# **MI DIVP API**

Version 2.03



# **REVISION HISTORY**

<b>Revision No.</b>	Description	Date
2.03	Initial release	04/12/2018

### **TABLE OF CONTENTS**

<b>RE</b>	VISIC	ON HISTORY	
		OF CONTENTS	
1.	API	参考	1
	1.1.	MI_DIVP_CreateChn	2
	1.2.	MI_DIVP_DestroyChn	3
	1.3.	MI_DIVP_DestroyChn MI_DIVP_SetChnAttr	3
	1.4.	MI_DIVP_GetChnAttr	۷ ک
	1.5.	MI_DIVP_StartChn	5
	1.6.	MI_DIVP_StopChn	6
	1.7.	MI_DIVP_SetOutputPortAttr	7
	1.8.	MI DIVP GetOutputPortAttr	8
	1.9.	MI_DIVP_RefreshChn	10
2.	DIV	P 数据类型	12
	2.1.	MI_DIVP_DiType_e	13
	2.2.	MI_DIVP_TnrLevel_e	13
		MI_DIVP_OutputPortAttr_t	
	2.4.	MI_DIVP_ChnAttr_t	17
3	DTV	D 無視和	10

# 1. API 参考

#### 该功能模块提供以下 API:

API 名	功能
MI_DIVP_CreateChn	创建一个 DIVP channel。
MI DIVP DestroyChn	销毁一个 DIVP channel。
MI_DIVP_SetChnAttr	设置 DIVP channel 的属性。
MI_DIVP_GetChnAttr	获取 DIVP channel <b>的属性。</b>
MI_DIVP_StartChn	开启一个通道。
MI_DIVP_StopChn	禁用一个通道。
MI_DIVP_SetOutputPortAttr	设置 DIVP output port 的属性。
MI_DIVP_GetOutputPortAttr	获取 DIVP output port 的属性。
MI DIVP RefreshChn	在暂停的状态下刷新某个 channel

### 1.1. MI\_DIVP\_CreateChn

▶ 功能

创建一个新的 DIVP channel。

▶ 语法

MI\_S32 MI\_DIVP\_CreateChn (
MI\_DIVP\_CHN\_DivpChn,
MI\_DIVP\_ChnAttr\_t\* pstAttr);

▶ 形参

参数名称	描述	输入/输出
DivpChn	返回被创建的 DIVP 通道的 ID,DIVP 最多支援 32 个通道,	输入
	取值范围[0,31]。	
pstAttr	DVIP 通道属性的指针,用于设定被创建的通道的属性。	输入

▶ 返回值

MI\_SUCCESS 成功创建一个新的 DIVP 通道。<br/>返回值 MI\_DIVP\_ERR\_FAILED 创建 DIVP 通道失败<br/>MI\_DIVP\_ERR\_NO\_RESOUCE 系统资源不足,创建 DIVP 通道失败

▶ 依赖

头文件: mi\_divp.h、mi\_divp\_datatype.h

库文件: libmi\_divp.so

※ 注意

DIVP 模块最多支援 32 个 channel, 当上层创建的 channel 总数大于 32 个或者系统资源不足时会创建失败。

▶ 举例

NA

▶ 相关主题

MI DIVP DestroyChn

# 1.2. MI\_DIVP\_DestroyChn

▶ 功能

销毁一个 DIVP 通道。

▶ 语法

MI\_S32 MI\_DIVP\_DestroyChn (MI\_DIVP\_CHN\_DivpChn);

▶ 形参

参数名称	描述	输入/输出
DivpChn	被销毁的 DVIP 通道的 ID	输入

▶ 返回值

```
返回值 MI_SUCCESS 成功销毁 DVIP 通道。

MI_DIVP_ERR_FAILED 销毁 DVIP 通道失败
```

▶ 依赖

头文件: mi\_divp.h、mi\_divp\_datatype.h

库文件: libmi\_divp.so

※ 注意

无

▶ 相关主题

MI\_DIVP\_CreateChn

# 1.3. MI\_DIVP\_SetChnAttr

▶ 功能

设置 DIVP channel 的属性。

▶ 语法

MI\_S32 MI\_DIVP\_SetChnAttr (
MI\_DIVP\_CHN\_DivpChn,
MI\_DIVP\_ChnAttr\_t\* pstAttr);

▶ 形参

参数名称	描述	输入/输出
DivpChn	DIVP 通道的 ID。	输入
pstAttr	设定的 DVIP 通道的属性的结构体指针。	输入

#### ▶ 返回值

版回值 MI\_SUCCESS 成功设定 DIVP 通道的属性 返回值 MI\_DIVP\_ERR\_FAILED 设置 DIVP 通道的属性失败 MI\_ERR\_INVALID\_PARAMETER 传入的参数有问题,请参考 MI\_DIVP\_ChnAttr\_t

▶ 依赖

头文件: mi\_divp.h、mi\_divp\_datatype.h

库文件: libmi\_divp.so

※ 注意

通道属性结构体成员中输入图像的最大宽度与最大高度为静态属性,通道创建后不能改变。

▶ 举例

无

▶ 相关主题

MI DIVP GetChnAttr

### 1.4. MI DIVP GetChnAttr

▶ 功能

获取 DIVP channel 的属性。

▶ 语法

MI\_S32 MI\_DIVP\_GetChnAttr(
MI\_DIVP\_CHN DivpChn,
MI\_DIVP\_ChnAttr\_t\* pstAttr);

#### ▶ 形参

参数名称	描述	输入/输出
DivpChn	DIVP 通道的 ID。	输入
pstAttr	传回的 DVIP 通道的属性的结构体指针。	输出

▶ 返回值

MI\_SUCCESS 成功获取 DIVP 通道的属性。 返回值 MI\_DIVP\_ERR\_FAILED 获取 DIVP 通道的属性失败。

▶ 依赖

头文件: mi\_divp.h、mi\_divp\_datatype.h

库文件: libmi\_divp.so

※ 注意

第一次获取时返回创建时的初始值。

▶ 举例

 $NA_{\circ}$ 

▶ 相关主题

MI\_DIVP\_SetChnAttr

# 1.5. MI\_DIVP\_StartChn

▶ 功能

开启一个通道。

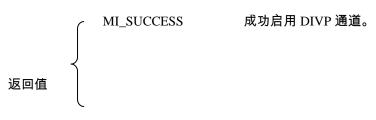
▶ 语法

 $MI\_S32\ MI\_DIVP\_StartChn\left(\underline{MI\ DIVP\ CHN}\ DivpChn\right);$ 

▶ 形参

参数名称	描述	输入/输出
DivpChn	DIVP 通道的 ID。	输入

▶ 返回值



MI\_DIVP\_ERR\_FAILED 启用 DIVP 通道失败。

▶ 依赖

头文件: mi\_divp.h、mi\_divp\_datatype.h

库文件: libmi\_divp.so

※ 注意

通道必须已经被创建,且没有被销毁。

▶ 举例

 $NA_{\text{o}}$ 

▶ 相关主题

MI DIVP StopChn

### 1.6. MI\_DIVP\_StopChn

▶ 功能

禁用一个通道。

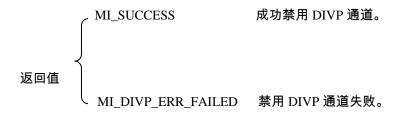
▶ 语法

MI\_S32 MI\_DIVP\_StopChn (MI\_DIVP\_CHN DivpChn);

▶ 形参

参数名称	描述	输入/输出
DivpChn	DIVP 通道的 ID。	输入

▶ 返回值



▶ 依赖

头文件: mi\_divp.h、mi\_divp\_datatype.h

库文件: libmi\_divp.so

※ 注意

重复禁用一个通道会返回 MI\_SUCCESS。

▶ 举例

 $NA_{\circ}$ 

▶ 相关主题

MI DIVP StartChn

# 1.7. MI\_DIVP\_SetOutputPortAttr

▶ 功能

DIVP 通道上 output port 的属性。

▶ 语法

```
MI_S32 MI_DIVP_SetOutputPortAttr (

MI_DIVP_CHN_DivpChn,

MI_DIVP_OutputPortAttr_t * pstOutputPortAttr);
```

#### ▶ 形参

参数名称	描述	输入/输出
DivpChn	DIVP 通道的 ID。	输入
pstOutputPortAttr	Output port的属性的指针。	输入

4/12/2018

#### 返回值

成功设定 DIVP 通道绑定的 output port 的属性。 MI\_SUCCESS 返回值 MI\_DIVP\_ERR\_FAILED 设定 DIVP 通道绑定的 output port 的属性失败。

依赖

头文件: mi\_divp.h、mi\_divp\_datatype.h

库文件: libmi\_divp.so

Ж 注意

无。

举例

NΑ。

相关主题

MI\_DIVP\_GetOutputPortAttr

# 1.8. MI\_DIVP\_GetOutputPortAttr

功能

获取 DIVP output port 的属性。

语法

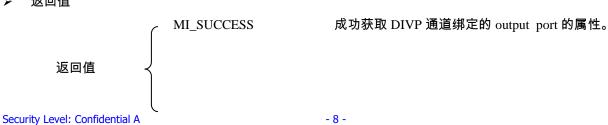
MI\_S32 MI\_DIVP\_GetOutputPortAttr( MI DIVP CHN DivpChn,

MI DIVP OutputPortAttr t \* pstOutputPortAttr);

#### 形参

参数名称	描述	输入/输出
DivpChn	DIVP 通道的 ID。	输入
pstOutputPortAttr	Output port 的属性的指针	输出

#### 返回值



MI\_DIVP\_ERR\_FAILED 获取 DIVP 通道绑定的 output port 的属性失败。

▶ 依赖

头文件: mi\_divp.h、mi\_divp\_datatype.h

库文件: libmi\_divp.so

※ 注意

无。

▶ 举例

NA

▶ 相关主题

MI DIVP SetOutputPortAttr

# 1.9. MI\_DIVP\_RefreshChn

▶ 功能

暂停状态下刷新 DIVP channel。

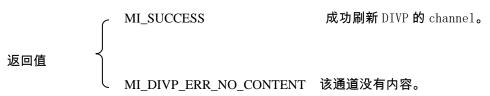
▶ 语法

MI\_S32 MI\_DIVP\_RefreshChn (
MI\_DIVP\_CHN DivpChn);

▶ 形参

参数名称	描述	输入/输出
DivpChn	被刷新的 channel 的 ID。	输入

▶ 返回值



▶ 依赖

头文件: mi\_divp.h、mi\_divp\_datatype.h

库文件: libmi\_divp.so

※ 注意

该接口适用于 channel 在暂停的状态下重新刷新。

Version 2.03

▶ 举例

NA

相关主题

NA

# 2. DIVP 数据类型

#### DIVP 相关数据类型定义如下:

MI_DIVP_DiType_e	定义 DIVP 的 deinterlace 的类型。	
MI_DIVP_TnrLevel_e	定义 DIVP TNR 的等级。	
MI DIVP OutputPortAttr t	定义 DIVP 绑定的 output port 的属性参数。	
MI_DIVP_ChnAttr_t	定义 DIVP 通道的属性参数。	
MI_DIVP_CHN	DIVP 通道的 ID。	

注:本节已涵盖各重要的数据类型,部分未列出数据类型请参见 mi\_divp\_datatype.h

# 2.1. MI\_DIVP\_DiType\_e

▶ 说明

定义 DIVP 的 deinterlace 的类型。

▶ 定义

```
typedef enum
{
    E_MI_DIVP_DITYPE_OFF, //off
    E_MI_DIVP_DITYPE_2D, ///2.5D DI
    E_MI_DIVP_DITYPE_3D, ///3D DI
    E_MI_DIVP_DITYPE_NUM,
} MI DIVP DiType e;
```

#### ▶ 成员

成员	描述
E_MI_DIVP_DITYPE_OFF	DIVP 通道上关闭 deinterlace。
E_MI_DIVP_DITYPE_2D	DIVP 通道上开启 2.5D deinterlace。
E_MI_DIVP_DITYPE_3D	DIVP 通道上开启 3D deinterlace。
E_MI_DIVP_DITYPE_NUM	DIVP 通道上 deinterlace 类型个数。

#### ※ 注意事项

1 开启 DI 时必须开 TNR, MSR930 只能支援 3D DI, TNR level 设定为 E\_MI\_DIVP\_TNRLEVEL\_MIDDLE。
2 3D DI 与 rotation 冲突,两个功能不能同时打开。

#### ▶ 相关数据类型及接口

无。

### 2.2. MI\_DIVP\_TnrLevel\_e

▶ 说明

定义 DIVP TNR 的等级。

▶ 定义

```
typedef enum
{
    E_MI_DIVP_TNRLEVEL_OFF,
    E_MI_DIVP_TNRLEVEL_LOW,
    E_MI_DIVP_TNRLEVEL_MIDDLE,
    E_MI_DIVP_TNRLEVEL_HIGH,
```

E\_MI\_DIVP\_TNRLEVEL\_NUM,
} MI\_DIVP\_TnrLevel\_e;

#### ▶ 成员

成员	描述
E_MI_DIVP_TNRLEVEL_OFF	DIVP 通道上关闭 TNR。
E_MI_DIVP_TNRLEVEL_LOW	DIVP 通道上开启弱等级的 TNR。
E_MI_DIVP_TNRLEVEL_MIDDLE	DIVP 通道上开启中等等级的 TNR。
E_MI_DIVP_TNRLEVEL_HIGH	DIVP 通道上开启强等级的 TNR。
E_MI_DIVP_TNRLEVEL_NUM	DIVP 通道上 TNR 等级的数量。

#### ※ 注意事项

MSR930 支援 TNR,但是不支持调节 TNR level。

相关数据类型及接口

无。

# 2.3. MI\_DIVP\_OutputPortAttr\_t

▶ 说明

定义 DIVP 绑定的 output port 的属性参数。

#### ▶ 定义

```
typedef struct MI_DIVP_OutputPortAttr_s
{
         MI_U32 u32Width;//output width
         MI_U32 u32Height;//output height
         MI_SYS_PixelFormat_e ePixelFormat;
         MI_SYS_CompressMode_e eCompMode;//compress mode
}MI_DIVP_OutputPortAttr_t;
```

#### ▶ 成员

成员名称	描述	
u32Width	DIVP 通道输出画面的宽度。	
u32Height	DIVP 通道输出画面的高度。	
ePixelFormat	DIVP 通道输出画面的像素格式。	
eCompMode	DIVP 通道输出图像的压缩格式,DIVP 通道只能输出	
	非压缩格式的图像。	

#### ※ 注意事项

1. MSR930 的 DIVP 通道只能输出非压缩格式的图像,输出的最小 size 为 128x64。输出的 pixel format 只能 支援E\_MI\_SYS\_PIXEL\_FRAME\_YUV422\_YUYV,E\_MI\_SYS\_PIXEL\_FRAME\_YUV\_SEMIPLANAR\_420

和 E\_MI\_SYS\_PIXEL\_FRAME\_YUV\_MST\_420 (这个格式只支持将输出送给 DISP 模块)。

2. 另外在 SSC329Q, SSC326D, SSA520, SSC328Q 支持的 pixel 格式为:

E\_MI\_SYS\_PIXEL\_FRAME\_YUV422\_YUYV

E\_MI\_SYS\_PIXEL\_FRAME\_YUV\_SEMIPLANAR\_420

E\_MI\_SYS\_PIXEL\_FRAME\_ARGB8888,

E\_MI\_SYS\_PIXEL\_FRAME\_ABGR8888,

E\_MI\_SYS\_PIXEL\_FRAME\_RGB565

#### ▶ 相关数据类型及接口

无。

### 2.4. MI\_DIVP\_ChnAttr\_t

▶ 说明

定义 DIVP 通道的属性参数。

▶ 定义

```
typedef struct MI_DIVP_ChnAttr_s
{
    MI_U32 u32MaxWidth; //support max input width
    MI_U32 u32MaxHeight; //support max input height
    MI_DIVP_TnrLevel_e eTnrLevel; //TNR level
    MI_DIVP_DiType_e eDiType; //DI type
    MI_SYS_Rotate_e eRotateType; //rotate angle
    MI_SYS_WindowRect_t stCropRect; //crop information
    MI_BOOL bHorMirror; //horizontal mirror
    MI_BOOL bVerMirror; //vertical mirror
} MI_DIVP_ChnAttr_t;
```

#### ▶ 成员

成员名称	描述	
u32MaxWidth	DIVP 通道支援的 input 的最大宽度。	
u32MaxHeight	DIVP 通道支援的 input 的最大高度。	
eTnrLevel	DIVP 通道上 TNR 的等级。	
eDiType	DIVP 通道上 DI 的类型。	
eRotateType	DIVP 通道上画面旋转的角度。	
stCropRect	DIVP 通道上的 crop 信息。	
bHorMirror	DIVP 通道上水平方向翻转	
bVerMirror	DIVP 通道上垂直方向翻转	

#### ※ 注意事项

DIVP 通道上支援的图像高度为静态属性,通道创建后不可以改变。该属性会影响该通道占用系统资源,影响后面继续创建通道。Crop window 的宽度和高度必须要 2 对齐。MSR930 的 DIVP 只支援旋转 90 度和 270 度,DIVP 旋转时只支持对 pixel format 为 E\_MI\_SYS\_PIXEL\_FRAME\_YUV\_SEMIPLANAR\_420 的 input。在旋转时 input source 的宽度和高度必须 64 对齐,支持旋转的最大 size 为 1920x1920。MSR930 不支持输出压缩格式的图像。 注:SSC329Q、SSC328D、SSC328Q 平台支持 bHorMirror / bVerMirror。

#### ▶ 相关数据类型及接口

无。

# 3. DIVP 错误码

DIVP API 返回值如<u>表 3-1</u>所示:

表 3-1 DIVP API 返回值

错误代码	宏定义	描述
0x0	MI_SUCCESS	succeeded
0xa00c2002	MI_DIVP_ERR_INVALID_CHNID	无效的 channel ID。
0xa00c2003	MI_DIVP_ERR_INVALID_PARAM	传入的参数无效
0xa00c2006	MI_DIVP_ERR_NULL_PTR	空指针异常
0xa00c201c	MI_DIVP_ERR_FAILED	DIVP 的操作失败
0xa00c2005	MI_DIVP_ERR_NO_RESOUCE	无资源可以使用
0xa00c201c	MI_DIVP_ERR_NO_CONTENT	通道中无显示内容