

# MI VDISP API

---

**Version 2.03**

© 2018 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

## TABLE OF CONTENTS

<b>REVISION HISTORY .....</b>	<b>i</b>
<b>TABLE OF CONTENTS.....</b>	<b>ii</b>
<b>1. API 参考 .....</b>	<b>1</b>
1.1. 概述.....	1
1.2. 功能模块 API .....	1
1.2.1 MI_VDISP_ModuleInit .....	2
1.2.2 MI_VDISP_ModuleDeinit .....	2
1.2.3 MI_VDISP_OpenDevice .....	4
1.2.4 MI_VDISP_CloseDevice .....	5
1.2.5 MI_VDISP_SetOutputPortAttr .....	5
1.2.6 MI_VDISP_GetOutputPortAttr .....	7
1.2.7 MI_VDISP_SetInputPortAttr .....	8
1.2.8 MI_VDISP_GetInputPortAttr .....	9
1.2.9 MI_VDISP_EnableInputPort .....	10
1.2.10 MI_VDISP_DisableInputPort .....	12
1.2.11 MI_VDISP_StartDev .....	13
1.2.12 MI_VDISP_StopDev .....	14
<b>2. 数据类型 .....</b>	<b>15</b>
2.1. MI_VDISP_DEV .....	16
2.2. MI_VDISP_PORT .....	16
2.3. MI_VDISP_OutputPortAttr_t .....	16
2.4. MI_VDISP_InputPortAttr_t .....	17
<b>3. 错误码 .....</b>	<b>19</b>

## 1. API 参考

---

### 1.1. 概述

MI\_VDISP 实现将多个前端模块的 output 或者用户的 inject 输入拼成一张 frame 输出的等功能。

### 1.2. 功能模块 API

API 名	功能
<a href="#"><u>MI_VDISP_ModuleInit</u></a>	<a href="#"><u>模块初始化</u></a>
<a href="#"><u>MI_VDISP_ModuleDeinit</u></a>	<a href="#"><u>模块去初始化</u></a>
<a href="#"><u>MI_VDISP_OpenDevice</u></a>	<a href="#"><u>打开一个 device</u></a>
<a href="#"><u>MI_VDISP_CloseDevice</u></a>	<a href="#"><u>关闭一个 device</u></a>
<a href="#"><u>MI_VDISP_SetOutputPortAttr</u></a>	<a href="#"><u>设置输出 port 属性</u></a>
<a href="#"><u>MI_VDISP_GetOutputPortAttr</u></a>	<a href="#"><u>获取输出 port 属性</u></a>
<a href="#"><u>MI_VDISP_SetInputPortAttr</u></a>	<a href="#"><u>设置输入 port 属性</u></a>
<a href="#"><u>MI_VDISP_GetInputPortAttr</u></a>	<a href="#"><u>获取输入 port 属性</u></a>
<a href="#"><u>MI_VDISP_EnableInputPort</u></a>	<a href="#"><u>enable 一个输入 port</u></a>
<a href="#"><u>MI_VDISP_DisableInputPort</u></a>	<a href="#"><u>disable 一个输入 port</u></a>
<a href="#"><u>MI_VDISP_StartDev</u></a>	<a href="#"><u>device 开始工作</u></a>
<a href="#"><u>MI_VDISP_StopDev</u></a>	<a href="#"><u>device 停止工作</u></a>

### 1.2.1 MI\_VDISP\_ModuleInit

➤ 功能

初始化 VDISP module。

➤ 语法

MI\_S32 MI\_VDISP\_ModuleInit(void);

➤ 形参

无

➤ 返回值

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

➤ 依赖

- 头文件：mi\_vdisp.h
- 库文件：libmi.a

※ 注意

无

➤ 举例

无

➤ 相关主题

[MI\\_VDISP\\_ModuleDeinit](#)

### 1.2.2 MI\_VDISP\_ModuleDeinit

➤ 功能

VDISP 去初始化。

➤ 语法

MI\_S32 MI\_VDISP\_ModuleDeinit(void);

➤ 形参

无

➤ 返回值

返回值 { 0 成功。  
非 0 失败, 参照[错误码 E:\svn\项目资料\I2 Module API\Mstar VDISP API V0.3.docx - 错误码。](#)

➤ 依赖

- 头文件：mi\_vdisp.h
- 库文件：libmi.a

※ 注意

无。

➤ 举例

无。

➤ 相关主题

无。

### 1.2.3 MI\_VDISP\_OpenDevice

➤ 功能

打开一个 VDISP device。

➤ 语法

MI\_S32 MI\_VDISP\_OpenDevice(MI\_VDISP\_DEV DevId)

➤ 形参

参数名称	描述	输入/输出
DevId	Device Id	输入

➤ 返回值

返回值 { 0 成功。  
非 0 失败, 参照[错误码 E:\svn\项目资料\I2\\_Module\\_API\Mstar\\_VDISP\\_API\\_V0.3.docx - 错误码](#)。

➤ 依赖

- 头文件：mi\_vdisp.h
- 库文件：libmi.a

※ 注意

无。

➤ 举例



无。

➤ 相关主题

无。

## 1.2.4 MI\_VDISP\_CloseDevice

➤ 功能

关闭一个 VDISP device。

➤ 语法

```
MI_S32 MI_VDISP_CloseDevice(MI_VDISP_DEV DevId);
```

➤ 形参

参数名称	描述	输入/输出
DevId	Device Id	输入

➤ 返回值

返回值 { 0 成功。  
非 0 失败,参照[错误码 E:\svn\项目资料\I2\\_Module\\_API\Mstar\\_VDISP\\_API\\_V0.3.docx - 错误码](#)。

➤ 依赖

- 头文件：mi\_vdisp.h
- 库文件：libmi.a

※ 注意

无。

➤ 举例

无。

➤ 相关主题

无。

## 1.2.5 MI\_VDISP\_SetOutputPortAttr

➤ 功能

设置输出端口属性。

➤ 语法

```
MI_S32 MI_VDISP_SetOutputPortAttr(MI_VDISP_DEV DevId, MI_VDISP_PORT PortId,  
MI_VDISP_OutputPortAttr_t *pstOutputPortAttr);
```

➤ 形参

参数名称	描述	输入/输出
DevId	Device Id	输入
PortId	Port Id	输入
pstOutputPortAttr	Output port 属性结构体指针	输入

➤ 返回值

返回值 { 0 成功。  
非 0 失败,参照[错误码 E:\svn\项目资料\I2\\_Module\\_API\Mstar\\_VDISP\\_API\\_V0.3.docx - 错误码。](#)

➤ 依赖

- 头文件：mi\_vdisp.h
- 库文件：libmi.a

※ 注意

无。

➤ 举例

无。

➤ 相关主题

无。

## 1.2.6 MI\_VDISP\_GetOutputPortAttr

➤ 功能

获取输出 port 属性。

➤ 语法

```
MI_S32 MI_VDISP_GetOutputPortAttr(MI_VDISP_DEV DevId, MI_VDISP_PORT PortId,
MI_VDISP_OutputPortAttr_t *pstOutputPortAttr);
```

➤ 形参

参数名称	描述	输入/输出
DevId	Device Id	输入
PortId	Port Id	输入
pstOutputPortAttr	Output port 属性结构体指针	输出

➤ 返回值

返回值 { 0 成功。  
非 0 失败,参照[错误码 E:\svn\项目资料\I2\\_Module\\_API\Mstar\\_VDISP\\_API\\_V0.3.docx - 错误码。](#)

➤ 依赖

- 头文件：mi\_vdisp.h
- 库文件：libmi.a

※ 注意

无。

➤ 举例

无。

➤ 相关主题

无。

## 1.2.7 MI\_VDISP\_SetInputPortAttr

➤ 功能

设置输入端口属性。

➤ 语法

```
MI_S32 MI_VDISP_SetInputPortAttr(MI_VDISP_DEV DevId, MI_VDISP_PORT
PortId, MI_VDISP_InputPortAttr_t *pstInputPortAttr);
```

➤ 形参

参数名称	描述	输入/输出
DevId	Device Id	输入
PortId	Port Id	输入
pstOutputPortAttr	Input port 属性结构体指针	输入

➤ 返回值

返回值 { 0 成功。  
非 0 失败, 参照[错误码 E:\svn\项目资料\I2 Module API\Mstar VDISP API V0.3.docx - 错误码](#)。

➤ 依赖

- 头文件：mi\_vdisp.h
- 库文件：libmi.a

※ 注意

无。

- 举例  
无。
- 相关主题  
无。

### 1.2.8 MI\_VDISP\_GetInputPortAttr

- 功能  
获取输入端口属性。

➤ 语法

```
MI_S32 MI_VDISP_GetInputPortAttr(MI_VDISP_DEV DevId, MI_VDISP_PORT
PortId, MI_VDISP_InputPortAttr_t *pstInputPortAttr);
```

➤ 形参

参数名称	描述	输入/输出
DevId	Device Id	输入
PortId	Port Id	输入
pstOutputPortAttr	Input port 属性结构体指针	输出

➤ 返回值

返回值 { 0 成功。  
非 0 失败, 参照[错误码 E:\svn\项目资料\I2\\_Module\\_API\Mstar\\_VDISP\\_API\\_V0.3.docx - 错误码](#)。

➤ 依赖

- 头文件：mi\_vdisp.h
- 库文件：libmi.a

※ 注意

无。

➤ 举例

无。

➤ 相关主题

无。

## 1.2.9 MI\_VDISP\_EnableInputPort

➤ 功能

Enable 一个输入端口。

➤ 语法

```
MI_S32 MI_VDISP_EnableInputPort(MI_VDISP_DEV DevId, MI_VDISP_PORT PortId);
```

➤ 形参

参数名称	描述	输入/输出
------	----	-------

DevId	Device Id	输入
PortId	Port Id	输入

➤ 返回值

返回值 { 0 成功。  
非 0 失败,参照[错误码 E:\svn\项目资料\I2\\_Module\\_API\Mstar\\_VDISP\\_API\\_V0.3.docx - 错误码。](#)

➤ 依赖

- 头文件：mi\_vdisp.h
- 库文件：libmi.a

※ 注意

无。

➤ 举例

无。

➤ 相关主题

无。

## 1.2.10 MI\_VDISP\_DisableInputPort

➤ 功能

Disable 一个输入端口。

➤ 语法

```
MI_S32 MI_VDISP_DisableInputPort(MI_VDISP_DEV DevId, MI_VDISP_PORT PortId);
```

➤ 形参

参数名称	描述	输入/输出
DevId	Device Id	输入
PortId	Port Id	输入

➤ 返回值

返回值 { 0 成功。  
非 0 失败,参照[错误码 E:\svn\项目资料\I2\\_Module\\_API\Mstar\\_VDISP\\_API\\_V0.3.docx - 错误码。](#)

➤ 依赖



- 头文件：mi\_vdisp.h
- 库文件：libmi.a

※ 注意

无。

➤ 举例

无。

➤ 相关主题

无。

### 1.2.11 MI\_VDISP\_StartDev

➤ 功能

开始一个 device 的工作。

➤ 语法

MI\_S32 MI\_VDISP\_StartDev(MI\_VDISP\_DEV DevId);

➤ 形参

参数名称	描述	输入/输出
DevId	Device Id	输入

➤ 返回值

返回值 { 0 成功。  
非 0 失败, 参照[错误码 E:\svn\项目资料\I2 Module API\Mstar VDISP API V0.3.docx - 错误码](#)。

➤ 依赖

- 头文件：mi\_vdisp.h
- 库文件：libmi.a

※ 注意

无。

➤ 功能

无。

- 相关主题  
无。

### 1.2.12 MI\_VDISP\_StopDev

- 功能  
停止一个 device 的工作。

- 语法  
MI\_S32 MI\_VDISP\_StopDev(MI\_VDISP\_DEV DevId);

- 形参

参数名称	描述	输入/输出
DevId	Device Id	输入

- 返回值
 

返回值 {
 

0    成功。  
 非 0    失败,参照[错误码 E:\svn\项目资料\I2\\_Module\\_API\Mstar\\_VDISP\\_API\\_V0.3.docx - 错误码](#)。

- 依赖
  - 头文件：mi\_vdisp.h
  - 库文件：libmi.a

- ※ 注意  
无。

- 举例  
无。

- 相关主题  
无。

## 2. 数据类型

---

MI VDISP 相关数据类型、数据结构定义如下：

数据结构	说明
<a href="#">MI_VDISP_DEV</a>	定义 device id 类型
<a href="#">MI_VDISP_PORT</a>	定义 port id 类型
<a href="#">MI_VDISP_OutputPortAttr_t</a>	定义输出端口属性
<a href="#">MI_VDISP_InputPortAttr_t</a>	定义输入端口属性

## 2.1. MI\_VDISP\_DEV

➤ 说明

定义 device id 类型。

➤ 定义

```
typedef MI_S32 MI_VDISP_DEV;
```

➤ 成员

无。

※ 注意事项

小于 0 为无效值

➤ 相关数据类型及接口

无。

## 2.2. MI\_VDISP\_PORT

➤ 说明

定义 port id 类型。

➤ 定义

```
typedef MI_S32 MI_VDISP_PORT;
```

➤ 成员

无。

※ 注意事项

小于 0 为无效值。

➤ 相关数据类型及接口

无。

## 2.3. MI\_VDISP\_OutputPortAttr\_t

➤ 说明

定义输出端口属性。

➤ 定义

```
typedef struct MI_VDISP_OutputPortAttr_s
{
    MI_U32 u32BgColor;    /* Background color of a output port, in YUV format. [23:16]:v, [15:8]:y,
[7:0]:u*/

    PIXEL_FORMAT_E ePixelFormat; /* pixel format of a output port */
    MI_U64 u64pts; /* current PTS */
    MI_U32 u32FrmRate; /* the frame rate of output port */
    MI_U32 u32Width; /* the frame width of a output port */
    MI_U32 u32Height; /* the frame height of a output port */
} MI_VDISP_OutputPortAttr_t;
```

➤ 成员

成员名称	描述
u32BgColor	输出 frame 的背景色, yuv 格式, [23:16]:v, [15:8]:y, [7:0]:u
ePixelFormat	输出 frame 的像素格式
u64pts	输出 frame 的当前 PTS
u32FrmRate	输出 frame 的帧率(置为 0:使用绑定端口时设定的值)
u32Width	输出 frame 的宽
u32Height	输出 frame 的高

※ 注意事项

无。

➤ 相关数据类型及接口

无。

## 2.4. MI\_VDISP\_InputPortAttr\_t

➤ 说明

定义输入端口属性。

➤ 定义

```
typedef struct MI_VDISP_InputPortAttr_s
{
    MI_U32 u32OutX; /* the output frame X position of this input port */
    MI_U32 u32OutY; /* the output frame Y position of this input port */
    MI_U32 u32OutWidth; /* the output frame width of this input port */
    MI_U32 u32OutHeight; /* the output frame height of this input port */
    MI_S32 s32IsFreeRun; /* is this port free run */
} MI_VDISP_InputPortAttr_t;
```

➤ 成员

成员名称	描述
u32OutX	输入 frame 在 output port 上的 X 位置
u32OutY	输入 frame 在 output port 上的 Y 位置
u32OutWidth	输入 frame 在 output port 上的宽
u32OutHeight	输入 frame 在 output port 上的高
s32IsFreeRun	如果非 0，则输入 frame 的时间戳不被检查

※ 注意事项

无。

➤ 相关数据类型及接口

无。

### 3. 错误码

---

错误码如表 3-1 所示：

表 3-1 API 错误码

错误代码	宏定义	描述
0	MI_SUCCESS	操作成功
0x240	E_MI_VDISP_ERR_MODINITED	VDISP 模块已经初始化过
0x241	E_MI_VDISP_ERR_DEVOPENED	VDISP 设备已被打开
0x242	E_MI_VDISP_ERR_DEVNOTOPEN	VDISP 设备未打开
0x243	E_MI_VDISP_ERR_DEVNOTSTOP	VDISP 设备未停止
0x244	E_MI_VDISP_ERR_DEVNOTCLOSE	VDISP 设备未关闭
0x245	E_MI_VDISP_ERR_PORTNOTDISABLE	VDISP 端口未 disable
0x246	E_MI_VDISP_ERR_PORTNOTUNBIND	VDISP 端口未取消绑定