCM TYPE IMA,

} ADPCM\_TYPE\_E;

【成员】

成员名称	。描述 <sub>mderi</sub>	其它说明
ADPCM_TYPE_DVI4	32kbit/s ADPCM(DVI4)。	暂不支持。
ADPCM_TYPE_IMA	32kbit/s ADPCM(IMA)。	暂不支持。

版权所有 © 珠海全志科技股份有限公司。保留-

### 【注意事项】

【相关数据类型及接口】

无。

### 10.5.2 音频编码

### 10.5.2.1 AENC\_CHN\_ATTR\_S

【说明】

定义音频编码通道属性结构体。

【定义】

Still A Still Beet of the Stil typedef struct AENC CHN ATTR S

{

AENC ATTR S AeAttr;

}AENC\_CHN\_ATTR\_S;

typedef struct AENC\_ATTR\_S

{

PAYLOAD\_TYPE\_E Type;

int sampleRate;

int channels;

int bitRate;

int bitsPerSample;

int attachAACHeader;

}AENC\_ATTR\_S;

【成员】

成员名称 描述 其它说明 Type 音频解码协议类型。



	\$11/	
成员名称	描述	其它说明
sampleRate	音频数据采样率。	支持。
channels	通道数量 (单或双通道)。	支持。
bitRate	编码码率。	类支持。
bitsPerSample	采样位宽。	支持。
attachAACHeader	编码为 AAC 的数据是否添加文件头信息。	支持。

### 【注意事项】

l编码通道属性至少需填充四个 field: Type、sampleRate、channels、bitsPerSample。

1 在 AAC、编码时,当通过网络传输希望每帧音频码流带头信息时》其中需设置 attachAAC。 Header 为 1, 当直接送 muxer 而无需码流头信息时,设置 attachAACHeader 为 0 即可。

1 在 G726 编码时,需设置 bitRate 参数 (16k/24k/32k/40k),用于调整 G726 编码输出数据 宽度 (2bit/3bit/4bit/5bit)。如果设置为 0,编码数据宽度自动调整为 2bit。

## 

定义音频解码通道属性结构体

### 【定义】

typedef struct ADEC CHN ATTR S

{

PAYLOAD TYPE E mType;

int sampleRate;

int channels;

int bitRate; // not use

int bitsPerSample;

int attachAACHeader;



### }ADEC\_CHN\_ATTR\_S;

### 【成员】

成员名称	描述	其它说明
mType	音频解码协议类型。	支持。
sampleRate	音频数据采样率。	支持。
channels	通道数量 (单或双通道)。	支持。
bitsPerSample	采样位宽。	支持。
attachAACHeader	用于码流类型为 AAC 的数据。	支持。

【注意事项】 无。 【相关数据类 无。 10.5.4 节	《型及接口】  《表情》  《表情》	式要求
编码器类型	输入 pcm 格式	输出
aac mp3 adpcm g711a/u g726 pcma	8k~48k 采样率, 单/双通道,16 位宽 8k~48k 采样率, 单/双通道,16 位宽 8k 采样率, 单通道,16 位宽 8k 采样率, 单通道,16 位宽 8k 采样率, 单通道,16 位宽 8k 采样率, 单通道,16 位宽	压缩率为 10 左右。8k/单通道下输出约 16kbps。 16k/24k/128kbps 由应用控制。默认 16kbps。 压缩率为 4。输出码率 32kbps。 压缩率为 2。输出码率 64kbps。 压缩 16->2/3/4/5bit,分别对应输出 16k/24k/32k/40kbpr。 不进行压缩。根据输入原样输出数据。

解码器类型	输入码流格式	输出
aac	8k~192k 采样率, 单/双通道,16 位宽	根据输入调整输出。
mp3	8k~192k 采样率, 单/双通道,16 位宽	根据输入调整输出。
adpcm	8k 采样率, 单通道,16 位宽	8k 采样率、16 位宽。
g711a/u	8k 采样率, 单通道,16 位宽	8k 采样率、16 位宽。
g726	8k 采样率, 单通道,16 位宽	8k 采样率、16 位宽。
pcma	8k~192k 采样率,单/双通道,16 位宽	根据输入调整输出。

不同编码器对输入数据格式要求不同。acc 和 mp3 编码器对输入数据格式要求只需保证 16 位



宽、对通道数量、采样率没有要求。adpcm、g711a/u、g726 编码器则对输入 pcm 数据格式 有较多限制,必须同时满足。8k 采样率、单通道、16 位宽的要求 (编码标准要求)。pcma 编码 为非压缩编码器,在内部实现为直接将输入数据送输出。在做 ai-aenc 开发时,如果需要输出 adpcm、g711a/u、g726 格式的数据,那么在 AI 设备属性需设置 8k 采样率/单通道/16 位宽 的参数。

目前, AIO 设备的数据位宽 (bitwidth) 只支持 16 位, 8、24、32 位宽大小暂不支持。

### 10.6 错误码

### 10.6.1 音频输入错误码

•	1。音频输入	错误码 main de la company de la	Minderi
_		N. T.	BID.
	错误码	宏定义	描述
	0xA0158001	ERR_AI_INVALID_DEVID	音频输入设备号无效
	0xA0158002	ERR_AI_INVALID_CHNID	音频输入通道号无效
	0xA0158003	ERR_AI_ILLEGAL_PARAM	音频输入参数设置无效
	0xA0158006	ERR_AI_NULL_PTR	输入参数空指针错误
	0xA0158007	ERR_AI_NOT_CONFIG	音频输入设备属性未设置
	0xA0158008	ERR_AI_NOT_SUPPORT	操作不支持
	0xA0158009	ERR_AI_NOT_PERM	操作不允许
	0xA0158005	ERR_AI_NOT_ENABLED	音频输入设备或通道没有使能
	0xA015800C	ERR_AI_NOMEM	分配内存失败
	0xA015800D	ERR_AI_NOBUF	音频输入缓存不足
	0xA015800E	ERR_AI_BUF_EMPTY	音频输入缓存为空
⟨	0xA015800F	ERR_AI_BUF_FULL	音频输入缓存为满
	0xA0158010	ERR_AI_SYS_NOTREADY	音频输入系统未初始化
	0xA0158012	ERR_AI_BUSY	音频输入系统忙
	106	-/A	76/

### 10.6.2 音频输出错误码

错误码	宏定义	
0xA0168001	ERR_AO_INVALID_DEVID	音频输出设备号无效
0xA0168002	ERR_AO_INVALID_CHNID	音频输出通道号无效
0xA0168003	ERR_AO_ILLEGAL_PARAM	音频输出参数设置无效
0xA0168006	ERR_AO_NULL_PTR	音频输出参数空指针错误
0xA0168007	ERR_AO_NOT_CONFIG	音频输出设备属性未设置
0xA0168008	ERR_AO_NOT_SUPPORT	操作不支持
0xA0168009	ERR_AO_NOT_PERM	操作不允许
		No.



	A11/	
错误码	宏定义	描述
0xA0168005	ERR_AO_NOT_ENABLED	音频输出设备或通道没有使能
0xA016800€	ERR_AO_NOMEM	系统内存不足
0xA016800D	ERR_AO_NOBUF	音频输出缓存不足
0xA016800E	ERR_AO_BUF_EMPTY	音频输出缓存为空
0xA016800F	ERR_AO_BUF_FULL	音频输出缓存为满
0xA0168010	ERR_AO_SYS_NOTREADY	音频输出系统未初始化
0xA0168012	ERR_AO_BUSY	音频输出系统忙

### 10.6.3 音频编码错误码

3 音频编码银	错误码		
inden	Willing St.	Minden Alinde	
错误码	宏定义	描述	
0xA0178001	ERR_AENC_INVALID_DEV	ID 音频编码设备号无效	
0xA0178002	ERR_AENC_INVALID_CHN	IID 音频编码通道号无效	
0xA0178003	ERR_AENC_ILLEGAL_PARA	AM 音频编码参数设置无效	
0xA0178004	ERR_AENC_EXIST	音频编码通道已经创建	
0xA0178005	ERR_AENC_UNEXIST	音频编码通道未创建	
0xA0178006	ERR_AENC_NULL_PTR	输入参数空指针错误	
0xA0178007	ERR_AENC_NOT_CONFIG	编码通道未配置	
0xA0178008	ERR_AENC_NOT_SUPPOR	T 操作不被支持	
0xA0178009	ERR_AENC_NOT_PERM	操作不允许	
0xA017800C	ERR_AENC_NOMEM	系统内存不足	
0xA017800D	ERR_AENC_NOBUF	编码通道缓存分配失败	i
0xA017800E	ERR_AENC_BUF_EMPTY	编码通道缓存空	
0xA017800F	ERR_AENC_BUF_FULL	编码通道缓存满	
0xA0178010	ERR_AENC_SYS_NOTREAL	DY 系统没有初始化	
	ERR_AENC_ENCODER_ER	R 音频编码数据错误	

### A THE THE PARTY OF 10.6.4 音频解码错误码

	错误码	宏定义	描述
	0xA0188001	ERR_ADEC_INVALID_DEVID	音频解码设备号无效
	0xA0188002	ERR_ADEC_INVALID_CHNID	音频解码通道号无效
	0xA0188003	ERR_ADEC_ILLEGAL_PARAM	音频解码参数设置无效
	0xA0188004	ERR_ADEC_EXIST	音频解码通道已经创建
	0xA0188005	ERR_ADEC_UNEXIST	音频解码通道未创建
	0xA0188006	ERR_ADEC_NULL_PTR	输入参数空指针错误
XXXXX	0xA0188007	ERR_ADEC_NOT_CONFIG	解码通道属性未配置
		in the second se	
	- KB/T	版权所有 © 珠海全志科技股份有限公司。保留一切	权利



Will Little White High State of the State of

		7
错误码	宏定义	描述
0xA0188008	ERR_ADEC_NOT_SUPPORT	操作不被支持
0xA0188009	ERR_ADEC_NOT_PERM	操作不允许
0xA018800C	ERR_ADEC_NOMEM	系统内存不足
0xA018800D	ERR_ADEC_NOBUF	解码通道缓存分配失败
0xA018800E	ERR_ADEC_BUF_EMPTY	解码通道缓存空
0xA018800F	ERR_ADEC_BUF_FULL	解码通道缓存满
0xA0188010	ERR_ADEC_SYS_NOTREADY	系统没有初始化
	ERR_ADEC_DECODER_ERR	音频解码数据错误

REAL REPORT OF THE REAL PROPERTY OF THE PROPER

版权所有 © 珠海全志科技股份有限公司。保留一切权利

430



### 11 Region 模块

### 11.1 概述

用户一般都需要在视频中叠加 OSD 用于显示一些特定的信息(如:通道号、时间戳等),必要时还会填充色块。这些叠加在视频上的 OSD 和遮挡在视频上的色块统称为区域。REGION 模块,用于统一管理这些区域资源。

区域管理可以实现区域的创建,并叠加到视频中或对视频进行遮挡。例如,实际应用中,用户创建一个区域,通过 AW\_MPI\_RGN\_AttachToChn,将该区域叠加到某个通道(如 VENC 通道)中。在通道进行调度时,则会将 OSD 叠加在视频中。一个区域支持通过设置通道显示属性接口指定到多个通道中(如:多个 VENC 通道,多个 VideoScaler 通道,甚至多个 VENC 和 VideoScaler 通道),且支持在每个通道的显示属性(如位置、透明度等)都不同。

### 11.2 功能描述

支持区域叠加 (overlay)、区域遮挡 (cover)、物体矩形框标注 (Object Rectangle Label,简写 ORL) 三种方式。其中叠加支持位图加载、反色等功能,遮挡则支持纯色块的遮挡;矩形框标注仅画线。

区域在不同通道拥有不同的通道显示属性,比如显示位置、层次和区域是否显示等属性。

V459 的 vipp 通道不支持 overlay, cover, 只支持 ORL。

V459 的 venc 通道支持 overlay, cover, 不支持 ORL。

### 11.2.1 状态

本组件没有内部线程,所以没有状态转换。

### 11.3 API 参考

区域管理模块提供以下 MPI:

IAW MPI RGN Create: 创建区域



l AW\_MPI\_RGN\_Destrory: 销毁区域

1 AW MPI RGN GetAttr: 获取区域属性

l AW\_MPI\_RGN\_SetAttr:设置区域属性

l AW MPI RGN SetBitMap: 设置区域位图

l AW MPI RGN AttachToChn: 将区域叠加到通道上

l AW MPI RGN DetachFrmChn:将区域从通道中撤出

l AW MPI RGN SetDisplayAttr: 设置区域的通道显示属性

l AW\_MPI\_RGN\_GetDisplayAttr: 获取区域的通道显示属性

### 11.3.1 AW\_MPI\_RGN\_Create

【描述】

创建区域。

【语法】

P ERRORTYPE AW MPI RGN Create(RGN HANDLE Handle, const RGN ATTR S \*pstRegion);

### 【参数】

参数名称。	描述	allinden	ilinden	i dindent	输入/输出
Handle	区域句柄号。	必须是未使用的	Handle 号。取值范围:[	[0, RGN_HANDLE_MAX)。	 输入
pstRegion	区域属性指针	t. X4	XX TON	A A A A A A A A A A A A A A A A A A A	输入

### 【返回值】

返回值	描述
0	成功
非 0	失败,参见错误码。

### 【需求】

l 头文件: mpi region.h

l 库文件: libmedia mpp.so



### 【注意】

l 创建 Cover 时,只需指定区域类型即可。其它的属性,如区域位置,层次等信息在调用 AW MPI RGN AttachToChn 接口时指定。

1 创建区域时,本接口只进行基本的参数的检查,譬如:最小宽高,最大宽高等;当区域 attach 到通道上时,根据各通道模块支持类型的约束条件进行更加有针对性的参数检查,譬如支持的像 素格式等。

1 对于准备添加到 VENC 通道的区域,其区域的宽高,位置坐标参数必须能是 16 的倍数,否则 无法添加到 VENC 通道。

### 【举例】

无。

11.3.2 AW\_MPI\_RGN\_Destroy

【描述】

销毁区域。

【语法】

ERRORTYPE AW MPI RGN Destroy(RGN HANDLE Handle);

### 【参数】

参数名称	描述	inderi!	inderi	输入/输出
Handle	区域句柄号。	取值范围:	[0, RGN_HANDLE_MAX)。	输入

### 【返回值】

返回值	描述
0	成功
非 0	失败,参见错误码。

### 【需求】

l 头文件: mpi region.h

l 库文件: libmedia mpp.so

【注意】



无必

【举例】

无。

### 11.3.3 AW\_MPI\_RGN\_GetAttr

【描述】

获取区域属性。

【语法】

ERRORTYPE AW\_MPI\_RGN\_GetAttr(RGN\_HANDLE Handle, RGN\_ATTR\_S \*pstRegion);

【参数】

参数名称	描述		*****	15	输入/输出
Handle	区域句柄号。	取值范围:	[0, RGN_HAND	LE_MAX)	 。 输入
pstRegion	区域属性指针	0			输出

【返回值】

 返回值
 描述

 0
 成功

 非 0
 失败,参见错误码。

【需求】

l 头文件: mpi\_region.h

l 库文件: libmedia\_mpp.so

【注意】

无。

【举例】

无。



### 11.3.4 AW\_MPI\_RGN\_SetAttr

### 【描述】

设置区域属性。

### 【语法】

ERRORTYPE AW\_MPI\_RGN\_SetAttr(RGN\_HANDLE Handle, const RGN\_ATTR\_S \*pstRegion);

### 【参数】

参数名称	描述	eri	indeni	输入/输出
Handle	区域句柄号。取值	范围: [0, RGN_HA	NDLE_MAX)。	输入
pstRegion	区域属性指针。	A A	Ja.	输入

### 【返回值】

返回值	描述	
0	成功	
非 0	失败,参见错误码。	) )

### 【需求】

l 头文件: mpi\_region.h

l 库文件: libmedia mpp.so

### 【注意】

l 当区域通过 AW\_MPI\_RGN\_AttachToChn 接口绑定到通道上时,本接口不可用于修改静态属性,但可修改动态属性;当区域没有 attach 到任何通道上时,本接口则既可用于修改静态属性,也可用于修改动态属性。

### 【举例】

无。

### 11.3.5 AW\_MPI\_RGN\_SetBitMap

### 【描述】

设置区域位图,即对区域进行位图填充。



### 【语法】

ERRORTYPE AW\_MPI\_RGN\_SetBitMap(RGN\_HANDLE Handle, const\_BITMAP\_S \*pstBitmap);

### 【参数】

参数名称	描述	输入/输出
Handle	区域句柄号。取值范围: [0, RGN_HANDLE_MAX)。	输入
pstBitmap	位图属性指针。	输入

### 【返回值】

返回值 描述 0 成功 非 0 失败,参见错误码。

### 【需求】

l 头文件: mpi\_region.h

l 库文件: libmedia mpp.so

### 【注意】

1只对 OVERLAY 类型的区域有效。

1位图像素格式必须与区域像素格式一致。

1 当位图大小与区域大小不一致时,区域大小将被修改为与位图大小保持一致。

### 【举例】

无。

### 11.3.6 AW\_MPI\_RGN\_AttachToChn

### 【描述】

将区域叠加到通道上。

### 【语法】

ERRORTYPE AW\_MPI\_RGN\_AttachToChn(RGN\_HANDLE Handle, const MPP\_CHN\_S pstChn, const RGN\_CHN\_ATTR\_S pstChnAttr);



### 【参数】

参数名称	描述		4-11	输入/输出
Handle	区域句柄号。	取值范围:	[0, RGN_HANDLE	_MAX)。输入
pstChn	通道结构体指		XV.	输入
pstChnAtt	r 区域通道显示	属性指针。		输入

### 【返回值】

返回值	描述	
0 非 Qrdsrii	成功 失败,参见错误码。	allinderi Renteri Rent
₹ QC°	失败,参见错误码。	alline alline
STATE OF THE PARTY		
	A TOP TO THE PERSON OF THE PER	
	γı.	
	110	
I 和 VEN	IC 两个通道上。	

### 【需求】

l 头文件: mpi\_region.h

l 库文件: libmedia mpp.so

### 【注意】

1目前为止,只能将区域叠加到 VI 和 VENC 两个通道上。

### 【举例】

无。

### 11.3.7 AW\_MPI\_RGN\_DetachFromChn

### 【描述】

将区域从通道中撤出。

### 【语法】

 $\label{lem:error} ERRORTYPE\ AW\_MPI\_RGN\_DetachFromChn(RGN\_HANDLE\ Handle,\ const\ MPP\_CHN\_S\ *pstChn);$ 

### 【参数】

参数名称	描述	<sub>seri</sub> i	ABC'I	输入/输出
Handle	区域句柄号。取值	范围: [0, RGN	_HANDLE_MAX)。	输入
pstChn	通道结构体指针。		AND THE PERSON NAMED IN COLUMN TO TH	输入



### 【返回值】

X,		7-X 3
	返回值	描述
	0	成功。
	非 0	失败,参见错误码。

### 【需求】

l 头文件: mpi region.h

l 库文件: libmedia mpp.so

11.3.8 AW\_MPI\_RGN\_SetDisplayAttr
【描述】

设置区域的通道显示属性。
【语法】

RROPINGER

ERRORTYPE AW\_MPI\_RGN\_SetDisplayAttr(RGN\_HANDLE Handle, const MPP\_CHN\_S pstChn, const RGN\_CHN\_ATTR\_S pstChnAttr);

### 【参数】

参数名称 💉	描述		类()	Š.	输入/输出
Handle	区域句柄号。	取值范围:	[0, RGN_HA	NDLE_MAX)。	输入
pstChn	通道结构体指	针。			输入
pstChnAttr	区域通道显示	属性指针。			输入

### 【返回值】

返回值	描述	
0 非 0	成功 失败,	参见错误码。



### 【需求】

l 头文件: mpi\_region.h

l 库文件: libmedia mpp.so

【注意】

1 静态属性不能修改,动态属性可以修改。

【举例】

无。

### 11.3.9 AW\_MPI\_RGN\_GetDisplayAttr

### 【描述】

获取区域的通道显示属性。

### 【语法】

ERRORTYPE AW\_MPI\_RGN\_GetDisplayAttr(RGN\_HANDLE Handle, const MPP\_CHN\_S pstChn, RGN\_CHN\_ATTR\_S pstChnAttr);

### 【参数】

参数名称	描述		输入/	输出
Handle	区域句柄号。取值范围:	[0, RGN_HANDLE_MAX)。	输入	
pstChn	通道结构体指针。		输入	
pstChnAttr	区域通道显示属性指针。		输出	NAME OF THE PERSON OF THE PERS

### 【返回值】

返回值	描述	
0	成功	
非 0	失败,参见错误码。	见错误码。

### 【需求】

l 头文件: mpi region.h

l 库文件: libmedia mpp.so 注意】



【举例】

无。

### 11.4 数据类型

### 11.4.1 RGN\_TYPE\_E

OVERLAY\_RGN = 0, /\* video overlay region \*/
COVER\_RGN,
COVEREX\_RGN,
VERLAYEX\_RGN,
RL\_RGN,
RI\_RGN,

RGN\_BUTT

RGN\_TYPE\_E;

【成员】

成员名称	描述
OVERLAY_RGN	通道视频叠加区域。
COVER_RGN	通道视频遮挡区域。
COVEREX_RGN	扩展视频遮挡区域。不支持。
OVERLAYEX_RGN	扩展视频遮挡叠加区域。不支持。
ORL_RGN	物体矩形框标注。

### 【注意事项】

上目前为止,COVEREX\_RGN 和 OVERLAYEX\_RGN 不可用,是保留项。



### 【相关数据类型及接口】

### 11.4.2 RGN\_AREA\_TYPE\_E

【说明】

定义 COVER、COVEREX RGN 类型

【定义】

typedef enum RGN AREA TYPE E

 $AREA_RECT = 0$ ,

AREA\_QUAD\_RANGLE,

AREA BUTT

} RGN\_AREA\_TYPE\_E;

【成员】

成员名称

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\*\*

\* 描述

AREA\_RECT

矩形区域。

AREA QUAD RANGLE

任意四边形区域。不支持。

【注意事项】

无。

【相关数据类型及接口】

无。

### 11.4.3 OVERLAY\_ATTR\_S

【说明】

定义通道叠加区域属性结构体。

【定义】



typedef struct OVERLAY ATTR S

PIXEL\_FORMAT\_E mPixelFmt;

unsigned int mBgColor;

SIZE\_S mSize;

}OVERLAY\_ATTR\_S;

【成员】

成员名称

描述

mPixelFmt 像素格式,只支持 ARGB1555 和 ARGB8888 两种格式。

mBgColor 未使用。

区域宽高大小、宽度: [4,4096],要求以2对齐。高度:宽度:[4,4096],要求以2对齐 mSize 

【注意事项】

无。

【相关数据类型及接口】

无。

### 11.4.4 OVERLAY\_INVERT\_COLOR\_S

【说明】

定义 OSD 反色相关属性。

【定义】

typedef struct OVERLAY INVERT COLOR S

{

SIZE S stInvColArea;

unsigned int mLumThresh;

INVERT COLOR MODE E enChgMod;

BOOL bInvColEn;

FOVERLAY\_INVERT\_COLOR S;

珠海全志科技股份有限公司。保留一切权利



### 【成员】

成员名称	描述。	THE PARTY.	
stInvColArea	单元反色区域,反色处		高度: [16,64], 需要
	16 对齐。宽度: [16,	,64],需要 16 对齐。未修	使用。V316 不支持单元反
	色区域。		
mLumThresh	亮度阀值。暂未使用。		
enChgMod	OSD 反色触发模式。	暂未使用。	
bInvColEn	OSD 反色开关。TRU	JE:开启反色; FALSE:	关闭反色。

### 【注意事项】

无。

【相关数据类型及接口】

无。

## 11.4.5 OVERLAY\_CHN\_ATTR\_S

【说明】

定义通道叠加区域的通道显示属性。

【定义】

typedef struct OVERLAY\_CHN\_ATTR\_S

POINT S stPoint;

unsigned int mFgAlpha;

unsigned int mBgAlpha;

unsigned int mLayer;

OVERLAY\_QP\_INFO\_S stQpInfo;

OVERLAY\_INVERT\_COLOR\_S stInvertColor;

}OVERLAY\_CHN\_ATTR S;



117	A 117	A 1/2	A117
成员名称	描述	A A A A A A A A A A A A A A A A A A A	NA TOP TO THE PERSON NAMED IN COLUMN
stPoint	区域位置:水平位置 X	:[0,4096], 要求以 4 对齐。垂直	直位置 Y:[0,4636],
	要求以 4 对齐。		
u32FgAlpha	※Alpha 位为 1 的像素点	的透明度(前景 Alpha),取	值范围为 [0,128],
	值越小,越透明。只针	对像素格式为 MM_PIXEL_FO	RMAT_RGB_1555
	<u>-</u>	置全局 alpha。对所有 overlay	有效。所以以最后
	一次设置的 overlay 的		
u32BgAlpha	-	的透明度(背景 Alpha ),取	值范围为 [0,128],
	值越小,越透明。未使		
u32Layer		: [0,63],值越大,层次越高。	
stQpInfo	区域编码使用的 QP 值	,未使用。	
stInvertColor	区域反色配置信息。	winder.	Original
ARIV TO STATE OF THE PERSON OF	W/A	W/V	WINT TO STATE OF THE PARTY OF T
【注意事项】	-144 Marv	-XX	
			O KLINETO
	erlay的起始位置和宽高限为	7	A SERVICE OF THE SERV
【相关数据类型及	y接口】		H. The state of th
		ME	
无。		4111	
11 4 6 60	VED CHALATED	S. S	
11.4.6 CO	VER_CHN_ALIR	13	
<b>E</b> WAR <b>Y</b>			
【说明】		À	
定义通道遮挡区域	的通道显示属性。	alinder.	lind
WILE.	ALL TO THE PARTY OF THE PARTY O	SELVY,	WILL STATES
【定义】	A TATALON OF THE PARTY OF THE P	NA TOPINA	A TON
EX.	1/2X	AX.	/4-X

### 【定义】

```
typedef struct COVER CHN_ATTR_S
```

RGN\_AREA\_TYPE\_E enCoverType;

union

{

RECT\_S stRect;

RGN QUADRANGLE S stQuadRangle;

unsigned int mColor;



unsigned int mLayer;

COVER\_CHN\_ATTR\_S

【成员】

成员名称	描述
enCoverType	只支持 AREA_RECT。
stRect	区域矩形位置,宽高。
st Quad Rangle	区域任意四边形位置形状。不支持。
mColor	区域颜色。ARGB 格式。
mLayer	区域层次。范围: [0,7]。

# 

typedef struct ORL CHN ATTR S

RGN\_AREA\_TYPE\_E enAreaType;

union

{

RECT S stRect;

RGN\_QUADRANGLE\_S stQuadRangle;

**}**;

unsigned int mColor;



unsigned int mThick;

unsigned int mLayer;

}ORL\_CHN\_ATTR\_S;

【成员】

	成员名称	描述	
	enAreaType	 只支持 AREA_RECT。	
	stRect	区域矩形位置,宽高。	
	st Quad Rangle	区域任意四边形位置形状。不支持。	
i	mColor	区域颜色。ARGB 格式。	i
allinder	mThick	矩形框线条厚度。(0,7]	dinder
Milderi Milideri	mLayer Market	区域层次。范围: [0,7]。	A Kally La linderi
A A A A A A A A A A A A A A A A A A A	A TAMES OF THE PARTY OF THE PAR	A A A A A A A A A A A A A A A A A A A	A TAMES OF THE PARTY OF THE PAR
【注意事项】			S. H. W. S.
无。		MINE	
【相关数据类型及接		IN	
无。			
		1.14	
11.4.8 RGN	ATTR U		
		. /	
【说明】 <sub>lin</sub> k <sup>gril</sup>	diin	gen! den!	Minden
定义区域属性联合体	· Illand	geri katika kati	A Kille Va linderi
CANAL CONTRACTOR OF THE PARTY O	A A A A A A A A A A A A A A A A A A A	A A A A A A A A A A A A A A A A A A A	
【定义】	The state of the s	La	

typedef union RGN\_ATTR\_U

{

OVERLAY ATTR S stOverlay;

OVERLAYEX\_ATTR\_S stOverlayEx;

} RGN\_ATTR\_U;

【成员】

成员名称	描述	ARIV TOTAL
stOverlay	通道叠加区域	或属性。

版权所有 © 珠海全志科技股份有限公司。保留·



成员名称 描述

扩展叠加区域属性。不支持。 stOverlayEx

【注意事项】

无。

【相关数据类型及接口】

无。

### 11.4,9 RGN\_CHN\_ATTR\_U

【说明】

定义区域通道显示属性联合体。

【定义】

typedef union RGN\_CHN\_ATTR\_U

{

White the state of OVERLAY\_CHN\_ATTR\_S stOverlayChn;

COVER\_CHN\_ATTR\_S stCoverChn;

COVEREX\_CHN\_ATTR\_S stCoverExChn;

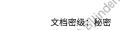
OVERLAYEX\_CHN\_ATTR\_S stOverlayExChn;

ORL CHN ATTR S stOrlChn

} RGN\_CHN\_ATTR\_U;

【成员】

成员名称	描述
stOverlayChn	叠加区域通道显示属性。
stCoverChn	遮挡区域通道显示属性。
stCoverExChn	扩展遮挡区域通道显示属性。不支持。
stOverlayExChn	扩展叠加区域通道显示属性。不支持。





【相关数据类型及接口】

无。

11.4.10 RGN\_ATTR\_S

【说明】

定义区域属性结构体。

【定义】

typedef struct RGN\_ATTR\_S

RGN\_TYPE\_E enType;

RGN\_ATTR\_U unAttr;

} RGN\_ATTR\_S;

【成员】

成员名称 描述

enType 区域类型。

unAttr

区域属性。

【注意事项】

无。

【相关数据类型及接口】

无。

11.4.11 RGN\_CHN\_ATTR\_S

【说明】

定义区域属性结构体。 (定义)



typedef struct RGN\_CHN\_ATTR\_S

**\***`{

BOOL bShow;

RGN\_TYPE\_E enType;

RGN\_CHN\_ATTR\_U unChnAttr;

} RGN CHN ATTR S;

【成员】

成员名称 描述
bShow 区域是否显示。enType 区域类型。unChnAttr 区域通道显示属性。

【注意事项】

无。

【相关数据类型及接口】

无。

### 11.5 错误码

K/X	No.	X/A*
错误代码	宏定义	描述
0xA0038001	ERR_RGN_INVALID_DEVID	设备 ID 超出合法范围
0xA0038002	ERR_RGN_INVALID_CHNID	通道组号错误或无效区域句柄
0xA0038003	ERR_RGN_ILLEGAL_PARAM	参数超出合法范围
0xA0038004	ERR_RGN_EXIST	重复创建已存在的设备、通道或资源
0xA0038005	ERR_RGN_UNEXIST	试图使用或者销毁不存在的设备、通道或资源
0xA0038006	ERR_RGN_NULL_PTR	函数参数中有空指针
0xA0038007	ERR_RGN_NOT_CONFIG	模块没有配置。
0xA0038008	ERR_RGN_NOT_SUPPORT	不支持的参数或者功能
0xA0038009	ERR_RGN_NOT_PERM	该操作不允许,如试图修改静态配置参数
0xA003800C	ERR_RGN_NOMEM	分配内存失败,如系统内存不足。
0xA003800D	ERR_RGN_NOBUF	分配缓存失败,如申请的数据缓冲区太大
0xA003800E	ERR_RGN_BUF_EMPTY	缓冲区中无数据
0xA003800F	ERR_RGN_BUF_FULL	缓冲区中数据满
E. C.	A STATE OF THE STA	A A A A A A A A A A A A A A A A A A A



文档密级、秘密

	A117	A 1\7	A117	A1\7
	错误代码	宏定义	描述	A PART OF THE PROPERTY OF THE PART OF THE
, Hill	0xA0038011	ERR_RGN_BADADDR	地址非法。	
12/2	0xA0038012	ERR_RGN_BUSY	系统忙	1/2/2
K. K.	0xA0038010	ERR_RGN_NOTREADY	系统没有初始化或没有	加载相应模块

Report of the state of the stat



### 著作权声明

版权所有 © 2020 珠海全志科技股份有限公司。保留一切权利。

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

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

### 商标声明



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

### 免责声明

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

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

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