

MI VPE API

Version 2.06

© 2019 SigmaStar Technology Corp. All rights reserved.

SigmaStar Technology makes no representations or warranties including, for example but not limited to, warranties of merchantability, fitness for a particular purpose, non-infringement of any intellectual property right or the accuracy or completeness of this document, and reserves the right to make changes without further notice to any products herein to improve reliability, function or design. No responsibility is assumed by SigmaStar Technology arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

SigmaStar is a trademark of SigmaStar Technology Corp. Other trademarks or names herein are only for identification purposes only and owned by their respective owners.

REVISION HISTORY

Revision No.	Description	Date
2.03	<ul style="list-style-type: none">Initial release	04/12/2018
2.04	<ul style="list-style-type: none">Updated MI_VPE_SetPortMode	01/02/2019
2.05	<ul style="list-style-type: none">Add MI_VPE_SetPortShowPosition	20/03/2019
2.06	<ul style="list-style-type: none">Add MI_VPE_IspInitPara_t	17/05/2019

TABLE OF CONTENTS

REVISION HISTORY	i
TABLE OF CONTENTS.....	ii
1. API 参考	1
1.1. MI_VPE_CreateChannel.....	3
1.2. MI_VPE_DestroyChannel	5
1.3. MI_VPE_GetChannelAttr	6
1.4. MI_VPE_SetChannelAttr	7
1.5. MI_VPE_StartChannel	8
1.6. MI_VPE_StopChannel.....	9
1.7. MI_VPE_EnablePort	9
1.8. MI_VPE_DisablePort.....	10
1.9. MI_VPE_SetChannelParam	11
1.10. MI_VPE_GetChannelParam	12
1.11. MI_VPE_SetChannelCrop.....	13
1.12. MI_VPE_GetChannelCrop	14
1.13. MI_VPE_GetChannelRegionLuma	14
1.14. MI_VPE_SetChannelRotation	15
1.15. MI_VPE_GetChannelRotation	16
1.16. MI_VPE_SetPortMode	17
1.17. MI_VPE_GetPortMode	18
1.18. MI_VPE_SetPortCrop.....	20
1.19. MI_VPE_GetPortCrop	21
1.20. MI_VPE_SetChannelIspParam.....	22
1.21. MI_VPE_GetChannelIspParam	23
1.22. MI_VPE_SetPortShowPosition	24
1.23. MI_VPE_GetPortShowPosition.....	25
2. VPE 数据类型	26
2.1. MI_VPE_CHANNEL.....	27
2.2. MI_VPE_PORT	27
2.3. MI_VPE_MAX_CHANNEL_NUM.....	27
2.4. MI_VPE_MAX_PORT_NUM.....	29
2.5. MI_VPE_RunningMode_e	29
2.6. MI_VPE_SensorChannel_e.....	29
2.7. MI_VPE_IspApiHeader_t	30
2.8. MI_VPE_IspApiData_t	30
2.9. MI_VPE_ChannelAttr_t.....	32
2.10. MI_VPE_PqParam_t.....	33
2.11. MI_VPE_HDRTYPE_e.....	34
2.12. MI_VPE_3DNR_Level_e.....	35
2.13. MI_VPE_ChannelPara_t.....	37
2.14. MI_VPE_RegionInfo_t	37
2.15. MI_VPE_PortMode_t	38
2.16. MI_VPE_ISPBScaler_t	38

2.17. MI_VPE_IspParamType_e40

2.18. MI_VPE_ChannelIspParam_t41

2.19. MI_VPE_IspInitPara_t41

3. 错误码43

1. API 参考

该功能模块提供以下 API:

API 名	功能
MI_VPE_CreateChannel	创建一个 VPE channel
MI_VPE_DestroyChannel	销毁一个 VPE channel
MI_VPE_GetChannelAttr	获取一个 VPE channel 属性
MI_VPE_SetChannelAttr	设定一个 VPE channel 属性
MI_VPE_StartChannel	启用 VPE channel
MI_VPE_StopChannel	禁用 VPE channel
MI_VPE_EnablePort	启用 VPE 端口
MI_VPE_DisablePort	禁用 VPE 端口
MI_VPE_SetChannelParam	设定 VPE channel 参数
MI_VPE_GetChannelParam	获取 VPE channel 参数
MI_VPE_SetChannelCrop	设定 VPE channel crop window
MI_VPE_GetChannelCrop	获取 VPE channel crop window
MI_VPE_GetChannelRegionLuma	获取 VPE 通道 Luma 直方图统计
MI_VPE_SetChannelRotation	设定 VPE 通道视频旋转类型
MI_VPE_GetChannelRotation	获取 VPE 通道视频旋转类型
MI_VPE_SetPortMode	设定 VPE 端口模式
MI_VPE_GetPortMode	获取 VPE 端口模式
MI_VPE_SetPortCrop	设定 VPE out port crop window
MI_VPE_GetPortCrop	获取 VPE out port crop window 设置参数
MI_VPE_SetChannelIspParam	设定 VPE isp 参数
MI_VPE_GetChannelIspParam	获取 VPE isp 设置的参数
MI_VPE_SetPortShowPosition	设置 vpe output port 显示位置

API 名	功能
MI_VPE_GetPortShowPosition	获取 vpe output port 显示位置

1.1. MI_VPE_CreateChannel

➤ 描述

创建一个 VPE channel。

➤ 语法

```
MI_S32 MI_VPE_CreateChannel(MI_VPE_CHANNEL VpeCh,  
MI_VPE_ChannelAttr_t*pstVpeChAttr);
```

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
pstVpeChAttr	VPE channel 属性指针。	输入

➤ 返回值

返回值 { 0 成功。
非 0 失败，参照[错误码](#)。

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

- 可创建多个 channel，最大 channel 数为 MI_VPE_MAX_CHANNEL_NUM.
- 当 eRunningMode 为 E_MI_VPE_RUNNING_MODE_REALTIME_MODE 和 E_MI_VPE_RUNNING_MODE_FRAMEBUF_CAM_MODE 时 u32MaxW/ u32MaxH 需要配置为 vpe input size 的 width 和 height。
- 不支持重复创建

➤ 举例

```
MI_VPE_ChannelAttr_t stChannelVpeAttr;  
MI_SYS_WindowRect_t stCropWin;  
MI_VPE_CHANNEL VpeChannel;  
MI_VPE_PORT VpePort;  
MI_S32 s32Ret = MI_OK;  
  
VpeChannel = 0;  
VpePort = 0;  
  
stChannelVpeAttr.u32MaxW = 1920;  
stChannelVpeAttr.u32MaxH = 1080;
```



```

stChannelVpeAttr.bNREn= FALSE;
stChannelVpeAttr.bEdgeEn= FALSE;
stChannelVpeAttr.bESEN= FALSE;
stChannelVpeAttr.bContrastEn= FALSE;
stChannelVpeAttr.bUVInvert= FALSE;
stChannelVpeAttr.ePixFmt = MI_VPE_PIXEL_FORMAT_YUV_SEMIPLANAR_420;
stChannelVpeAttr.eRunningMode = E_MI_VPE_RUNNING_MODE_FRAMEBUF_DVR_MODE;
stChannelVpeAttr.eSensorBindId= E_MI_VPE_SENSOR_INVALID;

s32Ret = MI\_VPE\_CreateChannel(VpeChannel, &stChannelVpeAttr);
if(s32Ret != MI_OK)
{
    return s32Ret;
}

s32Ret = MI\_VPE\_GetChannelAttr(VpeChannel, & stChannelVpeAttr);
if(s32Ret != MI_OK)
{
    return s32Ret;
}

stChannelVpeAttr.bContrastEn = TRUE;
stChannelVpeAttr.bNREn = TRUE;
s32Ret = MI\_VPE\_SetChannelAttr(VpeChannel, & stChannelVpeAttr);
if(s32Ret != MI_OK)
{
    return s32Ret;
}

s32Ret = MI\_VPE\_GetChannelCrop(VpeChannel, &stCropWin) ;
if(s32Ret != MI_OK)
{
    return s32Ret;
}
stCropWin.u16X = 20;
stCropWin.u16Y = 40;
stCropWin.u16Width = 1920;
stCropWin.u16Height = 1080;
s32Ret = MI\_VPE\_SetChannelCrop(VpeChannel, &stCropWin);
if(s32Ret != MI_OK)
{
    return s32Ret;
}

s32Ret = MI\_VPE\_EnablePort(VpeChannel, VpePort);
if(s32Ret != MI_OK)
{
    return s32Ret;
}

```

```
}

s32Ret = MI\_VPE\_StartChannel (VpeChannel);
if(s32Ret != MI_OK)
{
    return s32Ret;
}

/*****/
/* call sys bind interface */
/*****/
s32Ret = MI\_VPE\_StopChannel (VpeChannel);
if(s32Ret != MI_OK)
{
    return s32Ret;
}

s32Ret = MI\_VPE\_DisablePort(VpeChannel, VpePort);
{
    return s32Ret;
}

/*****/
/* call sys unbind interface */
/*****/
s32Ret = MI\_VPE\_DestroyChannel(VpeChannel);
if(s32Ret != MI_OK)
{
    return s32Ret;
}
```

➤ 相关主题

[MI_VPE_DestroyChannel](#)

1.2. [MI_VPE_DestroyChannel](#)

➤ 描述

销毁一个 VPE channel.

➤ 语法

MI_S32 MI_VPE_DestroyChannel ([MI_VPE_CHANNEL](#) VpeCh);

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。	输入

	取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	
--	--	--

➤ 返回值

返回值 { 0 成功。
非 0 失败，参照[错误码](#)。

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

无

➤ 举例

请参见 [MI_VPE_CreateChannel](#)

➤ 相关主题

[MI_VPE_CreateChannel](#)

1.3. MI_VPE_GetChannelAttr

➤ 描述

获取一个 VPE channel 属性。

➤ 语法

MI_S32 MI_VPE_GetChannelAttr([MI_VPE_CHANNEL](#) VpeCh, [MI_VPE_ChannelAttr_t](#) *pstGrpAttr);

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
pstVpeChAttr	VPE channel 属性指针。	输入

➤ 返回值

返回值 { 0 成功。
非 0 失败，参照[错误码](#)。

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

无

➤ 举例

请参见 [MI_VPE_CreateChannel](#)

➤ 相关主题

[MI_VPE_SetChannelAttr](#)

1.4. MI_VPE_SetChannelAttr

➤ 描述

设定一个 VPE channel 属性。

➤ 语法

MI_S32 MI_VPE_SetChannelAttr ([MI_VPE_CHANNEL](#) VpeCh, [MI_VPE_ChannelAttr_t](#) *pstVpeChAttr);

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
pstVpeChAttr	VPE channel 属性指针。	输入

➤ 返回值

返回值 $\left\{ \begin{array}{ll} 0 & \text{成功。} \\ \text{非 } 0 & \text{失败，参照[错误码](#)。} \end{array} \right.$

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

无

➤ 举例

请参见 [MI VPE CreateChannel](#)

➤ 相关主题

[MI VPE GetChannelAttr](#)

1.5. MI_VPE_StartChannel

➤ 描述

启用 VPE channel.

➤ 语法

MI_S32 MI_VPE_StartChannel([MI_VPE_CHANNEL](#) VpeCh);

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入

➤ 返回值

返回值 $\left\{ \begin{array}{ll} 0 & \text{成功。} \\ \text{非 } 0 & \text{失败，参照[错误码](#)。} \end{array} \right.$

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

无

➤ 举例

请参见 [MI_VPE_CreateChannel](#)

➤ 相关主题

[MI_VPE_StopChannel](#)

1.6. MI_VPE_StopChannel

➤ 描述

禁用 VPE channel.

➤ 语法

```
MI_S32 MI_VPE_StopChannel(MI\_VPE\_CHANNEL VpeCh);
```

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入

➤ 返回值

返回值 $\left\{ \begin{array}{ll} 0 & \text{成功。} \\ \text{非 0} & \text{失败，参照[错误码](#)。} \end{array} \right.$

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

无

➤ 举例

请参见 [MI_VPE_CreateChannel](#)

➤ 相关主题

[MI_VPE_StartChannel](#)

1.7. MI_VPE_EnablePort

➤ 描述

启用 VPE 端口.

➤ 语法

```
MI_S32 MI_VPE_EnablePort(MI\_VPE\_CHANNEL VpeCh, MI\_VPE\_PORT s32VpePort);
```

➤ 参数

参数名称	描述	输入/输出
------	----	-------

VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
s32VpePort	s32VpePort 号。 取值范围：[0, VPE_MAX_PORT_NUM)。	输入

➤ 返回值

返回值 $\begin{cases} 0 & \text{成功。} \\ \text{非 } 0 & \text{失败，参照[错误码](#)。} \end{cases}$

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

无

➤ 举例

请参见 [MI_VPE_CreateChannel](#)

➤ 相关主题

[MI_VPE_DisablePort](#)

1.8. MI_VPE_DisablePort

➤ 描述

禁用 VPE 端口。

➤ 语法

MI_S32 MI_VPE_DisablePort([MI_VPE_CHANNEL](#) VpeCh, [MI_VPE_PORT](#) VpePort);

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
VpePort	s32VpePort 号。 取值范围：[0, MI_VPE_MAX_PORT_NUM)。	输入

➤ 返回值

返回值 { 0 成功。
非 0 失败，参照[错误码](#)。

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

无

➤ 举例

请参见 [MI_VPE_CreateChannel](#)

➤ 相关主题

[MI_VPE_EnablePort](#)

1.9. MI_VPE_SetChannelParam

➤ 描述

设定 VPE channel 参数。

➤ 语法

MI_S32 MI_VPE_SetChannelParam ([MI_VPE_CHANNEL](#) VpeCh,[MI_VPE_ChannelPara_t](#) *pstVpeParam);

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
pstVpeParam	通道参数设置	输入

➤ 返回值

返回值 { 0 成功。
非 0 失败，参照[错误码](#)。

- 需求
 - 头文件：mi_vpe.h
 - 库文件：
- ※ 注意
 - Channel 已经创建成功。pstVpeParam 详见 [MI VPE ChannelPara_t](#) 说明。
 - 先通过 [MI VPE GetChannelParam](#) 获取当前参数，再进行设定。
- 举例

无
- 相关主题

[MI VPE GetChannelParam](#)

1.10. MI_VPE_GetChannelParam

- 描述

获取 VPE channel 参数。
- 语法

```
MI_S32 MI_VPE_GetChannelParam (MI VPE\_CHANNEL VpeCh, MI\_VPE\_ChannelPara\_t *pstVpeParam);
```

- 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
pstVpeParam	通道参数设置	输入

- 返回值

返回值 $\left\{ \begin{array}{ll} 0 & \text{成功。} \\ \text{非 } 0 & \text{失败，参照[错误码](#)。} \end{array} \right.$

- 需求
 - 头文件：mi_vpe.h
 - 库文件：
- ※ 注意
 - Channel 已经创建成功。pstVpeParam 详见 [MI VPE ChannelPara_t](#) 说明。

➤ 举例
无

➤ 相关主题
[MI_VPE_SetChannelParam](#)

1.11. MI_VPE_SetChannelCrop

➤ 描述
设定 VPE channel crop window。

➤ 语法
MI_S32 MI_VPE_SetChannelCrop ([MI_VPE_CHANNEL](#) VpeCh, MI_SYS_WindowRect_t *pstCropInfo);

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
pstCropInfo	通道 Crop Window 参数设置	输入

➤ 返回值
返回值 { 0 成功。
非 0 失败，参照[错误码](#)。

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

- Channel 已经创建成功。Crop window 的设定都是基于原始画面大小。

➤ 举例
无

➤ 相关主题
[MI_VPE_GetChannelCrop](#)

1.12. MI_VPE_GetChannelCrop

➤ 描述
获取 VPE channel crop window。

➤ 语法
MI_S32 MI_VPE_GetChannelCrop([MI_VPE_CHANNEL](#) VpeCh, MI_SYS_WindowRect_t *pstCropInfo);

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
pstCropInfo	通道 Crop Window 参数设置	输出

➤ 返回值

返回值 { 0 成功。
非 0 失败，参照[错误码](#)。

- 需求
- 头文件：mi_vpe.h
 - 库文件：

※ 注意

- Channel 已经创建成功。

➤ 举例
无

➤ 相关主题
[MI_VPE_SetChannelCrop](#)

1.13. MI_VPE_GetChannelRegionLuma

➤ 描述
获取 VPE 通道 Luma 直方图统计。

➤ 语法
MI_S32 MI_VPE_GetChannelRegionLuma ([MI_VPE_CHANNEL](#) VpeCh, [MI_VPE_RegionInfo_t](#) *pstRegionInfo, MI_U32 *pu32LumaData,MI_S32 s32MilliSec);

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
pstRegionInfo	指定统计区域，及数量。	输入
pu32LumaData	输出统计数据	输出
s32MilliSec	API 等待的 timeout,单位 ms	输入

➤ 返回值

返回值 { 0 成功。
 非 0 失败，参照[错误码](#)。

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

- Channel 已经创建成功。

➤ 举例

无

➤ 相关主题

无

1.14. MI_VPE_SetChannelRotation

➤ 描述

设定 VPE 通道视频旋转类型。

➤ 语法

MI_S32 MI_VPE_SetChannelRotation ([MI_VPE_CHANNEL](#) VpeCh, MI_SYS_Rotate_e eType);

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
eType	旋转角度设定	输入

➤ 返回值

返回值 { 0 成功。
非 0 失败，参照[错误码](#)。

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

- Channel 已经创建成功。
- 目前 I2 仅支援 270 度旋转。

➤ 举例

无

➤ 相关主题

[MI_VPE_GetChannelRotation](#)

1.15. MI_VPE_GetChannelRotation

➤ 描述

获取 VPE 通道视频旋转类型。

➤ 语法

MI_S32 MI_VPE_GetChannelRotation ([MI_VPE_CHANNEL](#) VpeCh, MI_SYS_Rotate_e *pType);

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
peType	旋转角度设定	输出

➤ 返回值

返回值 { 0 成功。
非 0 失败，参照[错误码](#)。

- 需求
 - 头文件：mi_vpe.h
 - 库文件：
- ※ 注意
 - Channel 已经创建成功。
- 举例

无
- 相关主题

[MI_VPE_SetChannelRotation](#)

1.16. MI_VPE_SetPortMode

- 描述

设定 VPE 端口模式.
- 语法

MI_S32 MI_VPE_SetPortMode([MI_VPE_CHANNEL](#) VpeCh, [MI_VPE_PORT](#) VpePort, [MI_VPE_PortMode_t](#) *pstVpeMode);
- 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
VpePort	VPE port 号。 取值范围：[0, MI_VPE_MAX_PORT_NUM)。	输入
pstVpeMode	VPE 端口模式	输出

➤ 返回值

返回值 { 0 成功。
非 0 失败，参照[错误码](#)。

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

PortId	pixel	MaxSize
Port0	YUV420_NV12, YUV422_YUYV	width max 3840
Port1	YUV420_NV12, YUV422_YUYV	width max 2688
Port2	YUV420_NV12, YUV422_YUYV	width max 2688
Port3	YUV422_YUYV, MST420/YUV420	width max 3840

1. 只有 MSR930、SAV610E、SAV538E/S、SAV638E/S、SAV838E/S 支持 port3 输出 MST420.
2. 只有 SSC329Q, SSC326D, SSA520, SSC328Q 这几款芯片 port3 可以输出 YUV420, 并且 port0, 1, 2, 3 支持 ARGB8888 和 BGRA8888 的输出。

➤ 举例

无

➤ 相关主题

[MI_VPE_GetPortMode](#)

1.17. MI_VPE_GetPortMode

➤ 描述

获取 VPE 端口模式.

➤ 语法

MI_S32 MI_VPE_GetPortMode ([MI_VPE_CHANNEL](#) VpeCh, [MI_VPE_PORT](#) VpePort, [MI_VPE_PortMode_t](#) *pstVpeMode);

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
VpePort	VPE port 号。 取值范围：[0, MI_VPE_MAX_PORT_NUM)。	输入

pstVpeMode	VPE 端口模式	输出
------------	----------	----

➤ 返回值

返回值 { 0 成功。
非 0 失败，参照[错误码](#)。

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

无

➤ 举例

无

➤ 相关主题

[MI_VPE_SetPortMode](#)

1.18. MI_VPE_SetPortCrop

➤ 描述

设定 VPE output port crop window.

➤ 语法

MI_S32 MI_VPE_SetPortCrop ([MI_VPE_CHANNEL](#) VpeCh, [MI_VPE_PORT](#) VpePort,
MI_SYS_WindowRect_t *pstOutCropInfo);

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
VpePort	VPE port 号。 取值范围：[0, MI_VPE_MAX_PORT_NUM)。	输入
*pstOutCropInfo	Output port crop window 设置	输入

➤ 返回值

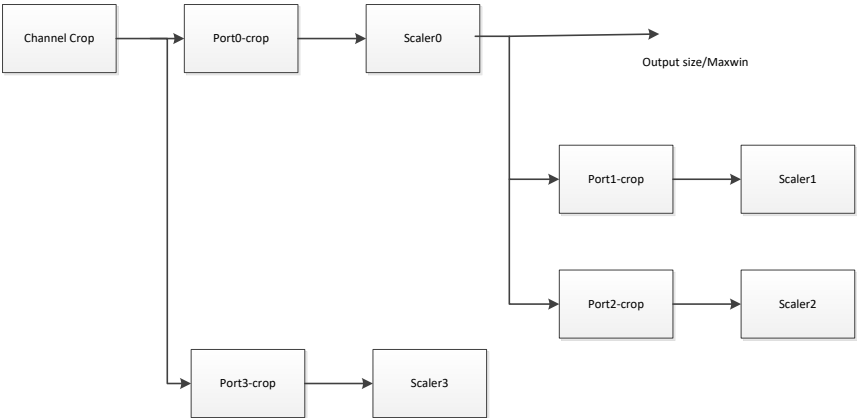
返回值 { 0 成功。
非 0 失败，参照[错误码](#)。

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

此 API 只有 SSC329Q, SSC326D, SSA520, SSC328Q 芯片有该功能。



如上图，由于 port1,2 的 input 是是 port0 的 output。

1. port0 enable, port1,2 的 crop win < port0 size,
2. port0 disable, port1,2 的 crop win < channelAttr.maxwin

➤ 举例

无

➤ 相关主题

[MI_VPE_GetPortCrop](#)

1.19. MI_VPE_GetPortCrop

➤ 描述

获取 VPE out port crop window 设置参数。

➤ 语法

MI_S32 MI_VPE_GetPortCrop ([MI_VPE_CHANNEL](#) VpeCh, [MI_VPE_PORT](#) VpePort, MI_SYS_WindowRect_t *pstOutCropInfo);

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
VpePort	VPE port 号。 取值范围：[0, MI_VPE_MAX_PORT_NUM)。	输入

*pstOutCropInfo	Output port crop window 参数	输出
-----------------	----------------------------	----

➤ 返回值

返回值 { 0 成功。
非 0 失败，参照[错误码](#)。

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

此 API 只有 SSC329Q，SSC326D，SSA520，SSC328Q 芯片有该功能。

➤ 举例

无

➤ 相关主题

[MI_VPE_SetPortCrop](#)

1.20. MI_VPE_SetChannelIspParam

➤ 描述

设定 VPE Isp 相关参数。

➤ 语法

MI_S32 MI_VPE_SetChannelIspParam([MI_VPE_CHANNEL](#) VpeCh, [MI_VPE_ChannelIspParam_t](#) *pstIspParam);

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
*pstIspParam	Isp 相关参数设置	输入

➤ 返回值

返回值 { 0 成功。
非 0 失败，参照[错误码](#)。

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

此 API 只有 MSR930、SAV610E、SAV538E/S、SAV638E/S、SAV838E/S 这几款芯片可以使用，其它芯片没有此功能。

➤ 举例

无

➤ 相关主题

[MI_VPE_GetChannelIspParam](#)

1.21. MI_VPE_GetChannelIspParam

➤ 描述

获取 VPE isp 设置的参数。

➤ 语法

```
MI_S32 MI_VPE_GetChannelIspParam(MI\_VPE\_CHANNEL VpeCh, MI\_VPE\_ChannelIspParam\_t *pstGetIspParam);
```

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
* pstGetIspParam	Isp 相关参数设置	输出

➤ 返回值

返回值 $\left\{ \begin{array}{ll} 0 & \text{成功。} \\ \text{非 } 0 & \text{失败，参照[错误码](#)。} \end{array} \right.$

➤ 需求

- 头文件：mi_vpe.h
- 库文件：

※ 注意

此 API 只有 MSR930、SAV610E、SAV538E/S、SAV638E/S、SAV838E/S 这几款芯片有该功能。

➤ 举例

无

➤ 相关主题

[MI_VPE_SetChannelIspParam](#)

1.22. MI_VPE_SetPortShowPosition

➤ 描述

设置 vpe output port 显示位置

➤ 语法

MI_S32 MI_VPE_SetPortShowPosition([MI_VPE_CHANNEL](#) VpeCh, [MI_VPE_PORT](#) VpePort, MI_SYS_WindowRect_t *pstPortPositionInfo);

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
VpePort	VPE port 号。 取值范围：[0, MI_VPE_MAX_PORT_NUM)。	输入
* pstPortPositionInfo	显示位置参数	输入

➤ 返回值

返回值 { 0 成功。
 非 0 失败，参照[错误码](#)。

➤ 需求

- 头文件：mi_vpe.h
- 库文件：libmi_vpe.so

※ 注意

1. 在没有调用此 API 的情况下，默认是全屏显示。
2. 需要在 [MI_VPE_SetPortMode](#) 后设置。

➤ 举例

无

➤ 相关主题

[MI_VPE_GetPortShowPosition](#)

1.23. MI_VPE_GetPortShowPosition

➤ 描述

获取 vpe output port 显示位置

➤ 语法

```
MI_S32 MI_VPE_GetPortShowPosition(MI\_VPE\_CHANNEL VpeCh, MI\_VPE\_PORT VpePort,
MI_SYS_WindowRect_t *pstPortPositionInfo);
```

➤ 参数

参数名称	描述	输入/输出
VpeCh	VPE channel 号。 取值范围：[0, MI_VPE_MAX_CHANNEL_NUM)。	输入
VpePort	VPE port 号。 取值范围：[0, MI_VPE_MAX_PORT_NUM)。	输入
* pstPortPositionInfo	显示位置参数	输出

➤ 返回值

返回值 { 1 成功。
 非 0 失败，参照[错误码](#)。

➤ 需求

- 头文件：mi_vpe.h
- 库文件：libmi_vpe.so

※ 注意

无

➤ 举例

无

➤ 相关主题

[MI_VPE_SetPortShowPosition](#)

2. VPE 数据类型

VPE 模块相关数据类型定义如下：

MI_VPE_CHANNEL	定义 VPE channel 的类型
MI_VPE_PORT	定义 VPE port 的类型
MI_VPE_MAX_CHANNEL_NUM	定义 VPE channel 的最大个数
MI_VPE_MAX_PORT_NUM	定义 VPE 每个 channel 的 output port 最大个数
MI_VPE_RunningMode_e	定义 VPE 运行模式
MI_VPE_SensorChannel_e	定义 VPE 绑定 sensor 的 ID
MI_VPE_IspApiHeader_t	定义 VPE 传给 ISP 数据头的信息
MI_VPE_IspApiData_t	定义 VPE 传给 ISP 数据头信息和数据内容
MI_VPE_ChannelAttr_t	定义 VPE Buffer 压缩类型
MI_VPE_PqParam_t	定义 VPE PQ 种 NR, EdgeGain, Contrast 的设置
MI_VPE_HDRType_e	定义 VPE HDR 开关模式
MI_VPE_3DNR_Level_e	定义 3DNR 设置等级
MI_VPE_ChannelPara_t	定义 VPE channel 属性参数
MI_VPE_RegionInfo_t	定义 VPE channel 亮度统计区域类型
MI_VPE_PortMode_t	定义 VPE 端口模式
MI_VPE_ISPScaler_t	定义 VPE ISP Bayer Scaler 参数
MI_VPE_IspParamType_e	定义 VPE ISP 参数设置类型
MI_VPE_ChannelIspParam_t	定义 VPE ISP 相关参数

2.1. MI_VPE_CHANNEL

- 说明
定义 MI_VPE_CHANNEL 类型。
- 定义
`typedef MI_S32 MI_VPE_CHANNEL`
- ※ 注意事项
无。
- 相关数据类型及接口
无。

2.2. MI_VPE_PORT

- 说明
定义 MI_VPE_PORT 类型。
- 定义
`typedef MI_S32 MI_VPE_PORT`
- ※ 注意事项
无。
- 相关数据类型及接口
无。

2.3. MI_VPE_MAX_CHANNEL_NUM

- 说明
定义 VPE channel 的最大个数。
- 定义
`#define MI_VPE_MAX_CHANNEL_NUM (256)`
- ※ 注意事项
无。
- 相关数据类型及接口

无。

2.4. MI_VPE_MAX_PORT_NUM

➤ 说明

定义 VPE port 的最大个数。

➤ 定义

```
#define MI_VPE_MAX_PORT_NUM (4)
```

※ 注意事项

无。

➤ 相关数据类型及接口

无。

2.5. MI_VPE_RunningMode_e

➤ 说明

定义 VPE 工作在某种模式下。

➤ 定义

```
typedef enum  
{  
    E_MI_VPE_RUNNING_MODE_INVALID = 0,  
    E_MI_VPE_RUNNING_MODE_FRAMEBUF_DVR_MODE,  
    E_MI_VPE_RUNNING_MODE_FRAMEBUF_CAM_MODE,  
    E_MI_VPE_RUNNING_MODE_REALTIME_MODE,  
    E_MI_VPE_RUNNING_MODE_MAX,  
} MI_VPE_RunningMode_e;
```

※ 注意事项

无。

➤ 相关数据类型及接口

无。

2.6. MI_VPE_SensorChannel_e

➤ 说明

定义 VPE 与哪个 sensor 有绑定关系。

➤ 定义

```
typedef enum
{
    E_MI_VPE_SENSOR_INVALID = 0,
    E_MI_VPE_SENSOR0,
    E_MI_VPE_SENSOR1,
    E_MI_VPE_SENSOR2,
    E_MI_VPE_SENSOR3,
    E_MI_VPE_SENSOR_MAX
}MI_VPE_SensorChannel_e;
与硬件上 sensor device0/1/2/3 对应。
```

※ 注意事项

当 vpe 前端不接 sensor 的时候使用 E_MI_VPE_SENSOR_INVALID，在 realtime mode 设定无效，默认使用 sensor0 连接。

➤ 相关数据类型及接口

[MI_VPE_ChannelAttr_t](#)。

2.7. MI_VPE_IspApiHeader_t

➤ 说明

定义 VPE 传给 ISP 的数据信息。

➤ 定义

```
typedef struct MI_VPE_IspApiHeader_s
{
    MI_U32 u32HeadSize;    //Size of MI_IspApiHeader_t
    MI_U32 u32DataLen;     //Data length;
    MI_U32 u32CtrlID;      //Function ID
    MI_U32 u32Channel;     //Isp channel number
    MI_S32 s32Ret;        //Isp api return value
} MI_VPE_IspApiHeader_t;
```

※ 注意事项

无。

➤ 相关数据类型及接口

[MI_VPE_IspApiData_t](#)。

2.8. MI_VPE_IspApiData_t

➤ 说明

定义 VPE 传给 ISP 的数据信息和数据内容。

➤ 定义

```
typedef struct MI_VPE_IspApiData_s
{
    MI_VPE_IspApiHeader_t stHeader; //数据信息
    MI_U8 u8Data[0];                //数据内容指针
}MI_VPE_IspApiData_t;
```

➤ 注意事项

无。

➤ 相关数据类型及接口

[MI_VPE_IspApiHeader_t](#)。

2.9. MI_VPE_ChannelAttr_t

➤ 说明

定义 VPE channel 静态属性。

➤ 定义

```
typedef struct MI_VPE_ChannelAttr_s
{
    MI_U16 u16MaxW;
    MI_U16 u16MaxH;
    MI_SYS_PixelFormat_e ePixFmt;
    MI_VPE_SensorChannel_e eSensorBindId;

    MI_BOOL bNrEn;
    MI_BOOL bEdgeEn;
    MI_BOOL bEsEn;
    MI_BOOL bContrastEn;
    MI_BOOL bUvInvert;
    MI_BOOL bRotation;
    MI_VPE_RunningMode_e eRunningMode;
    MI_VPE_IspInitPara_t tIspInitPara;
}MI_VPE_ChannelAttr_t;
```

➤ 成员

成员名称	描述
u32MaxW	最大图像宽度，静态属性不可更改。
u32MaxH	最大图像高度，静态属性不可更改。
ePixFmt	像素格式。DVR 模式只支持 SemiPlannar422 和 SemiPlannar420。前端为 sensor 时设置 sensor 的 pixel。设置为静态属性，创建 Chanel 时设定，不可更改。

成员名称	描述
eSensorBindId	前端接 sensor 时设置绑定 sensorid。
bNrEn	去噪使能
bEdgeEn	锐化使能
bEsEn	边缘平滑使能
bContrastEn	对比度使能
bUvInvert	UV 反转使能
eRunningMode	VPE 运行模式，静态属性不可更改。
bRotation	旋转功能使能
tIspInitPara	初始 isp 参数

※ 注意事项

静态属性创建 channel 时设定，不可更改。

➤ 相关数据类型及接口

[MI_VPE_RunningMode_e](#)

[MI_VPE_SensorChannel_e](#)

[MI_VPE_IspInitPara_t](#)

2.10. MI_VPE_PqParam_t

➤ 说明

定义 VPE channel 属性参数。

➤ 定义

```
typedef struct MI_VPE_ChannelPara_s
{
    MI_U8 u8NrcSfStr; //0 ~ 255;
    MI_U8 u8NrcTfStr; //0 ~ 255
    MI_U8 u8NrySfStr; //0 ~ 255
    MI_U8 u8NryTfStr; //0 ~ 255
    MI_U8 u8NryBlendMotionTh; //0 ~ 15
    MI_U8 u8NryBlendStillTh; //0 ~ 15
    MI_U8 u8NryBlendMotionWei; //0 ~ 31
    MI_U8 u8NryBlendOtherWei; //0 ~ 31
    MI_U8 u8NryBlendStillWei; //0 ~ 31
    MI_U8 u8EdgeGain[6]; //0~255
    MI_U8 u8Contrast; //0~255
} MI_VPE_ChannelPara_t;
```

➤ 成员

成员名称	描述
------	----

成员名称	描述
u8NrcSfStr	0~255, 空间域降彩噪强度
u8NrcTfStr	0~255, 时间域降彩噪强度
u8NrySfStr	0~255, 空间域降明度噪点强度
u8NryTfStr	0~255, 时间域降明度噪点强度
u8NryBlendMotionTh	0~15, 空间域明度降噪画面变动区侦测阈值
u8NryBlendStillTh	0~15, 空间域明度降噪画面静止区侦测阈值
u8NryBlendMotionWei	0~31, 画面变动区空间域降噪相对时域降噪的比重
u8NryBlendOtherWei	0~31, 静止与变动区间空间域降噪相对时域的比重
u8NryBlendStillWei	0~31, 画面静止区空间域降噪相对时域降噪的比重
u8EdgeGain[6]	根據不同邊緣程度, 做不同的銳化效果, index 0 代表发丝、草地等细小纹理, 越往大邊緣程度越大。
u8Contrast	影响暗区和亮区调整幅度, 值越大, 暗区调整越明显, 而亮度不會過曝

※ 注意事项

BLEND_WEI 数值愈大空间域去噪点强度愈强, 建议针对画面动态区设较强的值可达到较好的去噪点效果; 针对静止区设定较弱的值, 可保留较多细节。

➤ 相关数据类型及接口

[MI_VPE_ChannelPara_t](#)。

2.11. MI_VPE_HDRType_e

➤ 说明

定义 VPE 是否开 HDR, 开 HDR 是使用哪种方式。

➤ 定义

```
typedef enum
{
    E_MI_VPE_HDR_TYPE_OFF,
    E_MI_VPE_HDR_TYPE_VC,          //virtual channel mode HDR,vc0->long, vc1->short
    E_MI_VPE_HDR_TYPE_DOL,
    E_MI_VPE_HDR_TYPE_EMBEDDED, //compressed HDR mode
    E_MI_VPE_HDR_TYPE_LI,        //Line interlace HDR
    E_MI_VPE_HDR_TYPE_MAX
} MI_VPE_HDRType_e
```

※ 注意事项

目前只支持 E_MI_VPE_HDR_TYPE_DOL 这种, 只有在 E_MI_VPE_RUNNING_MODE_FRAMEBUF_CAM_MODE 和 E_MI_VPE_RUNNING_MODE_REALTIME_MODE 才可以设置打开, 为静态属性只能设置一次。

- 相关数据类型及接口

[MI_VPE_ChannelPara_t](#)。

2.12. MI_VPE_3DNR_Level_e

- 说明

定义 VPE 开启 3DNR 等级。

➤ 定义

```
typedef enum
{
    E_MI_VPE_3DNR_LEVEL_OFF,
    E_MI_VPE_3DNR_LEVEL1,
    E_MI_VPE_3DNR_LEVEL2,
    E_MI_VPE_3DNR_LEVEL3,
    E_MI_VPE_3DNR_LEVEL4,
    E_MI_VPE_3DNR_LEVEL5,
    E_MI_VPE_3DNR_LEVEL6,
    E_MI_VPE_3DNR_LEVEL7,
    E_MI_VPE_3DNR_TYPE_NUM
} MI_VPE_3DNR_Level_e;
```

※ 注意事项

Create channel 后设置，为静态属性只能设置一次。

有關 3DNR 等級，會導致部分 3DNR API 參數(NR3D_PARAM_t)無法使用，請參考下表。

【名稱】

變數名稱	LEVEL_OFF	Level 1 ~ 3	Level 4 ~ 7
u16MdThd	X	O	O
u16MdDiv	X	O	O
u8TfStr	X	O	O
u8TfStrEx	X	O	O
u16MdThdPre	X	X	O
u16MdGainPre	X	X	O
u8TfStrPre	X	X	O
u8TfStrExPre	X	X	O
u8MdThdByY[16]	X	O	O
u8MdDivByY[16]	X	O	O
u8M2SLut[16]	X	O	O
u8TfLut[16]	X	O	O
u8YSfStr	O	O	O
u8YSfBlendLut[16]	X	O	O
u8CSfStr	O	O	O
u8CSfExStr	O	O	O
u8CSfExBlendGain	X	O	O
u16CSfExBlendClip	X	O	O
u16ShpBlendLut[16]	X	O	O

➤ 相关数据类型及接口

[MI_VPE_ChannelPara_t](#)。
MI_ISP_IQ_SetNR3D, MI_ISP_IQ_GetNR3D。
MI_ISP_IQ_NR3D_TYPE_t。
NR3D_PARAM_t。

2.13. MI_VPE_ChannelPara_t

➤ 说明

定义 VPE 开启 3DNR 等级。

➤ 定义

```
typedef struct MI_VPE_ChannelPara_s
{
    MI_VPE_PqParam_t      stPqParam; // only dvr use
    MI_VPE_HDRTYPE_e      eHDRTYPE;
    MI_VPE_3DNR_Level_e   e3DNRLevel;
    MI_BOOL                bWdrEn;    //Wdr on/off
} MI_VPE_ChannelPara_t;
```

※ 注意事项

Create channel 后设置，为静态属性只能设置一次。

➤ 相关数据类型及接口

[MI_VPE_PqParam_t](#)
[MI_VPE_HDRTYPE_e](#)
[MI_VPE_3DNR_Level_e](#)

2.14. MI_VPE_RegionInfo_t

➤ 说明

定义 VPE channel 亮度统计区域信息类型。

➤ 定义

```
typedef struct MI_VPE_RegionInfo_s
{
    MI_VPE_Region_t *pstRegion; // region attribute
    MI_U32 u32RegionNum;        // count of the region
} MI_VPE_RegionInfo_t;
```

➤ 成员

成员名称	描述
------	----

pstRegion	统计区域信息
u32RegionNum	统计区域数量

※ 注意事项
无。

➤ 相关数据类型及接口
无。

2.15. MI_VPE_PortMode_t

➤ 说明
定义 VPE 端口模式。

➤ 定义

```
typedef struct MI_VPE_PortMode_s
{
    MI_U32 u32Width;                // Width of target image
    MI_U32 u32Height;               // Height of target image
    MI_SYS_PixelFormat_e ePixelFormat; // Pixel format of target image
    MI_SYS_CompressMode_e eCompressMode; // Compression mode of the output
}MI_VPE_PortMode_t;
```

➤ 成员

成员名称	描述
u32Width	端口输出画面水平大小
u32Height	端口输出画面竖直大小
ePixelFormat	端口输出 pixel format
eCompressMode	端口输出压缩模式

※ 注意事项
无

➤ 相关数据类型及接口

MI_SYS_PixelFormat_e
MI_SYS_CompressMode_e

2.16. MI_VPE_ISPBScaler_t

➤ 说明

定义 VPE ISP Bayer Scaler 参数属性。

➤ 定义

```
typedef struct MI_VPE_ISPScaler_s
{
    MI_SYS_WindowRect_t  stInCropInfo;
    MI_U16                u16OutWidth;
    MI_U16                u16OutHeight;
    MI_U32                u32Factor_Wm;
    MI_U32                u32Factor_Wn;
    MI_U32                u32Factor_Hm;
    MI_U32                u32Factor_Hn;
}MI_VPE_ISPScaler_t;
```

➤ 成员

成员名称	描述
stInCropInfo	Input Crop 位置设置
u16OutWidth	端口输出画面水平大小
u16OutHeight	端口输出画面竖直大小
u32Factor_Wm;u32Factor_Wn;	水平缩放比列调整。inputW *(Wn/Wm)=output W。 默认情况下都为 0，driver 使用默认设置。
u32Factor_Hm;u32Factor_Hn;	竖直缩放比列调整。inputH *(Hn/Hm)=output H。 默认情况下都为 0，driver 使用默认设置。

※ 注意事项

output W/H 小于等于 InCrop, Bayer Scaler 只能 scaling down, 不能 scaling up

➤ 相关数据类型及接口

```
MI_SYS_WindowRect_t
MI\_VPE\_ChannelIspParam\_t
```

2.17. MI_VPE_IspParamType_e

➤ 说明

定义 VPE ISP 参数设置类型。

➤ 定义

```
typedef enum
{
    E_MI_VPE_ISPPARAM_NULL,
    E_MI_VPE_ISPPARAM_BSCALER,
    E_MI_VPE_ISPPARAM_MAX,
}MI_VPE_IspParamType_e
```

※ 注意事项
无

➤ 相关数据类型及接口
[MI_VPE_ChannelIspParam_t](#)

2.18. [MI_VPE_ChannelIspParam_t](#)

➤ 说明
设置 VPE ISP 参数。

➤ 定义

```
typedef struct MI_VPE_ChannelIspParam_s
{
    MI\_VPE\_IspParamType\_e eIspParamType;
    union
    {
        MI\_VPE\_ISPBScaler\_t stBScalerSize;
    };
}MI_VPE_ChannelIspParam_t;
```

➤ 成员

成员名称	描述
eIspParamType	Isp 参数类型设置
stBScalerSize	Bayer scaler 参数设置

※ 注意事项
无

➤ 相关数据类型及接口
[MI_VPE_IspParamType_e](#)
[MI_VPE_ISPBScaler_t](#)

2.19. [MI_VPE_IspInitPara_t](#)

➤ 说明
定义 VPE ISP 初始参数。

➤ 定义

```
typedef struct MI_VPE_IspInitPara_s
{
```

```
MI_U16 u16Fps;  
MI_U16 u16Flicker;  
MI_U32 u32Shutter;  
MI_U32 u32SensorGainX1024;  
MI_U32 u32DigitalGain;  
}MI_VPE_IspInitPara_t;
```

➤ 成员

成员名称	描述
u16Fps	sensor 帧率。
u16Flicker	灯光频率抗闪烁设定。 [0] off, [1]50hz, [2]60hz, [3]auto
u32Shutter	快门时间(usec)。 取值范围: 1 ~ 1000000
u32SensorGainX1024	sensor增益。 取值范围: 1024 ~ 1024*4096
u32DigitalGain	Isp 增益。 取值范围: 1024~1024*1024

※ 注意事项

如果 u16Fps 为 0 , MI_VPE_IspInitPara_t 的所有参数都会使用默认值。
只支持 SSC323, SSC325, SSC325DE, SSC327, SSC327DE, SSC327Q。

➤ 相关数据类型及接口

[MI_VPE_ChannelAttr_t](#)

3. 错误码

VPE API 错误码如表 3-1 所示：

表 3-1 VPE API 错误码

错误代码	宏定义	描述
0xA0078001	MI_ERR_VPE_INVALID_DEVID	Chanel port 号无效
0xA0078002	MI_ERR_VPE_INVALID_CHNID	Chanel 通道号无效
0xA0078003	MI_ERR_VPE_ILLEGAL_PARAM	Chanel 参数设置无效
0xA0078004	MI_ERR_VPE_EXIST	Chanel port 已创建
0xA0078005	MI_ERR_VPE_UNEXIST	Chanel port 未创建
0xA0078006	MI_ERR_VPE_NULL_PTR	输入参数空指针错误
0xA0078008	MI_ERR_VPE_NOT_SUPPORT	操作不支持
0xA0078009	MI_ERR_VPE_NOT_PERM	操作不允许
0xA007800C	MI_ERR_VPE_NOMEM	分配内存失败
0xA007800D	MI_ERR_VPE_NOBUF	分配 BUF 池失败
0xA007800E	MI_ERR_VPE_BUF_EMPTY	图像队列为空
0xA0078010	MI_ERR_VPE_NOTREADY	Chanel 系统未初始化
0xA0078012	MI_ERR_VPE_BUSY	Chanel 系统忙