

# MI DISP API

---

**Version 2.05**

© 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"><li>Initial release</li></ul>	04/12/2018
2.04	<ul style="list-style-type: none"><li>新增 API<ul style="list-style-type: none"><li>MI_DISP_GetLcdParam</li><li>MI_DISP_SetLcdParam</li><li>MI_DISP_DeviceGetColorTempature</li><li>MI_DISP_DeviceSetColorTempature</li><li>MI_DISP_DeviceSetGammaParam</li><li>MI_DISP_SetVideoLayerRotateMode</li></ul></li><li>数据结构修改<ul style="list-style-type: none"><li>MI_DISP_InputPortAttr_t 增加参数: MI_U16 u16SrcWidth; MI_U16 u16SrcHeight;</li></ul></li></ul>	06/06/2019
2.05	<ul style="list-style-type: none"><li>新增 API<ul style="list-style-type: none"><li>MI_DISP_DeviceAttach</li><li>MI_DISP_DeviceDetach</li></ul></li></ul>	06/19/2019

## TABLE OF CONTENTS

REVISION HISTORY .....	i
TABLE OF CONTENTS.....	ii
<b>1. 概述.....</b>	<b>1</b>
1.1. 模块说明 .....	1
1.2. 流程框图 .....	1
1.2.1. 621/623/201/202 .....	1
1.2.2. 328Q/329D/326D/336D/336Q/339G.....	2
1.3. 关键字说明.....	2
<b>2. API 参考 .....</b>	<b>3</b>
2.1. MI_DISP_Enable.....	5
2.2. MI_DISP_Disable .....	6
2.3. MI_DISP_SetPubAttr .....	7
2.4. MI_DISP_GetPubAttr.....	7
2.5. MI_DISP_DeviceAttach .....	8
2.6. MI_DISP_DeviceDetach.....	9
2.7. MI_DISP_EnableVideoLayer.....	10
2.8. MI_DISP_DisableVideoLayer .....	10
2.9. MI_DISP_SetVideoLayerAttr .....	11
2.10. MI_DISP_GetVideoLayerAttr .....	12
2.11. MI_DISP_BindVideoLayer .....	12
2.12. MI_DISP_UnBindVideoLayer .....	13
2.13. MI_DISP_SetPlayToleration .....	14
2.14. MI_DISP_GetPlayToleration.....	15
2.15. MI_DISP_GetScreenFrame .....	15
2.16. MI_DISP_ReleaseScreenFrame .....	16
2.17. MI_DISP_EnableInputPort .....	17
2.18. MI_DISP_DisableInputPort .....	18
2.19. MI_DISP_SetInputPortAttr.....	18
2.20. MI_DISP_GetInputPortAttr .....	19
2.21. MI_DISP_SetInputPortDispPos.....	20
2.22. MI_DISP_GetInputPortDispPos .....	21
2.23. MI_DISP_PauseInputPort .....	21
2.24. MI_DISP_ResumeInputPort .....	22
2.25. MI_DISP_StepInputPort .....	23
2.26. MI_DISP_ShowInputPort.....	24
2.27. MI_DISP_HideInputPort .....	25
2.28. MI_DISP_SetInputPortSyncMode .....	26
2.29. MI_DISP_QueryInputPortStat .....	26
2.30. MI_DISP_GetVgaParam.....	27
2.31. MI_DISP_SetVgaParam .....	28
2.32. MI_DISP_GetHdmiParam.....	29
2.33. MI_DISP_SetHdmiParam .....	29

2.34. MI_DISP_GetLcdParam .....	30
2.35. MI_DISP_SetLcdParam .....	31
2.36. MI_DISP_DeviceGetColorTempature .....	32
2.37. MI_DISP_DeviceSetColorTempature.....	32
2.38. MI_DISP_DeviceSetGammaParam .....	33
2.39. MI_DISP_SetVideoLayerRotateMode .....	34
2.40. MI_DISP_ClearInputPortBuffer .....	35
<b>3. DISP 数据类型 .....</b>	<b>36</b>
3.1. MI_DISP_DEV .....	37
3.2. MI_DISP_LAYER .....	37
3.3. MI_DISP_INPUTPORT .....	37
3.4. MI_DISP_Interface_e.....	38
3.5. MI_DISP_OutputTiming_e.....	38
3.6. MI_DISP_SyncInfo_t.....	40
3.7. MI_DISP_PubAttr_t.....	41
3.8. MI_DISP_Csc_t.....	41
3.9. MI_DISP_CscMatrix_e.....	42
3.10. MI_DISP_VgaParam_t.....	43
3.11. MI_DISP_HdmiParam_t.....	43
3.12. MI_DISP_LcdParam_t .....	44
3.13. MI_DISP_ColorTemperature_t .....	45
3.14. MI_DISP_GammaParam_t .....	45
3.15. MI_DISP_VideoLayerSize_t.....	46
3.16. MI_DISP_VideoLayerAttr_t .....	47
3.17. MI_DISP_RotateMode_e .....	47
3.18. MI_DISP_RotateConfig_t.....	48
3.19. MI_DISP_VidWinRect_t .....	48
3.20. MI_DISP_InputPortAttr_t .....	49
3.21. MI_DISP_Position_t .....	50
3.22. MI_DISP_SyncMode_e .....	51
3.23. MI_DISP_InputPortStatus_e .....	51
3.24. MI_DISP_QueryChanelStatus_t.....	52
3.25. MI_DISP_VideoFrame_t .....	52
<b>4. DISP 错误码.....</b>	<b>54</b>

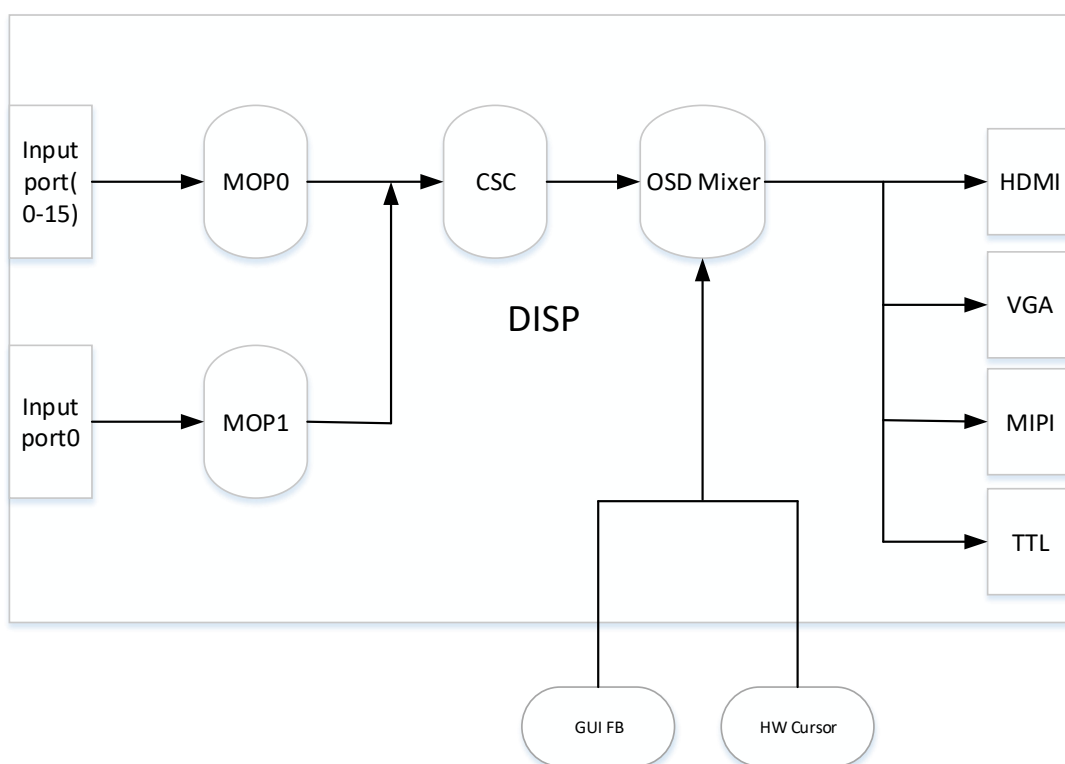
## 1. 概述

### 1.1. 模块说明

DISP 是一个视频显示单元，主要功能是对前端输出的图像做硬件拼图，并对硬件拼图后的图像进行颜色空间转换，最终通过 HDMI/VGA/MIPI/TTL 等接口输出到显示器或 LCD。

### 1.2. 流程框图

#### 1.2.1. 621/623/201/202



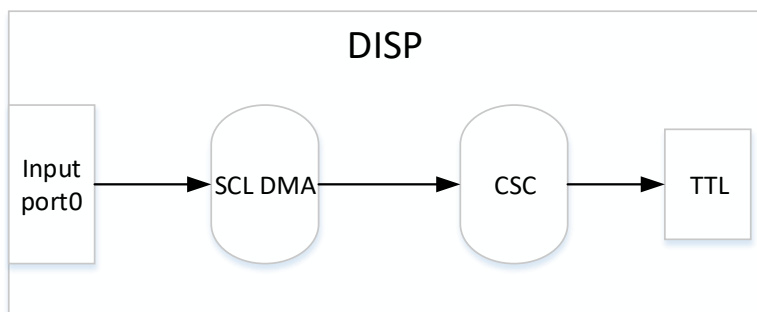
※ 注意

HDMI/VGA 接口可以同时输出，MIPI、TTL 接口只能单独输出。

MOP1 的输出画面会叠加到 MOP0 上，实现 PIP 的功能，即 MOP1 的显示优先级高于 MOP0。

MOP0 上面的 16 个 Input Port 的显示位置不能互相叠加。

### 1.2.2. 328Q/329D/326D/336D/336Q/339G



※ 注意

下表是不同芯片系列输出接口的差异

输出接口 芯片系列	HDMI	VGA	MIPI	TTL
326\3、28\329	不支持	不支持	不支持	支持
325\327	不支持	不支持	不支持	不支持
621\623	支持	支持	不支持	不支持
201\202	不支持	不支持	支持	支持
336\339	不支持	不支持	不支持	支持
335\337	不支持	不支持	不支持	不支持
MSR930	支持	支持	支持	支持

### 1.3. 关键字说明

- MOP  
读取内存图像数据并做拼图处理的硬件单元
- CSC  
颜色空间转换单元
- OSD Mixer  
视频层和 UI 叠加器件
- GUI FB/HW Cursor  
UI 层

## 2. API 参考

API 名	功能
MI_DISP_Enable	启用视频输出设备。
MI_DISP_Disable	禁用视频输出设备。
MI_DISP_SetPubAttr	配置视频输出设备的公共属性
MI_DISP_GetPubAttr	获取视频输出设备的公共属性
MI_DISP_DeviceAttach	使能视频同源输出
MI_DISP_DeviceDetach	禁用视频同源输出
MI_DISP_EnableVideoLayer	使能视频层
MI_DISP_DisableVideoLayer	禁用视频层。
MI_DISP_SetVideoLayerAttr	设定视频层属性
MI_DISP_GetVideoLayerAttr	获取视频层属性
MI_DISP_BindVideoLayer	绑定视频层到指定设备。
MI_DISP_UnBindVideoLayer	解绑定视频层到指定设备。
MI_DISP_SetPlayToleration	设置播放容忍度。
MI_DISP_GetPlayToleration	获取播放容忍度
MI_DISP_GetScreenFrame	获取输出屏幕图像数据。
MI_DISP_ReleaseScreenFrame	释放输出屏幕图像数据。
MI_DISP_EnableInputPort	启用指定的视频输入端口
MI_DISP_DisableInputPort	禁用指定的视频输入端口。
MI_DISP_SetInputPortAttr	配置指定视频输入端口的属性。
MI_DISP_GetInputPortAttr	获取指定视频输入端口的属性
MI_DISP_SetInputPortDispPos	设置指定视频输入端口的显示位置
MI_DISP_GetInputPortDispPos	获取指定视频输入端口的显示位置
MI_DISP_PauseInputPort	暂停指定的视频输入端口
MI_DISP_ResumeInputPort	恢复指定的视频输入端口
MI_DISP_StepInputPort	单帧播放指定的视频输入端口
MI_DISP_RefreshInputPort	刷新指定的视频输入端口。
MI_DISP_ShowInputPort	显示指定的视频输入端口
MI_DISP_HideInputPort	隐藏指定的视频输入端口。
MI_DISP_SetInputPortSyncMode	设定指定的视频输入端口同步模式
MI_DISP_QueryInputPortStat	查询视频输入端口状态

<b>API 名</b>	<b>功能</b>
<a href="#">MI_DISP_GetVgaParam</a>	获取 VGA 输出设备参数
<a href="#">MI_DISP_SetVgaParam</a>	设定 VGA 输出设备参数
<a href="#">MI_DISP_GetHdmiParam</a>	获取 HDMI 输出设备参数。
<a href="#">MI_DISP_SetHdmiParam</a>	设定 HDMI 输出设备参数。
<a href="#">MI_DISP_GetLcdParam</a>	获取 LCD 输出设备参数
<a href="#">MI_DISP_SetLcdParam</a>	设定 LCD 输出设备参数
<a href="#">MI_DISP_DeviceGetColorTempature</a>	获取当前输出图像的色温参数
<a href="#">MI_DISP_DeviceSetColorTempature</a>	设定当前输出图像的色温参数
<a href="#">MI_DISP_DeviceSetGammaParam</a>	设定当前输出图像的 Gamma
<a href="#">MI_DISP_SetVideoLayerRotateMode</a>	设定当前输出图像的旋转角度
<a href="#">MI_DISP_ClearInputPortBuffer</a>	清除当前显示窗口内容

## 2.1. MI\_DISP\_Enable

➤ 功能

启用视频输出设备。

➤ 语法

MI\_S32 MI\_DISP\_Enable (MI\_DISP\_DEV DispDev);

➤ 形参

参数名称	描述	输入/输出
DispDev	视频输出设备号。	输入

➤ 返回值

返回值 {  
     0      成功。  
   非 0    失败，详情参照[错误码](#)。

➤ 依赖

- 头文件：mi\_disp.h
- 库文件：

※ 注意

- 由于系统没有初始化设备为使能状态，所以在使用视频输出功能前必须先进行设备使能操作。
- 在调用设备使能前，必须对设备公共属性进行配置，否则返回设备未配置错误。

➤ 举例

```
MI_U32 DispDev = 0;
MI_U32 DispLayer = 0;
MI_U32 DispInport = 0;
MI_DISP_PubAttr_t stPubAttr;
MI_DISP_VideoLayerAttr_t stLayerAttr;
MI_DISP_InputPortAttr_t stInputPortAttr;

memset(&stPubAttr,0,sizeof(MI_DISP_PubAttr_t));
memset(&stLayerAttr,0,sizeof(MI_DISP_VideoLayerAttr_t));
memset(&stInputPortAttr,0,sizeof(MI_DISP_InputPortAttr_t));

stPubAttr.eIntfSync = E_MI_DISP_OUTPUT_1080P60;
stPubAttr.eIntfType = E_MI_DISP_INTF_HDMI;
MI_DISP_SetPubAttr(DispDev, &stPubAttr);
MI_DISP_Enable(DispDev);

stLayerAttr.stVidLayerSize.u16Width = 1920;
stLayerAttr.stVidLayerSize.u16Height = 1080;
stLayerAttr.stVidLayerDispWin.u16X = 0;
stLayerAttr.stVidLayerDispWin.u16Y = 0;
stLayerAttr.stVidLayerDispWin.u16Width = 1920;
```

```

stLayerAttr.stVidLayerDispWin.u16Height = 1080;
MI_DISP_BindVideoLayer(DispLayer,DispDev);
MI_DISP_SetVideoLayerAttr(DispLayer, &stLayerAttr);
MI_DISP_EnableVideoLayer(DispLayer);

stInputPortAttr.u16SrcWidth = 1920;
stInputPortAttr.u16SrcHeight = 1080;
stInputPortAttr.stDispWin.u16X = 0;
stInputPortAttr.stDispWin.u16Y = 0;
stInputPortAttr.stDispWin.u16Width = 1920;
stInputPortAttr.stDispWin.u16Height = 1080;
MI_DISP_SetInputPortAttr(DispLayer, DispInport, &stInputPortAttr);
MI_DISP_EnableInputPort(DispLayer, DispInport);

//exit flow
MI_DISP_DisableInputPort(DispLayer, DispInport);
MI_DISP_UnBindVideoLayer(DispLayer, DispDev);
MI_DISP_DisableVideoLayer(DispLayer);
MI_DISP_Disable(DispDev);

```

➤ 相关主题

[MI\\_DISP\\_Disable](#)

## 2.2. MI\_DISP\_Disable

➤ 功能

禁用视频输出设备。

➤ 语法

```
MI_S32 MI_DISP_Disable(MI_DISP_DEVDispDev);
```

➤ 形参

参数名称	描述	输入/输出
DispDev	视频输出设备号。	输入

➤ 返回值

返回值 {

- 0 成功。
- 非 0 失败，详情参照[错误码](#)。

➤ 依赖

- 头文件：mi\_disp.h
- 库文件：

※ 注意

无

➤ 举例  
无

➤ 相关主题  
[MI\\_DISP\\_Enable](#)

## 2.3. MI\_DISP\_SetPubAttr

➤ 功能  
配置视频输出设备的公共属性。

➤ 语法  
MI\_S32 MI\_DISP\_SetPubAttr(MI\_DISP\_DEV DispDev, MI\_DISP\_PubAttr\_t \*pstPubAttr);

➤ 形参

参数名称	描述	输入/输出
DispDev	输出设备号。	输入
pstPubAttr	输出设备公共属性结构体指针。	输入

➤ 返回值

返回值 {  
0 成功。  
非 0 失败，详情参照[错误码](#)。

➤ 依赖

- 头文件: mi\_disp.h
- 库文件:

※ 注意  
无

➤ 举例  
无

➤ 相关主题  
[MI\\_DISP\\_GetPubAttr](#)

## 2.4. MI\_DISP\_GetPubAttr

➤ 功能  
获取视频输出设备的公共属性。

➤ 语法  
MI\_S32 MI\_DISP\_GetPubAttr (MI\_DISP\_DEV DispDev, MI\_DISP\_PubAttr\_t \*pstPubAttr);

➤ 形参

参数名称	描述	输入/输出
DispDev	输出设备号。	输入
pstPubAttr	输出设备公共属性结构体指针。	输出

➤ 返回值

返回值	}	0	成功。
		非 0	失败，详情参照 <a href="#">错误码</a> 。

➤ 依赖

- 头文件：mi\_disp.h
- 库文件：

※ 注意

无

➤ 举例

无

➤ 相关主题

[MI\\_DISP\\_SetPubAttr](#)

## 2.5. MI\_DISP\_DeviceAttach

➤ 功能

使能视频同源输出，可以实现不同的输出接口同步显示相同画面。

➤ 语法

```
MI_S32 MI_DISP_DeviceAttach(MI_DISP_DEV DispSrcDev, MI_DISP_DEV DispDstDev);
```

➤ 形参

参数名称	描述	输入/输出
DispSrcDev	输出设备号（视频源输入设备）。	输入
DispDstDev	输出设备号（同源输出设备）。	输入

➤ 返回值

返回值	}	0	成功。
		非 0	失败，详情参照 <a href="#">错误码</a> 。

- 依赖
  - 头文件: `mi_disp.h`
  - 库文件:
- ※ 注意
 

适用于有两个可以同源视频输出设备的芯片, 目前只支持 MSR930、MSR650x。
- 举例
 

无
- 相关主题
 

[MI\\_DISP\\_DeviceDetach](#)

## 2.6. MI\_DISP\_DeviceDetach

- 功能
 

禁用视频同源输出。
- 语法
 

```
MI_S32 MI_DISP_DeviceDetach(MI_DISP_DEV DispSrcDev, MI_DISP_DEV DispDstDev);
```

- 形参

参数名称	描述	输入/输出
DispSrcDev	输出设备号 (视频源输入设备)。	输入
DispDstDev	输出设备号 (同源输出设备)。	输入

- 返回值
 

{	0	成功。
	非 0	失败, 详情参照 <a href="#">错误码</a> 。

- 依赖
  - 头文件: `mi_disp.h`
  - 库文件:
- ※ 注意
 

适用于有两个可以同源视频输出设备的芯片, 目前只支持 MSR930、MSR650x。
- 举例
 

无
- 相关主题
 

[MI\\_DISP\\_DeviceAttach](#)

## 2.7. MI\_DISP\_EnableVideoLayer

➤ 功能

使能视频层。

➤ 语法

MI\_S32 MI\_DISP\_EnableVideoLayer (MI\_DISP\_LAYER DispLayer);

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入

➤ 返回值

返回值 {  
     0     成功。  
   非 0   失败，详情参照[错误码](#)。

➤ 依赖

- 头文件: mi\_disp.h
- 库文件:

※ 注意

无

➤ 举例

无

➤ 相关主题

[MI\\_DISP\\_DisableVideoLayer](#)

## 2.8. MI\_DISP\_DisableVideoLayer

➤ 功能

禁用视频层。

➤ 语法

MI\_S32 MI\_DISP\_DisableVideoLayer (MI\_DISP\_LAYER DispLayer);

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入

- 返回值
 

返回值	{	0	成功。
		非 0	失败，详情参照 <a href="#">错误码</a> 。
- 依赖
  - 头文件：mi\_disp.h
  - 库文件：
- ※ 注意
 

无
- 举例
 

无
- 相关主题
 

[MI\\_DISP\\_EnableVideoLayer](#)

## 2.9. MI\_DISP\_SetVideoLayerAttr

- 功能
 

设定视频层属性。
- 语法
 

```
MI_S32 MI_DISP_SetVideoLayerAttr(MI_DISP_LAYER DispLayer,
MI_DISP_VideoLayerAttr_t *pstLayerAttr);
```
- 形参
 

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
pstLayerAttr	视频层属性结构体指针	输入
- 返回值
 

返回值	{	0	成功。
		非 0	失败，详情参照 <a href="#">错误码</a> 。
- 依赖
  - 头文件：mi\_disp.h
  - 库文件：
- ※ 注意
 

无

➤ 举例  
无

➤ 相关主题  
[MI\\_DISP\\_GetVideoLayerAttr](#)

## 2.10. MI\_DISP\_GetVideoLayerAttr

➤ 功能  
获取视频层属性。

➤ 语法  
MI\_S32 MI\_DISP\_GetVideoLayerAttr(MI\_DISP\_LAYER DispLayer,  
[MI\\_DISP\\_VideoLayerAttr\\_t](#) \*pstLayerAttr);

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
pstLayerAttr	视频层属性结构体指针	输出

➤ 返回值

返回值 {

- 0 成功。
- 非 0 失败，详情参照[错误码](#)。

➤ 依赖

- 头文件 : mi\_disp.h
- 库文件 :

※ 注意  
无

➤ 举例  
无

➤ 相关主题  
[MI\\_DISP\\_SetVideoLayerAttr](#)

## 2.11. MI\_DISP\_BindVideoLayer

➤ 功能  
绑定视频层到指定设备。

➤ 语法  
MI\_S32 MI\_DISP\_BindVideoLayer(MI\_DISP\_LAYER DispLayer, MI\_DISP\_DEV DispDev);

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
DispDev	输出设备号。	输入

➤ 返回值

返回值	}	0	成功。
		非 0	失败，详情参照 <a href="#">错误码</a> 。

➤ 依赖

- 头文件：mi\_disp.h
- 库文件：

※ 注意

无

➤ 举例

无

➤ 相关主题

[MI\\_DISP\\_UnBindVideoLayer](#)

## 2.12. MI\_DISP\_UnBindVideoLayer

➤ 功能

解绑定视频层到指定设备。

➤ 语法

MI\_S32 MI\_DISP\_UnBindVideoLayer(MI\_DISP\_LAYER DispLayer, MI\_DISP\_DEVDispDev);

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
DispDev	输出设备号。	输入

➤ 返回值

返回值	}	0	成功。
		非 0	失败，详情参照 <a href="#">错误码</a> 。

- 依赖
  - 头文件：mi\_disp.h
  - 库文件：
- ※ 注意
 

无
- 举例
 

无
- 相关主题
 

[MI\\_DISP\\_BindVideoLayer](#)

## 2.13. MI\_DISP\_SetPlayToleration

- 功能
 

设置播放容忍度。
- 语法
 

```
MI_S32 MI_DISP_SetPlayToleration(MI_DISP_LAYER DispLayer, MI_U32 u32Toleration);
```
- 形参
 

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
u32Toleration	播放容忍度，用来调整帧率控制算法的误差允许范围	输入
- 返回值
 

返回值 {
 
  - 0 成功。
  - 非 0 失败，详情参照[错误码](#)。
- 依赖
  - 头文件：mi\_disp.h
  - 库文件：
- ※ 注意
  - 播放容忍度单位为毫秒。
  - 调用前请保证视频层已经使能。
  - 目前只支持 MSR930、MSR650x。
- 举例
 

无
- 相关主题
 

[MI\\_DISP\\_GetPlayToleration](#)

## 2.14. MI\_DISP\_GetPlayToleration

➤ 功能

获取播放容忍度。

➤ 语法

```
MI_S32 MI_DISP_GetPlayToleration (MI_DISP_LAYER DispLayer, MI_U32 *pu32Toleration);
```

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
pu32Toleration	播放容忍度。	输出

➤ 返回值

返回值 { 0 成功。  
非 0 失败，详情参照[错误码](#)。

➤ 依赖

- 头文件: mi\_disp.h
- 库文件:

※ 注意

- 播放容忍度单位为毫秒。
- 调用前请保证视频层已经使能。
- 目前只支持 MSR930、MSR650x。

➤ 举例

无

➤ 相关主题

[MI\\_DISP\\_SetPlayToleration](#)

## 2.15. MI\_DISP\_GetScreenFrame

➤ 功能

获取输出频幕图像数据。

➤ 语法

```
MI_S32 MI_DISP_GetScreenFrame(MI_DISP_LAYER DispLayer, MI_DISP_VideoFrame_t *pstVFrame, MI_S32 s32MilliSec);
```

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
pstVFrame	获取的输出频幕图像数据信息结构体指针。	输出
s32MilliSec	超时参数 s32MilliSec 设为-1 时，为阻塞接口；0 时为非阻塞接口；大于 0 时为超时等待时间。超时时间的单位为毫秒（ms）。	输入

➤ 返回值

返回值	}	0	成功。
		非 0	失败，详情参照 <a href="#">错误码</a> 。

➤ 依赖

- 头文件：mi\_disp.h
- 库文件：

※ 注意

- 目前只支持 MSR930、MSR650x。
- 调用前请保证视频层已经使能。

➤ 举例

无

➤ 相关主题

[MI\\_DISP\\_ReleaseScreenFrame](#)

## 2.16. MI\_DISP\_ReleaseScreenFrame

➤ 功能

释放输出屏幕图像数据。

➤ 语法

```
MI_S32 MI_DISP_ReleaseScreenFrame (MI_DISP_LAYER DispLayer,
MI_DISP_VideoFrame_t *pstVFrame);
```

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
pstVFrame	获取的输出屏幕图像数据信息结构体指针。	输入

- 返回值
 

返回值	{	0	成功。
		非 0	失败，详情参照 <a href="#">错误码</a> 。
- 依赖
  - 头文件: `mi_disp.h`
  - 库文件:
- ※ 注意
  - 目前只支持 MSR930、MSR650x。
  - 调用前请保证视频层已经使能。
- 举例
 

无
- 相关主题
 

[MI\\_DISP\\_GetScreenFrame](#)

## 2.17. MI\_DISP\_EnableInputPort

- 功能
 

启用指定的视频输入端口。
- 语法
 

```
MI_S32 MI_DISP_EnableInputPort (MI_DISP_LAYER DispLayer, MI_DISP_INPUTPORT LayerInputPort);
```
- 形参
 

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
LayerInputPort	视频输入端口号。	输入
- 返回值
 

返回值	{	0	成功。
		非 0	失败，详情参照 <a href="#">错误码</a> 。
- 依赖
  - 头文件: `mi_disp.h`
  - 库文件:
- ※ 注意
 

调用前必须使能视频层。

- 举例  
无
- 相关主题  
[MI\\_DISP\\_DisableInputPort](#)

## 2.18. MI\_DISP\_DisableInputPort

- 功能  
禁用指定的视频输入端口。
- 语法  
MI\_S32 MI\_DISP\_DisableInputPort (MI\_DISP\_LAYER DispLayer, MI\_DISP\_INPUTPORT LayerInputPort);

- 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
LayerInputPort	视频输入端口号。	输入

- 返回值
 

{	0	成功。
	非 0	失败，详情参照 <a href="#">错误码</a> 。

- 依赖
  - 头文件: `mi_disp.h`
  - 库文件:

- ※ 注意  
调用前必须使能视频层。

- 举例  
无
- 相关主题  
[MI\\_DISP\\_EnableInputPort](#)

## 2.19. MI\_DISP\_SetInputPortAttr

- 功能  
配置指定视频输入端口的属性。

➤ 语法

```
MI_S32 MI_DISP_SetInputPortAttr(MI_DISP_LAYER DispLayer, MI_DISP_INPUTPORT
LayerInputPort, MI_DISP_InputPortAttr_t* pstInputPortAttr);
```

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
LayerInputPort	视频输入端口号。	输入
pstInputPortAttr	视频输入端口属性指针	输入

➤ 返回值

返回值 {

- 0 成功。
- 非 0 失败，详情参照[错误码](#)。

➤ 依赖

- 头文件: mi\_disp.h
- 库文件:

※ 注意

1. 调用前必须使能视频层。
2. 621、623、201、202 系列芯片，Input Port 在使用时，要当作 channel 和前端模块绑定，比如 Input Port0 应该用作 channel0 和前端绑定。

➤ 举例

无

➤ 相关主题

[MI\\_DISP\\_GetInputPortAttr](#)

## 2.20. MI\_DISP\_GetInputPortAttr

➤ 功能

获取指定视频输入端口的属性。

➤ 语法

```
MI_S32 MI_DISP_GetInputPortAttr
(MI_DISP_LAYER DispLayer, MI_DISP_INPUTPORT LayerInputPort,
MI_DISP_InputPortAttr_t *pstInputPortAttr);
```

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
LayerInputPort	视频输入端口号。	输入
pstInputPortAttr	视频输入端口属性指针	输出

- 返回值
 

{	返回值	0	成功。
		非 0	失败，详情参照 <a href="#">错误码</a> 。
- 依赖
  - 头文件: `mi_disp.h`
  - 库文件:
- ※ 注意
 

调用前必须使能视频层。
- 举例
 

无
- 相关主题
 

[MI\\_DISP\\_SetInputPortAttr](#)

## 2.21. MI\_DISP\_SetInputPortDispPos

- 功能
 

设置指定视频输入端口的显示位置。
- 语法
 

```
MI_S32 MI_DISP_SetInputPortDispPos(MI_DISP_LAYER DispLayer, MI_DISP_INPUTPORT
LayerInputPort, const MI_DISP_Position_t *pstDispPos);
```
- 形参
 

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
LayerInputPort	视频输入端口号。	输入
pstDispPos	输入端口位置指针	输入
- 返回值
 

{	返回值	0	成功。
		非 0	失败，详情参照 <a href="#">错误码</a> 。
- 依赖
  - 头文件: `mi_disp.h`
  - 库文件:
- ※ 注意
 

调用前必须使能视频层。

➤ 举例  
无

➤ 相关主题  
[MI\\_DISP\\_GetInputPortDispPos](#)

## 2.22. MI\_DISP\_GetInputPortDispPos

➤ 功能  
获取指定视频输入端口的显示位置。

➤ 语法  
MI\_S32 MI\_DISP\_GetInputPortDispPos  
(MI\_DISP\_LAYER DispLayer, MI\_DISP\_INPUTPORT LayerInputPort, MI\_DISP\_Position\_t \*pstDispPos);

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
LayerInputPort	视频输入端口号。	输入
pstDispPos	输入端口位置指针	输出

➤ 返回值

返回值 {  
     0     成功。  
   非 0   失败，详情参照[错误码](#)。

➤ 依赖

- 头文件: mi\_disp.h
- 库文件:

※ 注意  
调用前必须使能视频层。

➤ 举例  
无

➤ 相关主题  
[MI\\_DISP\\_SetInputPortDispPos](#)

## 2.23. MI\_DISP\_PauseInputPort

➤ 功能  
暂停指定的视频输入端口。

➤ 语法

MI\_S32 MI\_DISP\_PauseInputPort (MI\_DISP\_LAYER DispLayer, MI\_DISP\_INPUTPORT LayerInputPort);

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
LayerInputPort	视频输入端口号。	输入

➤ 返回值

返回值 { 0 成功。  
非 0 失败，详情参照[错误码](#)。

➤ 依赖

- 头文件: mi\_disp.h
- 库文件:

※ 注意

调用前必须使能视频层。

➤ 举例

无

➤ 相关主题

[MI\\_DISP\\_DisableInputPort](#)  
[MI\\_DISP\\_PauseInputPort](#)  
[MI\\_DISP\\_ResumeInputPort](#)  
[MI\\_DISP\\_StepInputPort](#)  
[MI\\_DISP\\_ShowInputPort](#)  
[MI\\_DISP\\_HideInputPort](#)

## 2.24. MI\_DISP\_ResumeInputPort

➤ 功能

恢复指定的视频输入端口。

➤ 语法

MI\_S32 MI\_DISP\_ResumeInputPort (MI\_DISP\_LAYER DispLayer, MI\_DISP\_INPUTPORT LayerInputPort);

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
LayerInputPort	视频输入端口号。	输入

➤ 返回值

返回值 { 0 成功。  
非 0 失败，详情参照[错误码](#)。

➤ 依赖

- 头文件: mi\_disp.h
- 库文件:

※ 注意

调用前必须使能视频层。

➤ 举例

无

➤ 相关主题

[MI\\_DISP\\_DisableInputPort](#)  
[MI\\_DISP\\_PauseInputPort](#)  
[MI\\_DISP\\_ResumeInputPort](#)  
[MI\\_DISP\\_StepInputPort](#)  
[MI\\_DISP\\_ShowInputPort](#)  
[MI\\_DISP\\_HideInputPort](#)

## 2.25. MI\_DISP\_StepInputPort

➤ 功能

单帧播放指定的视频输入端口。

➤ 语法

MI\_S32 MI\_DISP\_StepInputPort (MI\_DISP\_LAYER DispLayer, MI\_DISP\_INPUTPORT LayerInputPort);

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
LayerInputPort	视频输入端口号。	输入

➤ 返回值

返回值 { 0 成功。  
非 0 失败，详情参照[错误码](#)。

- 依赖
  - 头文件: mi\_disp.h
  - 库文件:
- ※ 注意
  - 目前只支持 MSR930、MSR650x。
  - 调用前必须使能视频层。
- 举例
 

无
- 相关主题
  - MI\_DISP\_DisableInputPort
  - MI\_DISP\_PauseInputPort
  - MI\_DISP\_ResumeInputPort
  - MI\_DISP\_StepInputPort
  - MI\_DISP\_ShowInputPort
  - MI\_DISP\_HideInputPort

## 2.26. MI\_DISP\_ShowInputPort

- 功能
 

显示指定的视频输入端口。
- 语法
 

MI\_S32 MI\_DISP\_ShowInputPort (MI\_DISP\_LAYER DispLayer, MI\_DISP\_INPUTPORT LayerInputPort);
- 形参
 

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
LayerInputPort	视频输入端口号。	输入
- 返回值
 

返回值 {
 
  - 0 成功。
  - 非 0 失败，详情参照[错误码](#)。
- 依赖
  - 头文件: mi\_disp.h
  - 库文件:
- ※ 注意
 

调用前必须使能视频层。

➤ 举例

无

➤ 相关主题

[MI\\_DISP\\_DisableInputPort](#)  
[MI\\_DISP\\_PauseInputPort](#)  
[MI\\_DISP\\_ResumeInputPort](#)  
[MI\\_DISP\\_StepInputPort](#)  
[MI\\_DISP\\_ShowInputPort](#)  
[MI\\_DISP\\_HideInputPort](#)

## 2.27. MI\_DISP\_HideInputPort

➤ 功能

隐藏指定的视频输入端口。

➤ 语法

MI\_S32 MI\_DISP\_HideInputPort (MI\_DISP\_LAYER DispLayer, MI\_DISP\_INPUTPORT LayerInputPort);

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
LayerInputPort	视频输入端口号。	输入

➤ 返回值

返回值 {  
     0      成功。  
   非 0    失败，详情参照[错误码](#)。

➤ 依赖

- 头文件: mi\_disp.h
- 库文件:

※ 注意

调用前必须使能视频层。

➤ 举例

无

➤ 相关主题

[MI\\_DISP\\_DisableInputPort](#)  
[MI\\_DISP\\_PauseInputPort](#)  
[MI\\_DISP\\_ResumeInputPort](#)  
[MI\\_DISP\\_StepInputPort](#)  
[MI\\_DISP\\_ShowInputPort](#)  
[MI\\_DISP\\_HideInputPort](#)

## 2.28. MI\_DISP\_SetInputPortSyncMode

➤ 功能

设定指定的视频输入端口同步模式。

➤ 语法

```
MI_S32 MI_DISP_SetInputPortSyncMode (MI_DISP_LAYER DispLayer,
MI_DISP_INPUTPORT LayerInputPort, MI_DISP_SyncMode_e eMode);
```

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
LayerInputPort	视频输入端口号。	输入
eMode	输入端口同步模式	输入

➤ 返回值

返回值 {  
     0      成功。  
   非 0    失败，详情参照[错误码](#)。

➤ 依赖

- 头文件: mi\_disp.h
- 库文件:

※ 注意

- 目前只支持 MSR930、MSR650x。
- 调用前必须使能视频层。

➤ 举例

无

➤ 相关主题

无

## 2.29. MI\_DISP\_QueryInputPortStat

➤ 功能

查询视频输入端口状态。

➤ 语法

```
MI_S32 MI_DISP_QueryInputPortStat(MI_DISP_LAYER DispLayer, MI_DISP_INPUTPORT
LayerInputPort, MI_DISP_QueryChanelStatus_t *pstStatus);
```

➤ 形参

参数名称	描述	输入/输出
DispLayer	视频输出视频层号。	输入
LayerInputPort	视频输入端口号。	输入
pstStatus	输入端口状态指针。	输出

➤ 返回值

返回值 { 0 成功。  
非 0 失败，详情参照[错误码](#)。

➤ 依赖

- 头文件: `mi_disp.h`
- 库文件:

※ 注意

调用前必须使能视频层。

➤ 举例

无

➤ 相关主题

无

## 2.30. MI\_DISP\_GetVgaParam

➤ 功能

获取VGA输出设备参数。

➤ 语法

```
MI_S32 MI_DISP_GetVgaParam(MI_DISP_DEV DispDev, MI_DISP_VgaParam_t *pstVgaParam);
```

➤ 形参

参数名称	描述	输入/输出
DispDev	视频输出设备号。	输入
pstVgaParam	VGA 图像输出效果结构体指针。	输出

➤ 返回值

返回值 { 0 成功。  
非 0 失败，详情参照[错误码](#)。

➤ 依赖

- 头文件: mi\_disp.h
- 库文件:

※ 注意

- 目前只支持 MSR930、MSR650x、621、623。
- 调用前必须使能视频层。

➤ 举例

无

➤ 相关主题

[MI\\_DISP\\_SetVgaParam](#)

## 2.31. MI\_DISP\_SetVgaParam

➤ 功能

设定VGA输出设备参数。

➤ 语法

MI\_S32 MI\_DISP\_SetVgaParam (MI\_DISP\_DEV DispDev, MI\_DISP\_VgaParam\_t \*pstVgaParam);

➤ 形参

参数名称	描述	输入/输出
DispDev	视频输出设备号。	输入
pstVgaParam	VGA 图像输出效果结构体指针。	输入

➤ 返回值

返回值 { 0 成功。  
非 0 失败，详情参照[错误码](#)。

➤ 依赖

- 头文件: mi\_disp.h
- 库文件:

※ 注意

- 目前只支持 MSR930、MSR650x、621、623。
- 调用前必须使能视频层。

➤ 举例

无

➤ 相关主题

[MI\\_DISP\\_GetVgaParam](#)

## 2.32. MI\_DISP\_GetHdmiParam

➤ 功能

获取HDMI输出设备参数。

➤ 语法

```
MI_S32 MI_DISP_GetHdmiParam(MI_DISP_DEV DispDev, MI_DISP_HdmiParam_t *pstHdmiParam);
```

➤ 形参

参数名称	描述	输入/输出
DispDev	视频输出设备号。	输入
pstVgaParam	VGA 图像输出效果结构体指针。	输出

➤ 返回值

返回值 {  
     0     成功。  
   非 0   失败，详情参照[错误码](#)。

➤ 依赖

- 头文件: mi\_disp.h
- 库文件:

※ 注意

- 目前只支持 MSR930、MSR650x、621、623。
- 调用前必须使能视频层。

➤ 举例

无

➤ 相关主题

[MI\\_DISP\\_SetHdmiParam](#)

## 2.33. MI\_DISP\_SetHdmiParam

➤ 功能

设定HDMI输出设备参数。

➤ 语法

```
MI_S32 MI_DISP_SetHdmiParam(MI_DISP_DEV DispDev, MI_DISP_HdmiParam_t *pstHdmiParam);
```

➤ 形参

参数名称	描述	输入/输出
DispDev	视频输出设备号。	输入
pstHdmiParam	HDMI 图像输出效果结构体指针。	输入

➤ 返回值

返回值 {  
     0     成功。  
   非 0   失败，详情参照[错误码](#)。

➤ 依赖

- 头文件: `mi_disp.h`
- 库文件:

※ 注意

- 目前只支持 MSR930、MSR650x、621、623。
- 调用前必须使能视频层。

➤ 举例

无

➤ 相关主题

[MI\\_DISP\\_GetHdmiParam](#)

## 2.34. MI\_DISP\_GetLcdParam

➤ 功能

获取LCD输出设备参数。

➤ 语法

```
MI_S32 MI_DISP_GetLcdParam(MI_DISP_DEV DispDev, MI_DISP_LcdParam_t *pstLcdParam);
```

➤ 形参

参数名称	描述	输入/输出
DispDev	视频输出设备号。	输入
pstLcdParam	LCD 图像输出效果结构体指针。	输入

➤ 返回值

返回值 {  
     0     成功。  
   非 0   失败，详情参照[错误码](#)。

- 依赖
  - 头文件: mi\_disp.h
  - 库文件:
- ※ 注意
  - 目前只支持 201、202。
  - 调用前必须使能视频层。
- 举例
 

无
- 相关主题
 

[MI\\_DISP\\_SetLcdParam](#)

## 2.35. MI\_DISP\_SetLcdParam

- 功能
 

设定LCD输出设备参数。
- 语法
 

```
MI_S32 MI_DISP_SetLcdParam(MI_DISP_DEV DispDev, MI_DISP_LcdParam_t *pstLcdParam);
```

- 形参

参数名称	描述	输入/输出
DispDev	视频输出设备号。	输入
pstLcdParam	LCD 图像输出效果结构体指针。	输入

- 返回值
 

返回值 {

  - 0 成功。
  - 非 0 失败，详情参照[错误码](#)。

- 依赖
  - 头文件: mi\_disp.h
  - 库文件:
- ※ 注意
  - 目前只支持 201、202。
  - 调用前必须使能视频层。
- 举例
 

无
- 相关主题
 

[MI\\_DISP\\_GetLcdParam](#)

## 2.36. MI\_DISP\_DeviceGetColorTemperture

➤ 功能

设定图像输出色温参数。

➤ 语法

```
MI_S32 MI_DISP_DeviceGetColorTemperture(MI_DISP_DEV DispDev,
MI_DISP_ColorTemperature_t *pstColorTempInfo);
```

➤ 形参

参数名称	描述	输入/输出
DispDev	视频输出设备号。	输入
pstColorTempInfo	图像输出色温参数结构体指针。	输入

➤ 返回值

返回值 { 0 成功。  
非 0 失败，详情参照[错误码](#)。

➤ 依赖

- 头文件: mi\_disp.h
- 库文件:

※ 注意

- 目前只支持 201、202。
- 调用前必须使能视频层。

➤ 举例

无

➤ 相关主题

[MI\\_DISP\\_DeviceSetColorTemperture](#)

## 2.37. MI\_DISP\_DeviceSetColorTemperture

➤ 功能

设定LCD输出设备参数。

➤ 语法

```
MI_S32 MI_DISP_DeviceSetColorTemperture(MI_DISP_DEV DispDev,
MI_DISP_ColorTemperature_t *pstColorTempInfo);
```

➤ 形参

参数名称	描述	输入/输出
DispDev	视频输出设备号。	输入
pstColorTempInfo	图像输出色温参数结构体指针。	输入

➤ 返回值

返回值	}	0	成功。
		非 0	失败，详情参照 <a href="#">错误码</a> 。

➤ 依赖

- 头文件: mi\_disp.h
- 库文件:

※ 注意

- 目前只支持 201、202。
- 调用前必须使能视频层。

➤ 举例

无

➤ 相关主题

[MI\\_DISP\\_DeviceGetColorTempature](#)

## 2.38. MI\_DISP\_DeviceSetGammaParam

➤ 功能

设定图像输出Gamma参数。

➤ 语法

```
MI_S32 MI_DISP_DeviceSetGammaParam(MI_DISP_DEV DispDev,
MI_DISP_GammaParam_t * pstGammaParam);
```

➤ 形参

参数名称	描述	输入/输出
DispDev	视频输出设备号。	输入
pstGammaParam	图像输出 Gamma 参数结构体指针。	输入

➤ 返回值

返回值	}	0	成功。
		非 0	失败，详情参照 <a href="#">错误码</a> 。

➤ 依赖

- 头文件: mi\_disp.h
- 库文件:

※ 注意

- 目前只支持 201、202。
- 调用前必须使能视频层。

➤ 举例

无

➤ 相关主题

无

## 2.39. MI\_DISP\_SetVideoLayerRotateMode

➤ 功能

设定图像输出色温参数。

➤ 语法

```
MI_S32 MI_DISP_SetVideoLayerRotateMode(MI_DISP_LAYER DispLayer,
MI_DISP_RotateConfig_t *pstRotateConfig);
```

➤ 形参

参数名称	描述	输入/输出
DispDev	视频输出设备号。	输入
pstRotateConfig	图像输出旋转参数结构体指针。	输入

➤ 返回值

返回值 { 0 成功。  
非 0 失败，详情参照[错误码](#)。

➤ 依赖

- 头文件: mi\_disp.h
- 库文件:

※ 注意

- 目前只支持 201、202。
- 调用前必须使能视频层。

➤ 举例

无

- 相关主题  
无

## 2.40. MI\_DISP\_ClearInputPortBuffer

- 功能  
清除当前显示窗口视频数据。

- 语法  

```
MI_S32 MI_DISP_ClearInputPortBuffer(MI_DISP_LAYER DispLayer, MI_DISP_INPUTPORT LayerInputPort);
```

- 形参

参数名称	描述	输入/输出
DispLayer	视频层号。	输入
LayerInputPort	视频输入端口号。	输入

- 返回值  
返回值 {
  - 0 成功。
  - 非 0 失败，详情参照[错误码](#)。

- 依赖
  - 头文件: mi\_disp.h
  - 库文件:

- ※ 注意  
调用前必须使能视频层。

- 举例  
无

- 相关主题  
无

### 3. DISP 数据类型

DISP 模块相关数据类型定义如下：

MI_DISP_DEV	定义 DISP_DEV 类型
MI_DISP_LAYER	定义 DISP_LAYER 类型
MI_DISP_INPUTPORT	定义 LAYER_INPUTPORT 类型。
MI_DISP_Interface_e	定义 DISP 视频输出接口
MI_DISP_OutputTiming_e	定义 DISP 视频输出 timing
MI_DISP_SyncInfo_t:	定义 DISP 视频输出时序相关参数。
MI_DISP_PubAttr_t	定义 DISP 公共属性。
MI_DISP_Csc_t	定义 DISP 颜色转换信息
MI_DISP_CscMatrix_e	定义 DISP 颜色转换矩阵
MI_DISP_VgaParam_t:	定义 DISP VGA 输出参数信息
MI_DISP_HdmiParam_t	定义 DISP HDMI 输出参数信息。
MI_DISP_LcdParam_t	定义 DISP LCD 输出参数信息
MI_DISP_ColorTemperature_t	定义 DISP 视频输出色温参数
MI_DISP_GammaParam_t	定义 DISP 视频输出 Gamma 参数
MI_DISP_VideoLayerSize_t	定义 DISP 视频层画面大小。
MI_DISP_VideoLayerAttr_t	定义 DISP 视频层属性
MI_DISP_RotateMode_e	定义 DISP 窗口属性
MI_DISP_RotateConfig_t	定义 DISP 视频旋转参数
MI_DISP_VidWinRect_t	定义 DISP 视频输出窗口信息
MI_DISP_InputPortAttr_t	定义 DISP 输入端口属性
MI_DISP_Position_t:	定义 DISP 输入端口位置
MI_DISP_SyncMode_e:	定义 DISP 同步模式
MI_DISP_InputPortStatus_e	定义 DISP 输入端口状态类型
MI_DISP_QueryChanelStatus_t:	定义 DISP 输入端口状态信息。
MI_DISP_VideoFrame_t	定义 DISP 视频帧信息

### 3.1. MI\_DISP\_DEV

- 说明  
定义 DISP\_DEV 类型。
- 定义  

```
typedef MI_S32 MI_DISP_DEV;
```
- 成员  
无。
- ※ 注意事项  
无。

### 3.2. MI\_DISP\_LAYER

- 说明  
定义 DISP\_LAYER 类型。  
DISP 以 video layer 作为输出单位,对于有多个 Input Port 的 DISP 来说,所有 Input Port 最终拼接为一个 video layer 输出到显示器;对于只有一个 Input Port 的 DISP, Input Port size 等于 video layer size。
- 定义  

```
typedef MI_S32 MI_DISP_LAYER;
```
- 成员  
无。
- 相关数据类型及接口  
无。

### 3.3. MI\_DISP\_INPUTPORT

- 说明  
定义 LAYER\_INPUTPORT 类型。
- 定义  

```
typedef MI_S32 MI_DISP_INPUTPORT;
```
- ※ 注意事项  
无。
- 相关数据类型及接口  
无。

## 3.4. MI\_DISP\_Interface\_e

➤ 说明

定义视频输出接口。

➤ 定义

```
typedef enum
{
    E_MI_DISP_INTF_CVBS = 0,
    E_MI_DISP_INTF_YPBPR,
    E_MI_DISP_INTF_VGA,
    E_MI_DISP_INTF_BT656,
    E_MI_DISP_INTF_BT1120,
    E_MI_DISP_INTF_HDMI,
    E_MI_DISP_INTF_LCD,
    E_MI_DISP_INTF_BT656_H,
    E_MI_DISP_INTF_BT656_L,
    E_MI_DISP_INTF_MAX,
}MI_DISP_Interface_e;
```

※ 注意事项

无。

➤ 相关数据类型及接口

MI\_DISP\_PubAttr\_t  
MI\_DISP\_SetPubAttr。

## 3.5. MI\_DISP\_OutputTiming\_e

➤ 说明

定义 DISP timing 类型。

➤ 定义

```
Typedef enum
{
    E_MI_DISP_OUTPUT_PAL = 0,
    E_MI_DISP_OUTPUT_NTSC,
    E_MI_DISP_OUTPUT_960H_PAL,
    /* ITU-R BT.1302 960 x 576 at 50 Hz (interlaced)*/
    E_MI_DISP_OUTPUT_960H_NTSC,
    /* ITU-R BT.1302 960 x 480 at 60 Hz (interlaced)*/
    E_MI_DISP_OUTPUT_1080P24,
    E_MI_DISP_OUTPUT_1080P25,
    E_MI_DISP_OUTPUT_1080P30,

    E_MI_DISP_OUTPUT_720P50,
    E_MI_DISP_OUTPUT_720P60,
```

```

E_MI_DISP_OUTPUT_1080I50,
E_MI_DISP_OUTPUT_1080I60,
E_MI_DISP_OUTPUT_1080P50,
E_MI_DISP_OUTPUT_1080P60,

E_MI_DISP_OUTPUT_576P50,
E_MI_DISP_OUTPUT_480P60,

E_MI_DISP_OUTPUT_640x480_60,
/* VESA 640 x 480 at 60 Hz (non-interlaced) CVT */
E_MI_DISP_OUTPUT_800x600_60,
/* VESA 800 x 600 at 60 Hz (non-interlaced) */
E_MI_DISP_OUTPUT_1024x768_60,
/* VESA 1024 x 768 at 60 Hz (non-interlaced) */
E_MI_DISP_OUTPUT_1280x1024_60,
/* VESA 1280 x 1024 at 60 Hz (non-interlaced) */
E_MI_DISP_OUTPUT_1366x768_60,
/* VESA 1366 x 768 at 60 Hz (non-interlaced) */
E_MI_DISP_OUTPUT_1440x900_60,
/* VESA 1440 x 900 at 60 Hz (non-interlaced) CVT Compliant */
E_MI_DISP_OUTPUT_1280x800_60,
/* 1280*800@60Hz VGA@60Hz*/
E_MI_DISP_OUTPUT_1680x1050_60,
/* VESA 1680 x 1050 at 60 Hz (non-interlaced) */
E_MI_DISP_OUTPUT_1920x2160_30,
/* 1920x2160_30 */
E_MI_DISP_OUTPUT_1600x1200_60,
/* VESA 1600 x 1200 at 60 Hz (non-interlaced) */
E_MI_DISP_OUTPUT_1920x1200_60,
/* VESA 1920 x 1600 at 60 Hz (non-interlaced) CVT (Reduced Blanking)*/
E_MI_DISP_OUTPUT_2560x1440_30, /* 2560x1440_30 */
E_MI_DISP_OUTPUT_2560x1600_60, /* 2560x1600_60 */
E_MI_DISP_OUTPUT_3840x2160_30, /* 3840x2160_30 */
E_MI_DISP_OUTPUT_3840x2160_60, /* 3840x2160_60 */
E_MI_DISP_OUTPUT_USER,
E_MI_DISP_OUTPUT_BUTT
} MI_DISP_IntfSync_e;

```

➤ 成员

各个 Timing 的定义，LCD 显示应该使用 E\_MI\_DISP\_OUTPUT\_USER。

※ 注意事项  
无

➤ 相关数据类型及接口

```

MI_DISP_PubAttr_t
MI_DISP_SetPubAttr。

```

### 3.6. MI\_DISP\_SyncInfo\_t

➤ 说明

定义 DISP 输出屏参信息。

➤ 定义

```
typedef struct MI_DISP_SyncInfo_s
{
    MI_BOOL  bSynm;    /* sync mode(0:timing,as BT.656; 1:signal,as LCD) */
    MI_BOOL  bIop;    /* interlaced or progressive display(0:i; 1:p) */
    MI_U8    u8Intfb; /* interlace bit width while output */

    MI_U16   u16Vact ; /* vertical active area */
    MI_U16   u16Vbb;  /* vertical back blank porch */
    MI_U16   u16Vfb;  /* vertical front blank porch */

    MI_U16   u16Hact; /* horizontal active area */
    MI_U16   u16Hbb;  /* horizontal back blank porch */
    MI_U16   u16Hfb;  /* horizontal front blank porch */
    MI_U16   u16Hmid; /* bottom horizontal active area */

    MI_U16   u16Bvact; /* bottom vertical active area */
    MI_U16   u16Bvbb;  /* bottom vertical back blank porch */
    MI_U16   u16Bvfb;  /* bottom vertical front blank porch */

    MI_U16   u16Hpw;  /* horizontal pulse width */
    MI_U16   u16Vpw;  /* vertical pulse width */

    MI_BOOL  bIdv;    /* inverse data valid of output */
    MI_BOOL  bIhs;    /* inverse horizontal synch signal */
    MI_BOOL  bIvs;    /* inverse vertical synch signal */
    MI_U32   u32FrameRate;
} MI_DISP_SyncInfo_t
```

➤ 成员

成员名称	描述
u16Vact	场有效行数
u16Vbb	场同步信号后肩
u16Vfb	场同步信号前肩
u16Hact	行有效像素点数
u16Hbb	行同步信号后肩
u16Hfb	行同步信号前肩
u16Hpw	行同步信号脉宽
u16Vpw	场同步信号脉宽
u32Framerate	帧率

表中未列参数暂未使用到

※ 注意事项

LCD 显示时才需要配置屏参信息。

- 相关数据类型及接口

MI\_DISP\_PubAttr\_t  
MI\_DISP\_SetPubAttr。

### 3.7. MI\_DISP\_PubAttr\_t

- 说明

定义 DISP 公共属性。

- 定义

```
typedef struct MI_DISP_PubAttr_s
{
    MI_U32      u32BgColor;          /* Background color of a device, in RGB format. */
    MI_U32      u32IntfType;        /* Type of a VO interface */
    MI_DISP_IntfSync_e  eIntfSync;  /* Type of a VO interface timing */
    MI_DISP_SyncInfo_t  stSyncInfo; /* Information about VO interface
timings */
} MI_DISP_PubAttr_t;
```

- 成员

成员名称	描述
u32BgColor	设定背景颜色。
u32IntfType	接口定义： MI_DISP_INTF_BT1120 MI_DISP_INTF_BT656 MI_DISP_INTF_BT656_H MI_DISP_INTF_BT656_L MI_DISP_INTF_CVBS MI_DISP_INTF_HDMI MI_DISP_INTF_LCD MI_DISP_INTF_VGA MI_DISP_INTF_YPBPR
eIntfSync	接口时序定义： MI_DISP_IntfSync_e
stSyncInfo	接口自定义时序定义： MI_DISP_SyncInfo_t

- ※ 注意事项

stSyncInfo 是针对用户选择 E\_MI\_DISP\_OUTPUT\_USER，即 Timing 自定义的 Case。

- 相关数据类型及接口

MI\_DISP\_PubAttr\_t  
MI\_DISP\_SetPubAttr。

### 3.8. MI\_DISP\_Csc\_t

- 说明

定义 DISP 颜色转换信息。

➤ 定义

```
typedef struct MI_DISP_Csc_s
{
    MI_DISP_CscMatrix_e eCscMatrix;
    MI_U32 u32Luma;           /* luminance: 0 ~ 100 default: 50 */
    MI_U32 u32Contrast;      /* contrast : 0 ~ 100 default: 50 */
    MI_U32 u32Hue;          /* hue      : 0 ~ 100 default: 50 */
    MI_U32 u32Saturation;    /* saturation: 0 ~ 100 default: 50 */
} MI_DISP_Csc_t;
```

➤ 成员

成员名称	描述
eCscMatrix	颜色转换矩阵
u32Luma	亮度调节
u32Contrast	对比度调节
u32Hue	色度调节
u32Saturation	饱和度调节

※ 注意事项

- 目前只支持 MSR930、MSR650x、621、623、201、202。

➤ 相关数据类型及接口

```
MI_DISP_GetVgaParam
MI_DISP_SetVgaParam
MI_DISP_GetHdmiParam
MI_DISP_SetHdmiParam
MI_DISP_GetLcdParam
MI_DISP_SetLcdParam
```

### 3.9. MI\_DISP\_CscMatrix\_e

➤ 说明

定义 DISP 颜色转换矩阵。

➤ 定义

```
typedef enum
{
    E_MI_DISP_CSC_MATRIX_BYPASS = 0,
    /* do not change color space */
    E_MI_DISP_CSC_MATRIX_BT601_TO_RGB_PC,
    /* change color space from BT.601 to RGB */
    E_MI_DISP_CSC_MATRIX_BT709_TO_RGB_PC,
    /* change color space from BT.709 to RGB */
    E_MI_DISP_CSC_MATRIX_RGB_TO_BT601_PC,
    /* change color space from RGB to BT.601 */
    E_MI_DISP_CSC_MATRIX_RGB_TO_BT709_PC,
    /* change color space from RGB to BT.709 */
    E_MI_DISP_CSC_MATRIX_NUM
} MI_DISP_CscMatrix_e;
```

➤ 成员

成员名称	描述
E_MI_DISP_CSC_MATRIX_BYPASS	不做颜色转换
E_MI_DISP_CSC_MATRIX_BT601_TO_RGB_PC	BT.601 到 RGB 色彩空间的 CSC 矩阵。
E_MI_DISP_CSC_MATRIX_BT709_TO_RGB_PC	BT.709 到 RGB 色彩空间的 CSC 矩阵。
E_MI_DISP_CSC_MATRIX_RGB_TO_BT601_PC	RGB 到 BT.601 色彩空间的 CSC 矩阵。
E_MI_DISP_CSC_MATRIX_RGB_TO_BT709_PC	RGB 到 BT.709 色彩空间的 CSC 矩阵。

※ 注意事项

- 目前只支持 MSR930、MSR650x、621、623、201、202。

➤ 相关数据类型及接口

MI\_DISP\_GetVgaParam  
MI\_DISP\_SetVgaParam  
MI\_DISP\_GetHdmiParam  
MI\_DISP\_SetHdmiParam  
MI\_DISP\_GetLcdParam  
MI\_DISP\_SetLcdParam

### 3.10. MI\_DISP\_VgaParam\_t

➤ 说明

定义 DISP VGA 设定参数信息。

➤ 定义

```
typedef struct MI_DISP_VgaParam_s
{
    MI_DISP_Csc_t stCsc;           /* color space */
    MI_U32 u32Gain;               /* current gain of VGA signals. [0, 64). default:0x30 */
    MI_U32 u32Sharpness;
} MI_DISP_VgaParam_t;
```

➤ 成员

成员名称	描述
stCsc	色彩转换信息定义
u32Gain	型号强度增益
u32Sharpness	锐利度定义

※ 注意事项

- 目前只支持 MSR930、MSR650x、621、623。

➤ 相关数据类型及接口

MI\_DISP\_Csc\_t  
MI\_DISP\_GetVgaParam  
MI\_DISP\_SetVgaParam

### 3.11. MI\_DISP\_HdmiParam\_t

➤ 说明

定义 DISP HDMI 设定参数信息。

➤ 定义

```
typedef struct MI_DISP_HdmiParam_s
{
    MI_DISP_Csc_t stCsc;                /* color space */
    MI_U32 u32Gain;                    /* current gain of HDMI signals. [0, 64). default:0x30 */
    MI_U32 u32Sharpness;
} MI_DISP_HdmiParam_t;
```

➤ 成员

成员名称	描述
stCsc	色彩转换信息定义
u32Gain	型号强度增益
u32Sharpness	锐利度定义

※ 注意事项

- 目前只支持 MSR930、MSR650x、621、623。

➤ 相关数据类型及接口

```
MI_DISP_Csc_t
MI_DISP_GetHdmiParam
MI_DISP_SetHdmiParam
```

### 3.12. MI\_DISP\_LcdParam\_t

➤ 说明

定义 DISP LCD 输出图像参数。

➤ 定义

```
typedef struct MI_DISP_LcdParam_s
{
    MI_DISP_Csc_t stCsc;                /* color space */
    MI_U32 u32Sharpness;
} MI_DISP_LcdParam_t;
```

➤ 成员

成员名称	描述
stCsc	色彩转换信息定义
u32Sharpness	锐利度定义

※ 注意事项

- 目前只支持 201、202。

➤ 相关数据类型及接口

MI\_DISP\_Csc\_t  
MI\_DISP\_GetLcdParam  
MI\_DISP\_SetLcdParam

### 3.13. MI\_DISP\_ColorTemperature\_t

➤ 说明

定义视频输出图像色温参数

➤ 定义

```
typedef struct
{
    MI_U16 u16RedOffset;
    MI_U16 u16GreenOffset;
    MI_U16 u16BlueOffset;

    MI_U16 u16RedColor; // 00~FF, 0x80 is no change
    MI_U16 u16GreenColor;// 00~FF, 0x80 is no change
    MI_U16 u16BlueColor; // 00~FF, 0x80 is no change
}MI_DISP_ColorTemperature_t;
```

➤ 成员

成员名称	描述
u16RedColor	色温矫正参数 R 分量
u16GreenColor	色温矫正参数 G 分量
u16BlueColor	色温矫正参数 B 分量

※ 注意事项

- 目前只支持 201、202。

➤ 相关数据类型及接口

MI\_DISP\_DeviceGetColorTemperture  
MI\_DISP\_DeviceSetColorTemperture

### 3.14. MI\_DISP\_GammaParam\_t

➤ 说明

定义视频输出图像 Gamma 参数

➤ 定义

```
typedef struct
{
    MI_BOOL bEn;
    MI_U16 u16EntryNum;
    MI_U8 *pu8ColorR;
    MI_U8 *pu8ColorG;
    MI_U8 *pu8ColorB;
}MI_DISP_GammaParam_t;
```

➤ 成员

成员名称	描述
bEn	使能 Gamma 调整
u16EntryNum	Gamma table 成员个数
pu8ColorR	R table 指针
pu8ColorG	G table 指针
pu8ColorB	B table 指针

※ 注意事项

- 目前只支持 201、202。

➤ 相关数据类型及接口

MI\_DISP\_DeviceSetGammaParam

### 3.15. MI\_DISP\_VideoLayerSize\_t

➤ 说明

定义 DISP layer 层画面大小。

➤ 定义

```
typedef struct MI_DISP_VideoLayerSize_s
{
    MI_U32 u32Width;
    MI_U32 u32Height;
} MI_DISP_VideoLayerSize_t;
```

➤ 成员

成员名称	描述
u32Width	水平方向大小
u32Height	垂直方向大小

※ 注意事项

无

➤ 相关数据类型及接口

MI\_DISP\_SetVideoLayerAttr  
MI\_DISP\_GetVideoLayerAttr

### 3.16. MI\_DISP\_VideoLayerAttr\_t

➤ 说明

定义 DISP 视频层属性。

➤ 定义

```
typedef struct MI_DISP_VideoLayerAttr_s
{
    MI_DISP_VidWinRect_t    stDispWin;           /* Display resolution */
    MI_DISP_VideoLayerSize_t stLayerSize;       /* Canvas size of the video layer */
    MI_DISP_PixelFormat_e  ePixelFormat;       /* Pixel format of the video layer */
} MI_DISP_VideoLayerAttr_t;
```

➤ 成员

成员名称	描述
stDispWin	输入端口显示位置
stLayerSize	视频层大小
enPixelFormat	输入像素数据格式

※ 注意事项

无

➤ 相关数据类型及接口

MI\_DISP\_SetVideoLayerAttr  
MI\_DISP\_GetVideoLayerAttr

### 3.17. MI\_DISP\_RotateMode\_e

➤ 说明

定义 DISP 视频输出旋转角度。

➤ 定义

```
typedef enum
{
    E_MI_DISP_ROTATE_NONE,
    E_MI_DISP_ROTATE_90,
    E_MI_DISP_ROTATE_270,
    E_MI_DISP_ROTATE_NUM,
}MI_DISP_RotateMode_e;
```

➤ 成员

成员名称	描述
E_MI_DISP_ROTATE_NONE	不做旋转
E_MI_DISP_ROTATE_90	旋转 90 度
E_MI_DISP_ROTATE_270	旋转 270 度

※ 注意事项

- 目前只支持 201、202。

- 相关数据类型及接口

MI\_DISP\_SetVideoLayerRotateMode

### 3.18. MI\_DISP\_RotateConfig\_t

- 说明

定义 DISP 视频输出旋转参数。

- 定义

```
typedef struct MI_DISP_RotateConfig_s
{
    MI_DISP_RotateMode_e eRotateMode;
}MI_DISP_RotateConfig_t;
```

- 成员

成员名称	描述
MI_DISP_RotateConfig_t;	配置输出视频旋转

- ※ 注意事项

- 目前只支持 201、202。

- 相关数据类型及接口

MI\_DISP\_SetVideoLayerRotateMode

### 3.19. MI\_DISP\_VidWinRect\_t

- 说明

定义 DISP 窗口属性。

- 定义

```
typedef struct MI_DISP_VidWinRect_s
{
    MI_U16 u16X;
    MI_U16 u16Y;
    MI_U16 u16Width;
    MI_U16 u16Height;
} MI_DISP_VidWinRect_t;
```

- 成员

成员名称	描述
u16X	水平方向起点
u16Y	垂直方向起点
u16Width	水平方向大小
u16Height	垂直方向大小

- ※ 注意事项

无

- 相关数据类型及接口

MI\_DISP\_SetZoomInWindow

### 3.20. MI\_DISP\_InputPortAttr\_t

- 说明

定义 DISP 输入端口属性。

- 定义

```
typedef struct MI_DISP_InputPortAttr_s
{
    MI_DISP_VidWinRect_t stDispWin;           /* rect of video out chn */
    MI_U16 u16SrcWidth;
    MI_U16 u16SrcHeight;
} MI_DISP_InputPortAttr_t;
```

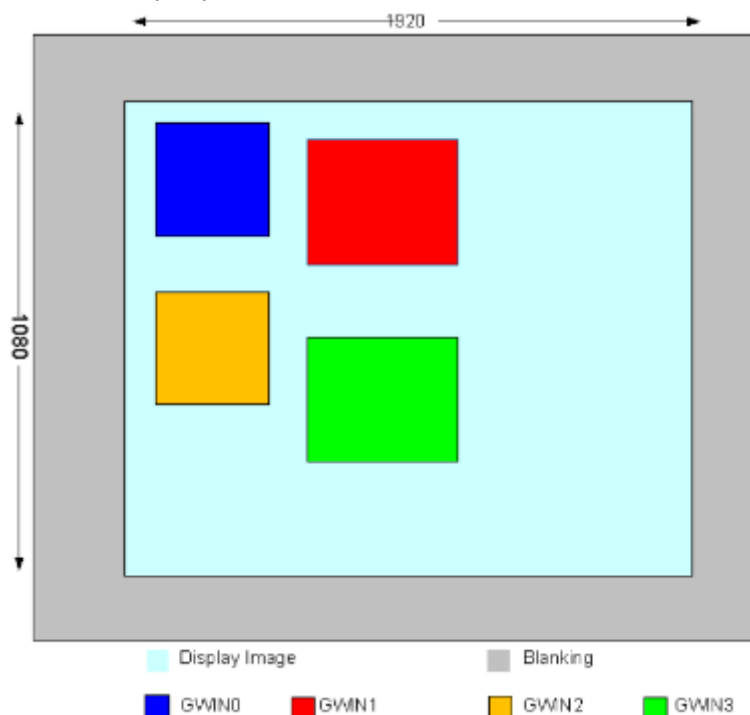
- 成员

成员名称	描述
stDispWin	指定输入端口输出窗口位置及尺寸
u16SrcWidth	指定端口输入图像宽
u16SrcHeight	指定端口输入图像高

- ※ 注意事项

201\202\621\623 系列芯片可以对每个 port 做 scaling up, 不支持独立对 port 做 scaling up 的芯片使用时注意 u16SrcWidth 要等于 stDispWin 中定义的 u16Width, u16SrcHeight 要等于 stDispWin 中定义的 u16Height。

对于有多个 input port 的 video layer 来说, 每个 port 在 video layer 上的显示位置可以任意设置, 但每个 input port 的显示位置不能相互叠加。



- 相关数据类型及接口  
MI\_DISP\_SetInputPortAttr  
MI\_DISP\_GetInputPortAttr

### 3.21. MI\_DISP\_Position\_t

- 说明  
定义 DISP 输入端口位置。

- 定义  

```
typedef struct MI_DISP_Position_s  
{  
    MI_U16 u16X;  
    MI_U16 u16Y;  
} MI_DISP_Position_t;
```

- 成员

成员名称	描述
u16X	水平方向起始点
u16Y	竖直方向起始点

- ※ 注意事项  
无

➤ 相关数据类型及接口

MI\_DISP\_GetInputPortDispPos  
MI\_DISP\_SetInputPortDispPos

### 3.22. MI\_DISP\_SyncMode\_e

➤ 说明

定义 DISP 同步模式。

➤ 定义

```
typedef enum
{
    E_MI_DISP_SYNC_MODE_INVALID = 0,
    E_MI_DISP_SYNC_MODE_CHECK_PTS,
    E_MI_DISP_SYNC_MODE_FREE_RUN,
    E_MI_DISP_SYNC_MODE_NUM,
} MI_DISP_SyncMode_e;
```

➤ 成员

成员名称	描述
E_MI_DISP_SYNC_MODE_INVALID	无效的同步模式
E_MI_DISP_SYNC_MODE_CHECK_PTS	检查 PTS 的同步模式
E_MI_DISP_SYNC_MODE_FREE_RUN	不检查 PTS 的同步模式

※ 注意事项

- 目前只支持 MSR930、MSR650x

➤ 相关数据类型及接口

MI\_DISP\_SetInputPortSyncMode

### 3.23. MI\_DISP\_InputPortStatus\_e

➤ 说明

定义 DISP 输入端口状态类型。

➤ 定义

```
typedef enum
{
    E_MI_DISP_INPUTPORT_STATUS_INVALID = 0,
    E_MI_DISP_INPUTPORT_STATUS_PAUSE,
    E_MI_DISP_INPUTPORT_STATUS_RESUME,
    E_MI_DISP_INPUTPORT_STATUS_STEP,
    E_MI_DISP_INPUTPORT_STATUS_REFRESH,
    E_MI_DISP_INPUTPORT_STATUS_SHOW,
    E_MI_DISP_INPUTPORT_STATUS_HIDE,
    E_MI_DISP_INPUTPORT_STATUS_NUM,
} MI_DISP_InputPortStatus_e;
```

➤ 成员

成员名称	描述
E_MI_DISP_INPUTPORT_STATUS_INVALID	无效的状态模式
E_MI_DISP_INPUTPORT_STATUS_PAUSE	暂停状态
E_MI_DISP_INPUTPORT_STATUS_RESUME	回复状态
E_MI_DISP_INPUTPORT_STATUS_STEP	步进状态
E_MI_DISP_INPUTPORT_STATUS_REFRESH	刷新状态
E_MI_DISP_INPUTPORT_STATUS_SHOW	显示状态
E_MI_DISP_INPUTPORT_STATUS_HIDE	隐藏状态

※ 注意事项

无

➤ 相关数据类型及接口

无

### 3.24. MI\_DISP\_QueryChanelStatus\_t

➤ 说明

定义 DISP 输入端口状态信息。

➤ 定义

```
typedef struct MI_DISP_QueryChanelStatus_s
{
    MI_BOOL bEnable;
    MI_DISP_InputPortStatus_e eStatus;
} MI_DISP_QueryChanelStatus_t;
```

➤ 成员

成员名称	描述
bEnable	输入端口使能状态
eStatus	输入端口状态类型

※ 注意事项

无

➤ 相关数据类型及接口

MI\_DISP\_InputPortStatus\_e  
MI\_DISP\_QueryInputPortStat

### 3.25. MI\_DISP\_VideoFrame\_t

➤ 说明

定义 DISP 视频帧信息。

➤ 定义

```
typedef struct MI_DISP_VideoFrame_s
{
    MI_U32          u32Width;
    MI_U32          u32Height;
    MI_DISP_PixelFormat_e ePixelFormat;
    MI_PHY          aphyAddr[3];
    Void *          *pavirAddr[3];
    MI_U64          u64pts;
    MI_U32          u32PrivateData;
} MI_DISP_VideoFrame_t;
```

➤ 成员

成员名称	描述
u32Width	图像宽度
u32Height	图像高度
ePixelFormat	像素点格式
u32PhyAddr	图像数据物理地址
pVirAddr	图像数据虚拟地址
u64pts	图像数据时间戳
u32PrivateData	私有数据

※ 注意事项

- 目前只支持 MSR930、MSR650x

➤ 相关数据类型及接口

MI\_DISP\_GetScreenFrame  
MI\_DISP\_ReleaseScreenFrame

## 4. DISP 错误码

DISP API 错误码如 3-1 所示。

表 3-1 VPE API 错误码

错误代码	宏定义	描述
0xA00F8001	MI_ERR_DISP_INVALID_DEVID	设备 ID 超出合法范围
0xA00F8002	MI_ERR_DISP_INVALID_CHNID	输入端口 ID 超出合法范围
0xA00F8003	MI_ERR_DISP_ILLEGAL_PARAM	参数超出合法范围
0xA00F8006	MI_ERR_DISP_NULL_PTR	函数参数中有空指针
0xA00F8008	MI_ERR_DISP_NOT_SUPPORT	不支持的操作
0xA00F8009	MI_ERR_DISP_NOT_PERMIT	操作不允许
0xA00F800C	MI_ERR_DISP_NO_MEM	内存不足
0xA00F8010	MI_ERR_DISP_SYS_NOTREADY	系统未初始化
0xA00F8012	MI_ERR_DISP_BUSY	资源忙
0xA00F8040	MI_ERR_DISP_DEV_NOT_CONFIG	设备未配置
0xA00F8041	MI_ERR_DISP_DEV_NOT_ENABLE	设备未使能
0xA00F8042	EN_ERR_VO_DEV_HAS_ENABLED	设备已使能
0xA00F8043	MI_ERR_DISP_DEV_HAS_BINDED	设备已被绑定
0xA00F8044	MI_ERR_DISP_DEV_NOT_BINDED	设备未被绑定
0xA00F8045	MI_ERR_DISP_VIDEO_NOT_ENABLE	视频层未使能
0xA00F8046	MI_ERR_DISP_VIDEO_NOT_DISABLE	视频层未禁止
0xA00F8047	MI_ERR_DISP_VIDEO_NOT_CONFIG	视频层未配置
0xA00F8048	MI_ERR_DISP_CHN_NOT_DISABLE	输入端口未禁止
0xA00F8049	MI_ERR_DISP_CHN_NOT_ENABLE	输入端口未使能
0xA00F804A	MI_ERR_DISP_CHN_NOT_CONFIG	输入端口未配置
0xA00F804B	MI_ERR_DISP_CHN_NOT_ALLOC	输入端口未分配资源
0xA00F804C	MI_ERR_DISP_INVALID_PATTERN	无效样式
0xA00F804D	MI_ERR_DISP_INVALID_POSITION	无效位置
0xA00F804E	MI_ERR_DISP_WAIT_TIMEOUT	等待超时
0xA00F804F	MI_ERR_DISP_INVALID_VFRAME	无效视频帧
0xA00F8050	MI_ERR_DISP_INVALID_RECT_PARA	无效矩形参数
0xA00F8065	MI_ERR_DISP_GFX_NOT_DISABLE	图形层未关闭
0xA00F8066	MI_ERR_DISP_GFX_NOT_BIND	图形层未绑定
0xA00F8067	MI_ERR_DISP_GFX_NOT_UNBIND	图形层未解绑定

错误代码	宏定义	描述
0xA00F8068	MI_ERR_DISP_GFX_INVALID_ID	超出范围
0xA00F806b	MI_ERR_DISP_CHN_AREA_OVERLAP	输入端口区域重叠
0xA00F806d	MI_ERR_DISP_INVALID_LAYERID	视频层号超出范围
0xA00F806e	MI_ERR_DISP_VIDEO_HAS_BINDED	视频层已绑定
0xA00F806f	MI_ERR_DISP_VIDEO_NOT_BINDED	视频层未绑定