

MI VPE API

Version 2.09

© 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"> Added MI_VPE_SetPortShowPosition 	03/20/2019
2.06	<ul style="list-style-type: none"> Added MI_VPE_IspInitPara_t 	05/17/2019
2.07	<ul style="list-style-type: none"> Added MI_VPE_AllocIspDataBuf Added MI_VPE_FreeIspDataBuf 	05/31/2019
2.08	<ul style="list-style-type: none"> Added MI_VPE_LDCEndViewConfig Added MI_VPE_LDCEndViewConfig Added MI_VPE_LDCEndViewConfig 	09/16/2019
2.09	<ul style="list-style-type: none"> Added shutter/gain short to MI_VPE_IspInitPara_t 	11/08/2019

TABLE OF CONTENTS

REVISION HISTORY	i
TABLE OF CONTENTS.....	ii
1. SCOPE	1
1.1. Module description.....	1
1.2. Flow chart.....	1
1.2.1 328Q/329D/326D Flow chart	1
1.2.2 325/325DE/327DE Flow chart	1
1.2.3 336D/336Q/339G Flow chart.....	1
1.2.4 335/337DE 框图	2
1.3. Keyword.....	2
2. API LIST	4
2.1. MI_VPE_CreateChannel.....	5
2.2. MI_VPE_DestroyChannel	8
2.3. MI_VPE_GetChannelAttr	9
2.4. MI_VPE_SetChannelAttr	9
2.5. MI_VPE_StartChannel	10
2.6. MI_VPE_StopChannel.....	11
2.7. MI_VPE_EnablePort	11
2.8. MI_VPE_DisablePort.....	12
2.9. MI_VPE_SetChannelParam	13
2.10. MI_VPE_GetChannelParam	13
2.11. MI_VPE_SetChannelCrop.....	14
2.12. MI_VPE_GetChannelCrop	15
2.13. MI_VPE_GetChannelRegionLuma	16
2.14. MI_VPE_SetChannelRotation	16
2.15. MI_VPE_GetChannelRotation	18
2.16. MI_VPE_SetPortMode	19
2.17. MI_VPE_GetPortMode	20
2.18. MI_VPE_SetPortCrop.....	21
2.19. MI_VPE_GetPortCrop	23
2.20. MI_VPE_SetPortShowPosition	23
2.21. MI_VPE_GetPortShowPosition.....	24
2.22. MI_VPE_Alloc_IspDataBuf	25
2.23. MI_VPE_Free_IspDataBuf	26
2.24. MI_VPE_LDCBegViewConfig	27
2.25. MI_VPE_LDCEndViewConfig	27
2.26. MI_VPE_LDCSetViewConfig	28
2.27. MI_VPE_LDCEndViewConfig	29
2.28. MI_VPE_SkipFrame.....	30
3. VPE data type.....	31
3.1. MI_VPE_CHANNEL.....	32
3.2. MI_VPE_PORT	32
3.3. MI_VPE_RunningMode_e	32

3.4.	MI_VPE_SensorChannel_e.....	33
3.5.	MI_VPE_ChnPortMode_e.....	34
3.6.	MI_VPE_IspApiHeader_t	34
3.7.	MI_VPE_ChannelAttr_t.....	35
3.8.	MI_VPE_PqParam_t.....	37
3.9.	MI_VPE_HDRTYPE_e.....	38
3.10.	MI_VPE_3DNR_Level_e.....	39
3.11.	MI_VPE_ChannelPara_t.....	40
3.12.	MI_VPE_RegionInfo_t	41
3.13.	MI_VPE_PortMode_t	42
3.14.	MI_VPE_IspInitPara_t	43
4.	Error code	44

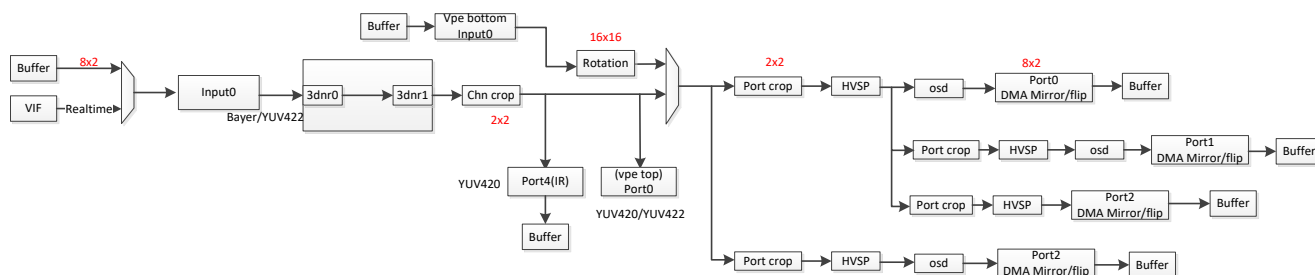
1. SCOPE

1.1. Module Description

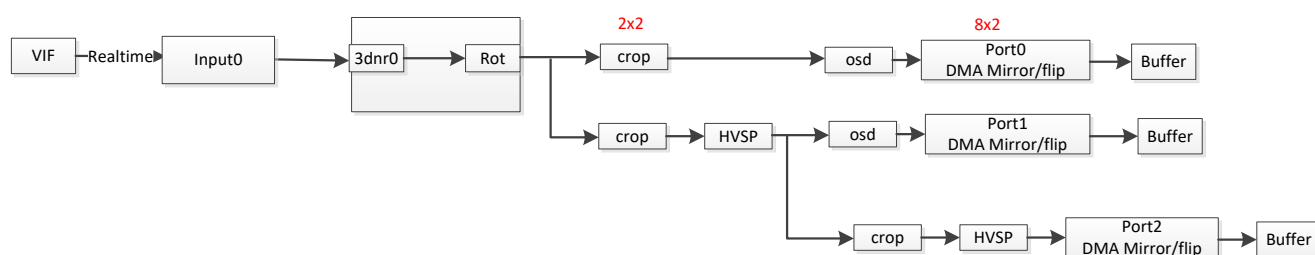
VPE (video process engine) image processing engine supports image quality adjustment for an input image, including noise reduction, sharpening, brightness adjustment, etc. After the image quality adjustment, it zooms the image to a certain resolution to output through each output port. The module also includes HDR, rotation, cutting and other functions.

1.2. Flow Chart

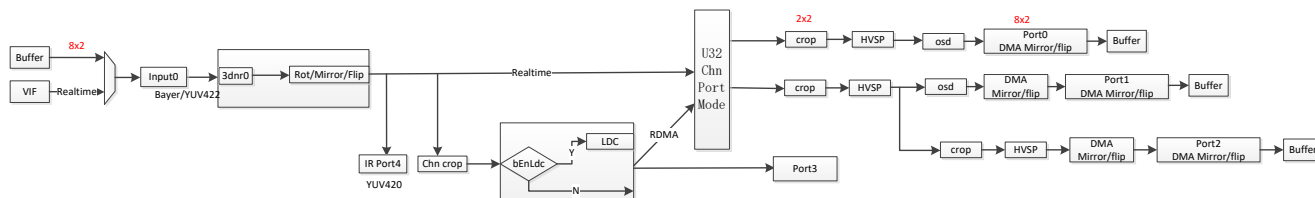
1.2.1 328Q/329D/326D Flow Chart



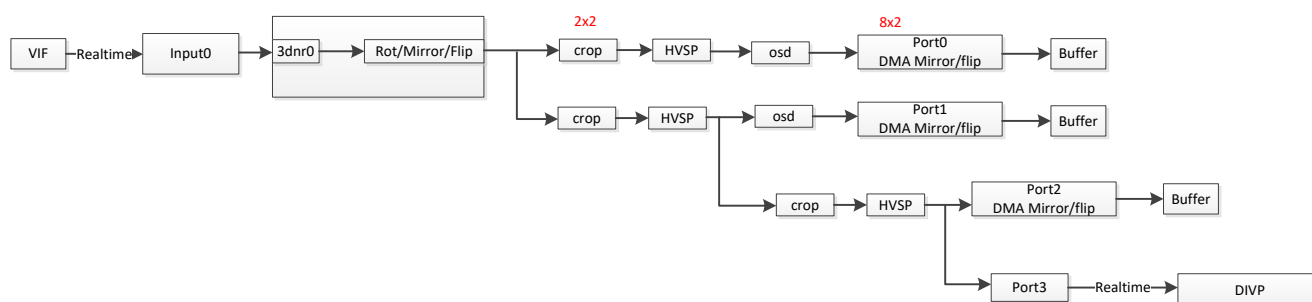
1.2.2 325/325DE/327DE Flow Chart



1.2.3 336D/336Q/339G Flow Chart



1.2.4 335/337DE Flow Chart



※ Note:

In the flow chart, 8x2, 2x2 and 16x16 indicate the width/height alignment limit of this position.

1.3. Keyword

- Channel
VPE module processes channels, and each channel's time-sharing multiplexing VPE hardware.
- ISP
Image signal processing unit is responsible for image noise reduction, color rendering, brightness adjustment and other functions.
- SCL
Short name for scaler.
- Port
Port. VPE includes an input port, and the distribution of output port refers to the flow chart.
- 3DNR
3D noise reduction.

2D noise reduction: average one pixel with the surrounding pixels, and reduce the noise after average, with the disadvantage that the image will be blurred.

3D noise reduction: add time-domain processing. 2D noise reduction only considers one frame of image, while 3D noise reduction further considers the time-domain relationship between frames, averaging each pixel in the time domain.
- HDR
High dynamic range.
- Rotation
Rotate the original image around the center point by 0/90/180/270°.
- Crop
Crop the input image.

- HVSP
Scaling process in H/V direction.
- LDC
Lens distortion correction.
- ZOOM
Fast zoom in/out, electronic zoom function.
- View
Window, a window in the internal LDC function.
- Realtime
Direct hardware connection between two modules
Advantages: DRAM is not consumed between modules
Disadvantage: Time-sharing is not supported; only one channel can be created at a time.

2. API LIST

This module provides the following APIs:

Name of API	Function
MI_VPE_CreateChannel	Create a VPE channel
MI_VPE_DestroyChannel	Destroy a VPE channel
MI_VPE_GetChannelAttr	Get a VPE channel attribute
MI_VPE_SetChannelAttr	Set a VPE channel attribute
MI_VPE_StartChannel	Start VPE channel
MI_VPE_StopChannel	Stop VPE channel
MI_VPE_EnablePort	Enable VPE port
MI_VPE_DisablePort	Disable VPE port
MI_VPE_SetChannelParam	Set VPE channel parameter
MI_VPE_GetChannelParam	Get VPE channel parameter
MI_VPE_SetChannelCrop	Set VPE channel crop window
MI_VPE_GetChannelCrop	Get VPE channel crop window
MI_VPE_GetChannelRegionLuma	Get VPE channel Luma histogram statistic
MI_VPE_SetChannelRotation	Set VPE channel video rotation type
MI_VPE_GetChannelRotation	Get VPE channel video rotation type
MI_VPE_SetPortMode	Set VPE port mode
MI_VPE_GetPortMode	Get VPE port mode
MI_VPE_SetPortCrop	Set VPE out port crop window Parameters
MI_VPE_GetPortCrop	Get VPE out port crop window Parameters
MI_VPE_SetPortShowPosition	Set VPE Output port show position
MI_VPE_GetPortShowPosition	Get VPE Output port show position
MI_VPE_Alloc_IspDataBuf	Allocate MI_ISP API data buffer
MI_VPE_Free_IspDataBuf	Free MI_ISP API data buffer
MI_VPE_LDCBegViewConfig	View start of the configuration
MI_VPE_LDCEndViewConfig	View end of the configuration
MI_VPE_LDCSetViewConfig	View the config bin buffer
MI_VPE_SkipFrame	Set skip frame num

2.1. MI_VPE_CreateChannel

➤ Description

Create a VPE channel.

➤ Syntax

```
MI_S32 MI_VPE_CreateChannel(MI\_VPE\_CHANNEL VpeCh,  
MI\_VPE\_ChannelAttr\_t*pstVpeChAttr);
```

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
pstVpeChAttr	VPE channel attribute pointer.	Input

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

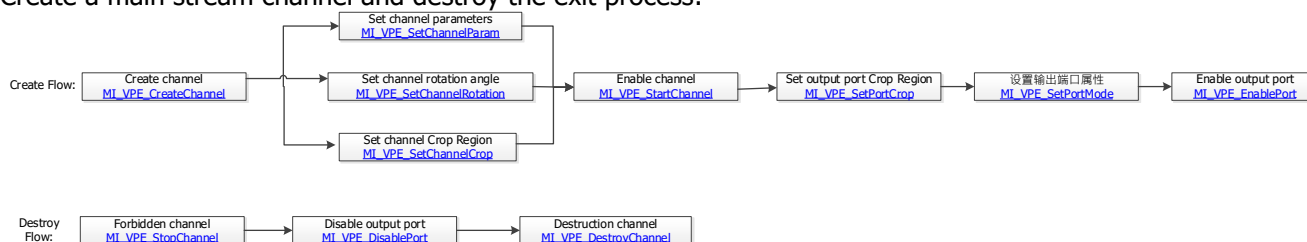
- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

➤ Note

- Multi channel is not supported in realtime mode.

➤ Example

Create a main stream channel and destroy the exit process:



```

MI\_VPE\_ChannelAttr\_t stChannelVpeAttr;
MI\_SYS\_WindowRect\_t stChnCropWin;
MI\_VPE\_CHANNEL VpeChannel=0;
MI\_VPE\_PORT VpePort=0;
MI\_VPE\_ChannelPara\_t stChannelVpeParam;
MI\_SYS\_Rotate\_e eRot = E\_MI\_SYS\_ROTATE\_NONE;
MI\_SYS\_WindowRect\_t stPortCropWin;
MI\_S32 s32Ret = MI\_SUCCESS;

MI\_SNR\_PAD\_ID\_e eSnrPadId = E\_MI\_SNR\_PAD\_ID\_0;
MI\_SNR\_PlaneInfo\_t stSnrPlane0Info;
MI\_U32 u32CapWidth = 0, u32CapHeight = 0;
MI\_SYS\_PixelFormat\_e ePixFormat;
  
```

```

memset(&stChannelVpeAttr, 0x0, sizeof(MI_VPE_ChannelAttr_t));
memset(&stChannelVpeParam, 0x0, sizeof(MI_VPE_ChannelPara_t));
memset(&stSnrPlane0Info, 0x0, sizeof(MI_SNR_PlaneInfo_t));
memset(&stPortCrop, 0x0, sizeof(MI_SYS_WindowRect_t));
memset(&stChnCropWin, 0x0, sizeof(MI_SYS_WindowRect_t));

MI_SNR_GetPlaneInfo(eSnrPadId, 0, &stSnrPlane0Info);

u32CapWidth = stSnrPlane0Info.stCapRect.u16Width;
u32CapHeight = stSnrPlane0Info.stCapRect.u16Height;
ePixFormat = (MI_SYS_PixelFormat_e)RGB_BAYER_PIXEL(stSnrPlane0Info.ePixPrecision,
stSnrPlane0Info.eBayerId);

stChannelVpeAttr.u32MaxW = u32CapWidth ;
stChannelVpeAttr.u32MaxH = u32CapHeight ;
stChannelVpeAttr.bNREn= FALSE;
stChannelVpeAttr.bEdgeEn= FALSE;
stChannelVpeAttr.bESEN= FALSE;
stChannelVpeAttr.bContrastEn= FALSE;
stChannelVpeAttr.bUVInvert= FALSE;
stChannelVpeAttr.ePixFmt = ePixFormat;
stChannelVpeAttr.eRunningMode = E_MI_VPE_RUN_REALTIME_MODE;
stChannelVpeAttr.eSensorBindId= E_MI_VPE_SENSOR0;

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

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

stChannelVpeParam.eHDRType = E_MI_VPE_HDR_TYPE_OFF;
stChannelVpeParam.e3DNRLLevel = E_MI_VPE_3DNRL_LEVEL2;
stChannelVpeParam.bMirror = FALSE;
stChannelVpeParam.bFlip = FALSE;
s32Ret = MI\_VPE\_SetChannelParam(VpeChannel, &stChannelVpeParam);
if(s32Ret != MI_SUCCESS)
{
    return s32Ret;
}

s32Ret = MI\_VPE\_SetChannelRotation(VpeChannel, eRot);
if(s32Ret != MI_SUCCESS)
{
    return s32Ret;
}

s32Ret = MI\_VPE\_GetChannelCrop(VpeChannel, &stCropWin) ;
if(s32Ret != MI_SUCCESS)

```

```

{
    return s32Ret;
}

stChnCropWin .u16X = 0;
stChnCropWin .u16Y = 0;
stChnCropWin .u16Width = 0;
stChnCropWin .u16Height = 0;
s32Ret = MI\_VPE\_SetChannelCrop(VpeChannel, & stChnCropWin );
if(s32Ret != MI_SUCCESS)
{
    return s32Ret;
}

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

stPortCrop.u16X = 0;
stPortCrop.u16Y = 0;
stPortCrop.u16Width = 0;
stPortCrop.u16Height = 0;
s32Ret=MI\_VPE\_SetPortCrop(VpeChannel,VpePort,&stPortCrop);
if(s32Ret != MI_SUCCESS)
{
    return s32Ret;
}

stVpeMode.u16Width = u32CapWidth;
stVpeMode.u16Height = u32CapHeight;
stVpeMode.ePixelFormat = E_MI_SYS_PIXEL_FRAME_YUV_SEMIPLANAR_420;
stVpeMode.eCompressMode = E_MI_SYS_COMPRESS_MODE_NONE;
stVpeMode.bMirror = FALSE;
stVpeMode.bFlip = FALSE;
s32Ret = MI\_VPE\_SetPortMode(VpeChannel,VpePort, &stVpeMode);
if(s32Ret != MI_SUCCESS)
{
    return s32Ret;
}

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

/*****
/*  call sys bind interface */
*****/

/*****/

```

```

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

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

s32Ret = MI\_VPE\_DestroyChannel(VpeChannel);
if(s32Ret != MI_SUCCESS)
{
    return s32Ret;
}

```

➤ Related APIs

[MI_VPE_DestroyChannel](#)

2.2. [MI_VPE_DestroyChannel](#)

➤ Description

Destroy a VPE channel.

➤ Syntax

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

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number. s	Input

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

➤ Note

None

➤ Example

Refer to [MI VPE CreateChannel](#)

➤ Related APIs

[MI VPE CreateChannel](#)

2.3. MI_VPE_GetChannelAttr

➤ Description

Get a VPE channel attribute

➤ Syntax

```
MI_S32 MI_VPE_GetChannelAttr(MI\_VPE\_CHANNEL VpeCh, MI\_VPE\_ChannelAttr\_t
*pstGrpAttr);
```

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
pstVpeChAttr	VPE channel attribute pointer.	Output

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

➤ Note

None

➤ Example

Refer to [MI VPE CreateChannel](#)

➤ Related APIs

[MI VPE SetChannelAttr](#)

2.4. MI_VPE_SetChannelAttr

➤ Description

Set a VPE channel attribute.

➤ Syntax

```
MI_S32 MI_VPE_SetChannelAttr(MI\_VPE\_CHANNEL VpeCh, MI\_VPE\_ChannelAttr\_t
*pstVpeChAttr);
```

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number. Value range: [0, MI_VPE_MAX_CHANNEL_NUM].	Input
pstVpeChAttr	VPE channel attribute pointer.	Input

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

➤ Note

Only bNrEn, bEdgeEn, bEsEn, bContrastEn and bUVInvert parameters in MI_VPE_ChannelAttr_t can be changed in DVR mode.

➤ Example

Refer to [MI_VPE_CreateChannel](#)

➤ Related APIs

[MI_VPE_GetChannelAttr](#)

2.5. MI_VPE_StartChannel

➤ Description

Start VPE channel.

➤ Syntax

```
MI_S32 MI_VPE_StartChannel(MI_VPE_CHANNEL VpeCh);
```

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number. Value range: [0, MI_VPE_MAX_CHANNEL_NUM].	Input

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

- Note
None
- Example
Refer to [MI VPE CreateChannel](#)
- Related APIs
[MI VPE StopChannel](#)

2.6. MI_VPE_StopChannel

- Description
Stop VPE channel
- Syntax

```
MI_S32 MI_VPE_StopChannel(MI VPE CHANNEL VpeCh);
```
- Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
- Return Value
 - Zero: Successful
 - Non-zero: Failed, see error code for details
- Dependence
 - Header file: mi_vpe.h
 - Lib: libmi_vpe.a/libmi_vpe.so
- Note
When changing the channel property, you need first to call the API, disable the channel, and then turn it on again after the channel property is set.
- Example
Refer to [MI VPE CreateChannel](#)
- Related APIs
[MI VPE StartChannel](#)

2.7. MI_VPE_EnablePort

- Description
Enable VPE port.
- Syntax

```
MI_S32 MI_VPE_EnablePort(MI VPE CHANNEL VpeCh, MI VPE PORT s32VpePort);
```


➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
s32VpePort	s32VpePort number.	Input

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

➤ Note

None

➤ Example

Refer to [MI VPE CreateChannel](#)

➤ Related APIs

[MI VPE DisablePort](#)

2.8. MI_VPE_DisablePort

➤ Description

Disable VPE port.

➤ Syntax

MI_S32 MI_VPE_DisablePort([MI VPE CHANNEL](#) VpeCh, [MI VPE PORT](#) VpePort);

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
VpePort	32VpePort number.	Input

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

➤ Note

When changing the port property, you need first to disable the port, and then enable it again after setting the port property.

- Example
Refer to [MI VPE CreateChannel](#)

- Related APIs
[MI VPE EnablePort](#)

2.9. MI_VPE_SetChannelParam

- Description
Set VPE channel parameter.
- Syntax
`MI_S32 MI_VPE_SetChannelParam (MI_VPE_CHANNEL VpeCh, MI_VPE_ChannelPara_t *pstVpeParam);`

- Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
pstVpeParam	Channel parameters	Input

- Return Value
 - Zero: Successful
 - Non-zero: Failed, see error code for details
- Dependence
 - Header file: `mi_vpe.h`
 - Lib: `libmi_vpe.a/libmi_vpe.so`
- Note
 - Refer to [MI_VPE_ChannelPara_t](#) for `pstVpeParam` after channel creation.
 - Set this API after getting present parameters by [MI_VPE_GetChannelParam](#).
- Example
None
- Related APIs
[MI_VPE_GetChannelParam](#)

2.10. MI_VPE_GetChannelParam

- Description
Get VPE channel parameter.

➤ Syntax

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

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
pstVpeParam	Channel parameters	Output

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

➤ Note

Refer to [MI_VPE_ChannelPara_t](#) for pstVpeParam after channel creation.

➤ Example

None

➤ Related APIs

[MI_VPE_SetChannelParam](#)

2.11. [MI_VPE_SetChannelCrop](#)

➤ Description

Set VPE channel crop window.

➤ Syntax

```
MI_S32 MI_VPE_SetChannelCrop (MI\_VPE\_CHANNEL VpeCh, MI\_SYS\_WindowRect\_t
*pstCropInfo);
```

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number. Value range: [0, MI_VPE_MAX_CHANNEL_NUM].	Input
pstCropInfo	Channel Crop Window parameter	Input

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

➤ Note

Chip	Supported or not
328Q/329D/326D	Supported
325/325DE/327DE	Not supported
336D/336Q/339G	Supported in LDC/Zoom scenarios
335/337DE	Not supported

- The channel has been created successfully. The settings of the crop window are all based on the original screen size.

➤ Example

None

➤ Related APIs

[MI_VPE_GetChannelCrop](#)

2.12. MI_VPE_GetChannelCrop

➤ Description

Get VPE channel crop window.

➤ Syntax

```
MI_S32 MI_VPE_GetChannelCrop(MI\_VPE\_CHANNEL VpeCh, MI_SYS_WindowRect_t *pstCropInfo);
```

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
pstCropInfo	Channel Crop Window parameter	Output

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

➤ Note

Get VPE channel crop window after channel creation.

- Example
None

- Related APIs
[MI_VPE_SetChannelCrop](#)

2.13. MI_VPE_GetChannelRegionLuma

- Description
Get VPE channel Luma histogram statistics.

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

- Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
pstRegionInfo	Indicate region and number of statistics.	Input
pu32LumaData	Statistical data	Output
s32MilliSec	Timeout (ms) of this API	Input

- Return Value
 - Zero: Successful
 - Non-zero: Failed, see error code for details
- Dependence
 - Header file: mi_vpe.h
 - Lib: libmi_vpe.a/libmi_vpe.so
- Note
 - The API is only supported by SAV538E/S, SAV638E/S, and SAV838E/S.
 - Channel has been created.

- Example
None

- Related APIs
None

2.14. MI_VPE_SetChannelRotation

- Description
Set VPE channel video rotation type.

➤ Syntax

MI_S32 MI_VPE_SetChannelRotation ([MI VPE CHANNEL](#) VpeCh, MI_SYS_Rotate_e eType);

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
eType	Rotation angle	Input

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

➤ Note

Chip	Usage Mode
328Q/329D/326D	The two channel running modes match with top and bottom respectively. The API is used on the bottom channel.
325/325DE/327DE	Need Sensor Mirror/Flip Coordination: Rot 90 + Sensor Mirror Rot 180 + Sensor Mirror/Flip Rot 270 + Sensor Flip
336D/336Q/339G	Simple use
335/337DE	Simple use

➤ Example

The following is an example use of the 328Q/329D/326D Rotation feature:

```
stVpeChannelInfo.eRunningMode = E_MI_VPE_RUN_REALTIME_TOP_MODE;
stVpeChannelInfo.eBindSensorId = E_MI_VPE_SENSOR0;
stVpeChannelInfo.bRotation = TRUE;
STCHECKRESULT(MI VPE CreateChannel(vpechn_top, &stVpeChannelInfo));
STCHECKRESULT(MI VPE StartChannel(vpechn_top));

VpePort=0 ; //only can use port0
STCHECKRESULT(MI VPE SetPortMode(vpechn_top, VpePort, &stVpeMode));
STCHECKRESULT(MI VPE EnablePort(vpechn_top, VpePort));

stVpeChannelInfo.eRunningMode = E_MI_VPE_RUN_REALTIME_BOTTOM_MODE;
stVpeChannelInfo.eBindSensorId = E_MI_VPE_SENSOR_INVALID;
stVpeChannelInfo.bRotation = TRUE;
STCHECKRESULT(MI VPE CreateChannel(vpechn_bot, &stVpeChannelInfo));
STCHECKRESULT(MI VPE StartChannel(vpechn_bot));

STCHECKRESULT(MI VPE SetChannelRotation(vpechn_bot, E_MI_SYS_ROTATE_90));
```

```
VpePort=2 ; //0~3
STCHECKRESULT(MI_VPE_SetPortMode(vpechn_bot, VpePort, &stVpeMode));
STCHECKRESULT(MI_VPE_EnablePort(vpechn_bot, VpePort));

/*****
/* call sys bind interface
   Bind vpechn_top vpechn_bot*/
*****/
```

➤ Related APIs

[MI_VPE_GetChannelRotation](#)

2.15. MI_VPE_GetChannelRotation

➤ Description

Get VPE channel video rotation type.

➤ Syntax

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

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
peType	Rotation angle	Output

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

➤ Note

Get VPE channel video rotation type after channel creation

➤ Example

None

➤ Related APIs

[MI_VPE_SetChannelRotation](#)

2.16. MI_VPE_SetPortMode

➤ Description

Set VPE port mode.

➤ Syntax

```
MI_S32 MI_VPE_SetPortMode(MI\_VPE\_CHANNEL VpeCh, MI\_VPE\_PORT VpePort,
MI\_VPE\_PortMode\_t *pstVpeMode);
```

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
VpePort	VPE port number.	Input
pstVpeMode	VPE port mode	Output

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: [mi_vpe.h](#)
- Lib: [libmi_vpe.a/libmi_vpe.so](#)

➤ Note

Max width/height

Chip	Port0	Port1	Port2	Port3	Port4 (IR port)
328Q/329D/ 326D	MaxWidth = 3840	MaxWidth = 2688	MaxWidth = 2688	MaxWidth = 3840	No Scaling: Width= 1/2 SrcWidth Height= 1/2 SrcHeigh
325/325DE/ 327DE	No Scaling: Width= SrcWidth Height=SrcHeigh	MaxWidth = 2688	MaxWidth = 2688	No support	No support
336D/336Q/ 339G	MaxWidth = 3840	MaxWidth = 3840	MaxWidth = 3840	No Scaling: Width= SrcWidth Height=SrcHeigh h	No Scaling: Width= 1/2 SrcWidth Height= 1/2 SrcHeigh
335/337DE	MaxWidth = 2688	MaxWidth = 2688	MaxWidth = 1920	MaxWidth = 1920	No support

Pixel Format

Chip	Port0	Port1	Port2	Port3	Port4 (IR port)
328Q/329D/326D	YUV420/YUV422/ ARGB8888/BGRA8888	Same as Port0	Same as Port0	Same as Port0	YUV420 NV12
325/325DE/327DE	YUV420/YUV422	Same as Port0	Same as Port0	Same as Port0	No support
336D/336Q/339G	YUV420/YUV422/ ARGB8888/BGRA8888/ABGR8888	Same as Port0	Same as Port0	No Ldc/Zoom: YUV420/YUV422 Ldc/Zoom:	YUV420 NV12
335/337DE	YUV420/YUV422	Same as Port0	Same as Port0	Same as Port0	No support

➤ Example

None

➤ Related APIs

[MI_VPE_GetPortMode](#)

2.17. MI_VPE_GetPortMode

➤ Description

Get VPE port mode.

➤ Syntax

```
MI_S32 MI_VPE_GetPortMode (MI_VPE_CHANNEL VpeCh, MI_VPE_PORT VpePort,
MI_VPE_PortMode_t *pstVpeMode);
```

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
VpePort	VPE port number.	Input
pstVpeMode	VPE port mode	Output

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

- Note
 - None
- Example
 - None
- Related APIs
 - [MI_VPE_SetPortMode](#)

2.18. MI_VPE_SetPortCrop

- Description
 - Set VPE out port crop window.
- Syntax


```
MI_S32 MI_VPE_SetPortCrop (MI\_VPE\_CHANNEL VpeCh, MI\_VPE\_PORT VpePort,
MI_SYS_WindowRect_t *pstOutCropInfo);
```

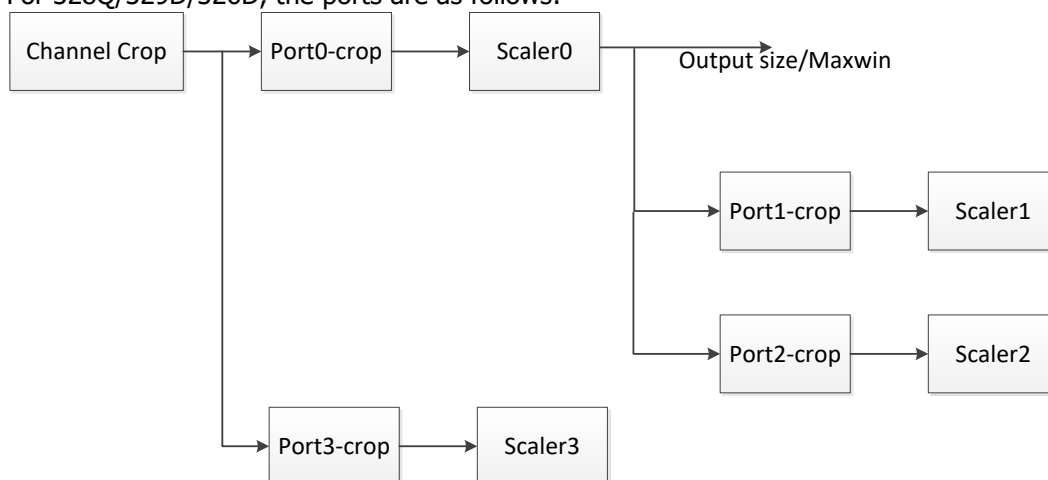
- Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
VpePort	VPE port number.	Input
*pstOutCropInfo	Output port crop window setting	Input

- Return Value
 - Zero: Successful
 - Non-zero: Failed, see error code for details
- Dependence
 - Header file: mi_vpe.h
 - Lib: libmi_vpe.a/libmi_vpe.so

➤ Note

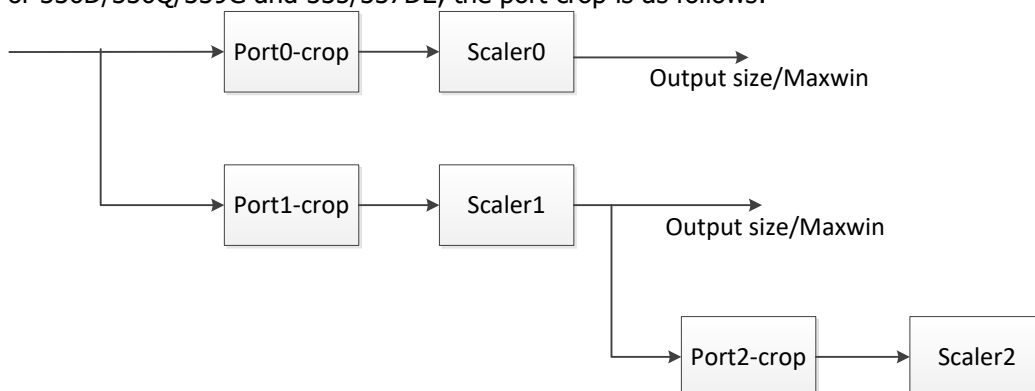
- For 328Q/329D/326D, the ports are as follows:



Because the source of port1,2 is the output of port0:

1. The cross win of port0 enable, port1,2 < port0 size,
2. Cross win < VPE input of port0 disable, port1,2

- For 336D/336Q/339G and 335/337DE, the port crop is as follows:



325/325DE/327DE do not have the scaler0 shown in the figure above; the others are consistent.

Because the source of port2 is the output of port1:

1. Port1 enable, cross win < port1 size of port2
2. Port1 disable, cross win < VPE input of port2

➤ Example

None

➤ Related APIs

[MI_VPE_GetPortCrop](#)

2.19. MI_VPE_GetPortCrop

➤ Description

Get VPE out port crop window parameters.

➤ Syntax

```
MI_S32 MI_VPE_GetPortCrop (MI_VPE_CHANNEL VpeCh, MI_VPE_PORT VpePort,
MI_SYS_WindowRect_t *pstOutCropInfo);
```

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
VpePort	VPE port number.	Input
*pstOutCropInfo	Output port crop window parameter	Output

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

➤ Note

Only SSC329Q, SSC326D, SSA520, and SSC328Q support this API.

➤ Example

None

➤ Related APIs

[MI_VPE_SetPortCrop](#)

2.20. MI_VPE_SetPortShowPosition

➤ Description

Set VPE output port show position.

➤ Syntax

```
MI_S32 MI_VPE_SetPortShowPosition(MI_VPE_CHANNEL VpeCh, MI_VPE_PORT VpePort,
MI_SYS_WindowRect_t *pstPortPositionInfo);
```

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number	Input
VpePort	VPE port number	Input
* pstPortPositionInfo	Show position parameters	Input

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

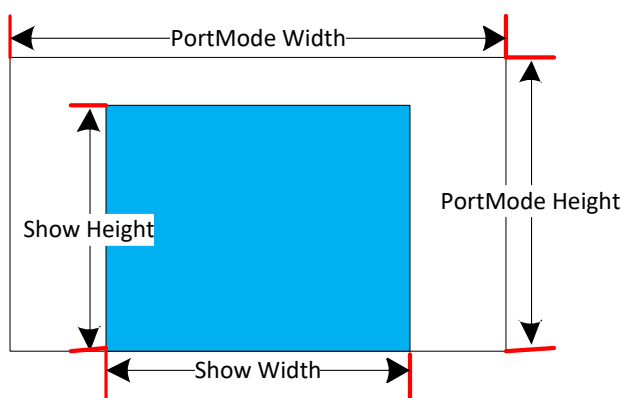
➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

※ Note

- If this API is not called, show width/height = portmode width/height is displayed by default.
- You need to set this API after MI VPE setportmode, and after that, the area without show screen will be filled in black.

The relationship between portmode width/height and show width/height is as follows:



➤ Example

None

➤ Related APIs

[MI_VPE_GetPortShowPosition](#)

2.21. MI_VPE_GetPortShowPosition

➤ Description

Get VPE output port show position.

➤ Syntax

```
MI_S32 MI_VPE_GetPortShowPosition(MI_VPE_CHANNEL VpeCh, MI_VPE_PORT VpePort,
MI_SYS_WindowRect_t *pstPortPositionInfo);
```

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number	Input
VpePort	VPE port number	Input
* pstPortPositionInfo	Show position parameters	Output

- Return Value
 - Zero: Successful
 - Non-zero: Failed, see error code for details

- Dependence
 - Header file: mi_vpe.h
 - Lib: libmi_vpe.a/libmi_vpe.so

※ Note
None

➤ Example
None

- Related APIs
[MI_VPE_SetPortShowPosition](#)

2.22. MI_VPE_Alloc_IspDataBuf

- Description
Allocate MI_ISP API Data Buffer

- Syntax
`MI_S32 MI_VPE_Alloc_IspDataBuf(MI_U32 u32Size,void **pUserVirAddr);`

- Parameters

Parameter Name	Description	Input/Output
u32Size	Allocate Buffer Size	Input
**pUserVirAddr	User Buffer pointer address	Output

- Return Value
 - Zero: Successful
 - Non-zero: Failed, see error code for details

- Dependence
 - Header file: mi_vpe.h
 - Lib: libmi_vpe.a/libmi_vpe.so

※ Note
A thread applies for an ISP data buffer to avoid buffer sharing among threads, which results in buffer stepping on each other.

➤ Example

```
#define MI_ISP_MAX_DATA_SIZE (80*1024)
MI_VPE_Alloc_IspDataBuf(MI_ISP_MAX_DATA_SIZE, &pIspBuffer);
MI_ISP_IQ_COLORTOGRAY_TYPE_t *pstColorToGray = (MI_ISP_IQ_COLORTOGRAY_TYPE_t *)pIspBuffer;
MI_ISP_IQ_GetColorToGray( Channel, pstColorToGray);
if(pstColorToGray->bEnable == SS_TRUE)
    pstColorToGray->bEnable = SS_FALSE;
else
    pstColorToGray->bEnable = SS_TRUE;
MI_ISP_IQ_SetColorToGray( Channel, pstColorToGray);

MI_ISP_IQ_CONTRAST_TYPE_t *pstContrast = (MI_ISP_IQ_CONTRAST_TYPE_t *)pIspBuffer;
MI_ISP_IQ_GetContrast( Channel, pstContrast);
MI_ISP_IQ_SetContrast( Channel, pstContrast);

MI_VPE_Free_IspDataBuf(pIspBuffer)
```

➤ Related APIs

[MI_VPE_Free_IspDataBuf](#)

2.23. MI_VPE_Free_IspDataBuf

➤ Description

Buffer for releasing MI_ISP API data application

➤ Syntax

MI_S32 MI_VPE_Free_IspDataBuf(void *pUserBuf);

➤ Parameters

Parameter Name	Description	Input/Output
*pUserBuf	Pointer to applied ISP API Data Buffer	Input

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

※ Note

Paired with MI_VPE_Alloc_IspDataBuf

➤ Example

See example in MI_VPE_Alloc_IspDataBuf

➤ Related APIs

[MI_VPE_Alloc_IspDataBuf](#)

2.24. MI_VPE_LDCBegViewConfig

➤ Description

Start configuring bin files for all LDC Windows

➤ Syntax

```
MI_S32 MI_VPE_LDCBegViewConfig(MI_VPE_CHANNEL VpeCh);
```

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

※ Note

- Only 336D/336Q/339G supports LDC function. benldc = true in MI_VPE_channelattr_t will enable LDC.
- Pair with MI_VPE_LDCendviewconfig.
- You need to use this API only when the number of windows changes. There is no need to call this API if only one of the window properties is to be changed.

➤ Example

```
MI_VPE_LDCBegViewConfig(vpechn);
for(i=0; i<viewnum;i++)
{
    MI_VPE_LDCSetViewConfig(vpechn, ldcBinBuffer[i], u32LdcBinSize[i]);
    free(ldcBinBuffer[i]);
}
MI_VPE_LDCEndViewConfig(vpechn);
```

➤ Related APIs

[MI_VPE_LDCEndViewConfig](#)

2.25. MI_VPE_LDCEndViewConfig

➤ Description

End the bin file for configuring all LDC Windows

➤ Syntax

```
MI_S32 MI_VPE_LDCEndViewConfig(MI_VPE_CHANNEL VpeCh)
```


➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

※ Note

- Only 336D/336Q/339G supports LDC function. benldc = true in MI_VPE_channelattr_t will enable LDC.
- Pair with MI VPE ldcendviewconfig.
- You need to use this API only when the number of windows changes. There is no need to call this API if only one of the window properties is to be changed.

➤ Example

See example in [MI VPE LDCBegViewConfig](#)

➤ Related APIs

[MI VPE LDCBegViewConfig](#)

2.26. MI_VPE_LDCSetViewConfig

➤ Description

Configure the LDC window bin file

➤ Syntax

```
MI_S32 MI_VPE_LDCSetViewConfig(MI_VPE_CHANNEL VpeCh, void *pConfigAddr, MI_U32 u32ConfigSize);
```

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
pConfigAddr	Bin buffer pointer address	Input
u32ConfigSize	Bin buffer size	Input

➤ Return Value

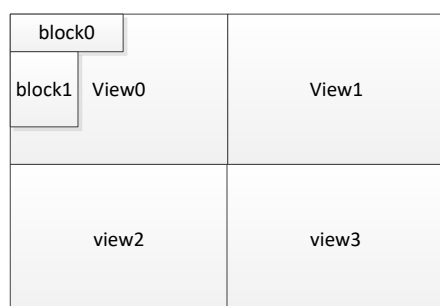
- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

※ Note

- If there is no change in the number of Windows save a change in some window bin buffer, the API can be used separately. When the number of Windows changes, it needs to be used together with MI_VPE_LDCBegViewConfig and MI_VPE_LDCEndViewConfig.



- Each config bin buffer corresponds to the settings of a view window.

➤ Example

See example in [MI_VPE_LDCBegViewConfig](#)

➤ Related APIs

[MI_VPE_LDCBegViewConfig](#)

[MI_VPE_LDCEndViewConfig](#)

2.27. MI_VPE_LDCEndViewConfig

➤ Description

End the bin file for configuring all LDC Windows

➤ Syntax

MI_S32 MI_VPE_LDCEndViewConfig(MI_VPE_CHANNEL VpeCh)

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

※ Note

- Only 336D/336Q/339G supports LDC function. benldc = true in MI_VPE_channelattr_t will enable LDC.
- Pair with MI_VPE_ldcendviewconfig.
- You need to use this API only when the number of windows changes. There is no need to call this API if only one of the window properties is to be changed.

➤ Example

See example in [MI_VPE_LDCBegViewConfig](#)

➤ Related APIs

[MI_VPE_LDCBegViewConfig](#)

2.28. MI_VPE_SkipFrame

➤ Description

Set Skip Frame Num

➤ Syntax

MI_S32 MI_VPE_SkipFrame(MI_VPE_CHANNEL VpeCh, MI_U32 u32FrameNum);

➤ Parameters

Parameter Name	Description	Input/Output
VpeCh	VPE channel number.	Input
u32FrameNum	Frame Num	Input

➤ Return Value

- Zero: Successful
- Non-zero: Failed, see error code for details

➤ Dependence

- Header file: mi_vpe.h
- Lib: libmi_vpe.a/libmi_vpe.so

※ Note

None

➤ Example

None

➤ Related APIs

None

3. VPE DATA TYPE

VPE module related data types are listed below:

<u>MI_VPE_CHANNEL</u>	Define VPE channel type
<u>MI_VPE_PORT</u>	Define VPE port type
<u>MI_VPE_RunningMode_e</u>	Define VPE running mode
<u>MI_VPE_SensorChannel_e</u>	Define VPE sensor bind ID
<u>MI_VPE_IspApiHeader_t</u>	Define VPE data header information to ISP
<u>MI_VPE_ChannelAttr_t</u>	Define VPE channel attribute
<u>MI_VPE_PqParam_t</u>	Define VPE PQ parameter (NR, EdgeGain, Contrast)
<u>MI_VPE_HDRType_e</u>	Define VPE HDR type
<u>MI_VPE_3DNR_Level_e</u>	Define 3DNR level
<u>MI_VPE_ChannelPara_t</u>	Define VPE channel parameter
<u>MI_VPE_RegionInfo_t</u>	Define VPE channel region statistics information
<u>MI_VPE_PortMode_t</u>	Define VPE port mode
<u>MI_VPE_ChnPortMode_e</u>	Define the output schema for Port

3.1. MI_VPE_CHANNEL

- Description
Define VPE channel type
- Definition
typedef MI_S32 MI_VPE_CHANNEL
- Note
None
- Related Type
None

3.2. MI_VPE_PORT

- Description
Define VPE port type
- Definition
typedef MI_S32 MI_VPE_PORT
- Note
None
- Related Type
None

3.3. MI_VPE_RunningMode_e

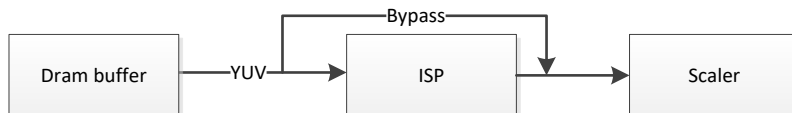
- Description
Define VPE running mode
- Definition

```
typedef enum
{
    E_MI_VPE_RUN_INVALID                = 0x00,
    E_MI_VPE_RUN_DVR_MODE                = 0x01,
    E_MI_VPE_RUN_CAM_TOP_MODE            = 0x02,
    E_MI_VPE_RUN_CAM_BOTTOM_MODE        = 0x04,
    E_MI_VPE_RUN_CAM_MODE                =
    E_MI_VPE_RUN_CAM_TOP_MODE|E_MI_VPE_RUN_CAM_BOTTOM_MODE,
    E_MI_VPE_RUN_REALTIME_TOP_MODE       = 0x08,
    E_MI_VPE_RUN_REALTIME_BOTTOM_MODE    = 0x10,
    E_MI_VPE_RUN_REALTIME_MODE           =
    E_MI_VPE_RUN_REALTIME_TOP_MODE |
    E_MI_VPE_RUN_REALTIME_BOTTOM_MODE,
    E_MI_VPE_RUNNING_MODE_MAX,
} MI_VPE_RunningMode_e;
```

➤ Note

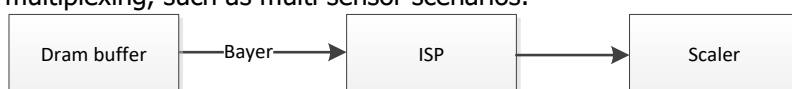
E_MI_VPE_RUN_DVR_MODE:

When the input is in YUV format, ISP bypass is not processed by ISP.



E_MI_VPE_RUN_CAM_MODE:

When the input is in Bayer format and the data comes from DRAM, ISP supports time-sharing multiplexing, such as multi sensor scenarios.

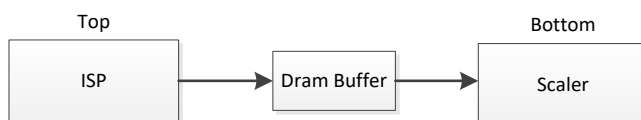


E_MI_VPE_RUN_REALTIME_MODE:

When the input is in Bayer format, the hardware between Vif and ISP is directly connected. ISP does not support time-sharing multiplexing, but only supports one channel, such as single sensor scenario.



Top/bottom mode is only supported on 328Q/329D/326D chips. It is used in rotation scenarios. Top/bottom refers to the connection between ISP and scaler.



The top channel does not have scaling capability. The bottom channel implements rotation. For the usage method, refer to the example of MI VPE setchannelrotation.

➤ Related Type

None

3.4. MI_VPE_SensorChannel_e

➤ Description

Define VPE sensor bind ID

➤ Definition

```
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;
Mapping sensor device0/1/2/3.
```

➤ Note

- Use E_MI_VPE_SENSOR_INVALID with no sensor

➤ Related Type

[MI_VPE_ChannelAttr_t](#).

3.5. MI_VPE_ChnPortMode_e

➤ Description

Define the output schema for Port

➤ Definition

```
typedef enum
{
    E_MI_VPE_ZOOM_LDC_NULL,
    E_MI_VPE_ZOOM_LDC_PORT0 = 0X01,
    E_MI_VPE_ZOOM_LDC_PORT1 = 0X02,
    E_MI_VPE_ZOOM_LDC_PORT2 = 0X04,
    E_MI_VPE_ZOOM_LDC_MAX = E_MI_VPE_ZOOM_LDC_PORT0|

    E_MI_VPE_ZOOM_LDC_PORT1|E_MI_VPE_ZOOM_LDC_PORT2,
}MI_VPE_ChnPortMode_e;
The port0,1, and 2 outputs contain zoom/ LDC effects.
```

➤ Note

Refer to notes in [MI_VPE_ChannelAttr_t](#).

➤ Related Type

[MI_VPE_ChannelAttr_t](#).

3.6. MI_VPE_IspApiHeader_t

➤ Description

Define VPE data header information to ISP

➤ Definition

```
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;
```

➤ Note

None

➤ Related Type

[MI_VPE_IspApiData_t.](#)

3.7. MI_VPE_ChannelAttr_t

➤ Description

Define VPE channel attribute

➤ Definition

```
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_BOOL bEnLdc; // true port3 for ldc or for Isp(skip pass2)
    MI_U32 u32ChnPortMode;
}MI_VPE_ChannelAttr_t;
```

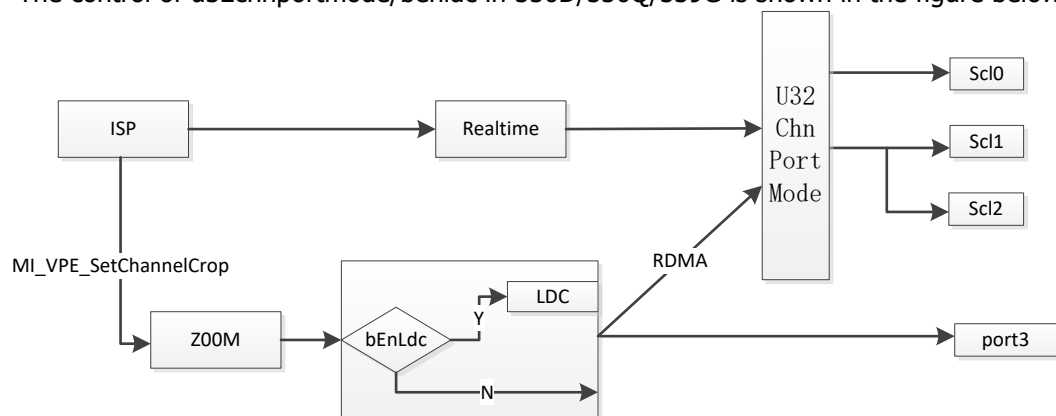
➤ Member

Member Name	Description
u32MaxW	Maximum width
u32MaxH	Maximum height
ePixFmt	Input Pixel format.
eSensorBindId	Set sensor ID boundary if sensor exists in front end
bNrEn	Enable noise reduction
bEdgeEn	Enable edge

Member Name	Description
bEsEn	Enable edge smooth
bContrastEn	Enable contrast
bUvInvert	Enable UV invert
eRunningMode	VPE running mode
bRotation	Enable rotation
tIspInitPara	ISP Init parameter
bEnLdc	Enable LDC
u32ChnPortMode	The port output mode under the current channel is assigned by the MI_VPE_ChnPortMode_e member.

➤ Note

- Set to static attribute when channel is created, cannot be modified.
- ePixFmt: When the erunningmode is E_MI_VPE_RUN_DVR_MODE, only E_MI_SYS_PIXEL_FRAME_YUV422_YUYV is supported. Other running mode pixels are obtained by sensor conversion, such as:
`ePixFormat = (MI_SYS_PixelFormat_e)RGB_BAYER_PIXEL(stSnrPlane0Info.ePixPrecision, stSnrPlane0Info.eBayerId);`
- eSensorBindId: The position of the sensor pad inserted in the front-end sensor of VPE channel. When erunningmode is E_MI_VPE_RUN_DVR_MODE, eSensorBindId = E_MI_VPE_SENSOR_INVALID.
- For bNrEn/bEdgeEn/bEsEn/bContrastEn/bUvInvert, only setting on MSR930 chip is valid.
- bRotation is only set in 328q/329d/326d series chip rotation scenario. Please refer to MI_VPE_SetChannelRotation example.
- The control of u32chnportmode/benldc in 336D/336Q/339G is shown in the figure below:



- The VPE internal flow chart is shown above:
 - 1) Normal scenario, u32ChnPortMode=0, realtime connection between ISP and all SCL, direct hardware connection, no buffer consumption.
 - 2) Scl1, 2 are the same source. Scl0 and scl1/2 can independently choose whether the source is from Realtime or RDMA through u32ChnPortMode. There are altogether 4 combinations, for example:


```
u32ChnPortMode = E_MI_VPE_ZOOM_LDC_PORT0;    //port0 from RDMA
u32ChnPortMode = E_MI_VPE_ZOOM_LDC_PORT1|
E_MI_VPE_ZOOM_LDC_PORT2; //port1,2 from RDMA
u32ChnPortMode = E_MI_VPE_ZOOM_LDC_NULL;      //all scl from realtime
u32ChnPortMode = E_MI_VPE_ZOOM_LDC_PORT0|E_MI_VPE_ZOOM_LDC_PORT1|
E_MI_VPE_ZOOM_LDC_PORT2;                      //all scl from RDMA
```
 - 3) bEnLdc==TRUE, RDMA buffer from LDC out, port3 from LDC,
bEnLdc==FALSE, RDMA buffer from ZOOM out, port3 from ZOOM.

➤ Related Type

[MI_VPE_RunningMode_e](#)

[MI_VPE_SensorChannel_e](#)

[MI_VPE_IspInitPara_t](#)

[MI_VPE_ChPortMode_e](#)

3.8. MI_VPE_PqParam_t

➤ Description

Define VPE PQ parameter (NR, EdgeGain, Contrast)

➤ Definition

```
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;
```

➤ Member

Member Name	Description
u8NrcSfStr	0~255, special field color noise reduction strength
u8NrcTfStr	0~255, time field color noise reduction strength
u8NrySfStr	0~255, special field Y noise reduction strength
u8NryTfStr	0~255, time field Y noise reduction strength
u8NryBlendMotionTh	0~15, special field noise reduction of Motion region threshold
u8NryBlendStillTh	0~15, special field noise reduction of still region threshold
u8NryBlendMotionWei	0~31, Motion region noise reduction weight in special field over time field
u8NryBlendOtherWei	0~31, Motion region and still region noise reduction weight in special field over time field
u8NryBlendStillWei	0~31, Still region noise reduction weight in special field over time field
u8EdgeGain[6]	Sharpening effect upon different edge level Index 0 indicates small texture like hair, grasslands... The more this index increases, the sharper the texture is handled
u8Contrast	Influence on dark region and bright region tuning scale. A bigger value can enhance dark region tuning effect obviously, but the brightness will not be over-exposed.

➤ Note

- When the value of BLEND_WEI is bigger, this special field noise reduction is stronger.
- It is suggested to set a bigger value for motion region to get better noise reduction effect.
- It is suggested to set a smaller value for still region to reserve more texture.

➤ Related Type

[MI_VPE_ChannelPara_t.](#)

3.9. MI_VPE_HDRType_e

➤ Description

Define VPE HDR type

➤ Definition

```
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
```

- Note

Which HDR type to use can be obtained through the MI_SNR_GetPadInfo interface.
- Related Type

[MI_VPE_ChannelPara_t](#).

3.10. MI_VPE_3DNR_Level_e

- Description

Define 3DNR level
- Definition

```
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;
```
- Note
 - Set this data type after channel creation.
 - Set this just once because of static attribute.
 - 3DNR level could cause some 3DNR API parameters (NR3D_PARAM_t) to be unavailable, see the table below:

➤ Name

Parameter Name	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

➤ Related Type

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

3.11. MI_VPE_ChannelPara_t

➤ Description

Define VPE channel parameter

➤ Definition

```
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                bMirror;
    MI_BOOL                bFlip;
    MI_BOOL                bWdrEn;    //Wdr on/off
    MI_BOOL                bEnLdc;
} MI_VPE_ChannelPara_t;
```

➤ Member

Member Name	Description
stPqParam	Pq Parameter Settings
eHDRTYPE	HDR on/off parameter
e3DNRLevel	3dnr level parameter
bMirror	Input Mirror on/off
bFlip	Input Flip on/off
bWdrEn	WDR on/off
bEnLdc	LDC on/off

➤ Note

Chip	MAX e3DNRLevel	bMirror/ bFlip Support
328Q/329D/326D	E_MI_VPE_3DNR_LEVEL7	Not supported
325/325DE/327DE	E_MI_VPE_3DNR_LEVEL2	Not supported
336D/336Q/339G	E_MI_VPE_3DNR_LEVEL2	Supported
335/337DE	E_MI_VPE_3DNR_LEVEL2	Supported

- The maximum 3dnr level supported by different chips is as shown in the table above. If the setting exceeds max, max level will be automatically used internally. The higher the level is, the stronger the 3dnr strength is, and the more buffers will be consumed.
- The supported ehdrtype can be queried through MI_SNR_GetPadInfo.
- stPqParam is only available in DVR mode
- bmirror / bflip supports the chip family as shown in the table above to prevent some sensors from not supporting flipping

➤ Related Type

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

3.12. MI_VPE_RegionInfo_t

➤ Description

Define VPE channel region statistics information

➤ Definition

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

➤ Member

Member Name	Description
pstRegion	Statistics region information
u32RegionNum	Statistics region number

➤ Note

None

➤ Related Type

None

3.13. MI_VPE_PortMode_t

➤ Description

Define VPE port mode

➤ Definition

```
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;
```

➤ Member

Member Name	Description
u32Width	Width of port output
u32Height	Height of port output
ePixelFormat	Pixel format of port output
eCompressMode	Compress mode of port output

➤ Note

Refer to the notes in MI_VPE_SetPortMode for relevant restrictions

➤ Related Type

MI_SYS_PixelFormat_e
MI_SYS_CompressMode_e

3.14. MI_VPE_IspInitPara_t

➤ Description

Define ISP Init parameter

➤ Definition

```
typedef struct MI_VPE_IspInitPara_s
{
    MI_U16 u16Fps;
    MI_U16 u16Flicker;
    MI_U32 u32Shutter;
    MI_U32 u32SensorGainX1024;
    MI_U32 u32DigitalGain;
    MI_U32 u32ShutterShort;
    MI_U32 u32GainX1024Short;
    MI_U32 u32DGainShort;
}MI_VPE_IspInitPara_t;
```

➤ Member

Member Name	Description
u16Fps	Sensor fps.
u16Flicker	Anti-flicker [0] off, [1] 50hz, [2] 60hz, [3] auto
u32Shutter	1~1000000, shutter (usec).
u32SensorGainX1024	1024~1024*4096, sensor gain.
u32DigitalGain	1024~1024*1024, ISP gain.
u32ShutterShort	1~1000000, short exposure shutter (usec).
u32GainX1024Short	1024~1024*4096, short exposure sensor gain.
u32DGainShort	1024~1024*1024, short exposure ISP gain.

➤ Note

- Members of MI_VPE_IspInitPara_t will use default value when u16Fps is 0.
- Only SSC323, SSC325, SSC325DE, SSC327, SSC327DE, and SSC327Q are supported.

➤ Related Type

[MI_VPE_ChannelAttr_t](#)

4. ERROR CODE

VPE API error code is defined in the table below:

Table 1: VPE API error code

Error Code	Macro Definition	Description
0xA0078001	MI_ERR_VPE_INVALID_DEVID	Device ID invalid
0xA0078002	MI_ERR_VPE_INVALID_CHNID	Channel ID invalid
0xA0078003	MI_ERR_VPE_ILLEGAL_PARAM	Channel parameter illegal
0xA0078004	MI_ERR_VPE_EXIST	Channel port existed
0xA0078005	MI_ERR_VPE_UNEXIST	Channel port un-existed
0xA0078006	MI_ERR_VPE_NULL_PTR	Null pointer
0xA0078008	MI_ERR_VPE_NOT_SUPPORT	Not supported
0xA0078009	MI_ERR_VPE_NOT_PERM	Not permitted
0xA007800C	MI_ERR_VPE_NOMEM	No memory
0xA007800D	MI_ERR_VPE_NOBUF	No buffer
0xA007800E	MI_ERR_VPE_BUF_EMPTY	Buffer empty
0xA0078010	MI_ERR_VPE_NOTREADY	Channel not ready
0xA0078012	MI_ERR_VPE_BUSY	Channel busy