

MI DVP API

Version 2.04

© 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">Added new API:<ul style="list-style-type: none">MI DIVP StretchBuf	12/16/2019

TABLE OF CONTENTS

REVISION HISTORY	i
TABLE OF CONTENTS.....	ii
1. API 参考	1
1.1. MI_DIVP_CreateChn	2
1.2. MI_DIVP_DestroyChn	3
1.3. MI_DIVP_SetChnAttr	3
1.4. MI_DIVP_GetChnAttr	4
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
1.10. MI_DIVP_StretchBuf	11
2. DIVP 数据类型	13
2.1. MI_DIVP_DiType_e	14
2.2. MI_DIVP_TnrLevel_e	14
2.3. MI_DIVP_OutputPortAttr_t	16
2.4. MI_DIVP_ChnAttr_t	18
2.5. MI_DIVP_DirectBuf_t	19
3. DIVP 错误码.....	21

1. API 参考

该功能模块提供以下 API:

API 名	功能
<u>MI_DVP_CreateChn</u>	创建一个 DVP channel。
<u>MI_DVP_DestroyChn</u>	销毁一个 DVP channel。
<u>MI_DVP_SetChnAttr</u>	设置 DVP channel 的属性。
<u>MI_DVP_GetChnAttr</u>	获取 DVP channel 的属性。
<u>MI_DVP_StartChn</u>	开启一个通道。
<u>MI_DVP_StopChn</u>	禁用一个通道。
<u>MI_DVP_SetOutputPortAttr</u>	设置 DVP output port 的属性。
<u>MI_DVP_GetOutputPortAttr</u>	获取 DVP output port 的属性。
<u>MI_DVP_RefreshChn</u>	在暂停的状态下刷新某个 channel
<u>MI_DVP_StretchBuf</u>	缩放、拉伸、裁剪指定内存中的图像数据

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, 取值范围[0, 31]。	输入
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

返回值 {

MI_SUCCESS	成功创建一个新的 DIVP 通道。
MI_DIVP_ERR_FAILED	创建 DIVP 通道失败
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 DIVP 通道的属性的结构体指针。	输入

➤ 返回值

返回值	MI_SUCCESS	成功设定 DIVP 通道的属性
	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 DIVP 通道的属性的结构体指针。	输出

➤ 返回值

返回值	{	MI_SUCCESS	成功获取 DIVP 通道的属性。
		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。

➤ 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 ([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

返回值	{	MI_SUCCESS	成功启用 DIVP 通道。
		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。

➤ 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

返回值	{	MI_SUCCESS	成功禁用 DIVP 通道。
		MI_DIVP_ERR_FAILED	禁用 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

重复禁用一个通道会返回 MI_SUCCESS。

➤ 举例

NA。

➤ 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

➤ 返回值

返回值	{	MI_SUCCESS	成功设定 DIVP 通道绑定的 output port 的属性。
		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

※ 注意

无。

➤ 举例

NA。

➤ 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

返回值	{	MI_SUCCESS	成功获取 DIVP 通道绑定的 output port 的属性。

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

返回值	{	MI_SUCCESS	成功刷新 DIVP 的 channel。
		MI_DIVP_ERR_NO_CONTENT	该通道没有内容。

➤ 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。

➤ 举例

NA

➤ 相关主题

NA

1.10. MI_DIVP_StretchBuf

➤ 功能

缩放、拉伸、裁剪指定内存区域中的图像数据，并输出到目标内存。

➤ 语法

MI_S32 MI_DIVP_StretchBuf([MI_DIVP_DirectBuf_t](#) *pstSrcBuf, MI_SYS_WindowRect_t *pstSrcCrop, [MI_DIVP_DirectBuf_t](#) *pstDstBuf);

➤ 形参

参数名称	描述	输入/输出
pstSrcBuf	用来存放待处理图像内存信息的结构体指针	输入
pstSrcCrop	图像裁剪相关信息的结构体指针	输入
pstDstBuf	保存处理后图像内存信息的结构体指针	输入

➤ 返回值

返回值 {
MI_SUCCESS 图像处理成功。
MI_DIVP_ERR_FAILED 图像处理失败。

➤ 依赖

头文件：mi_divp.h、mi_divp_datatype.h

库文件：libmi_divp.so

※ 注意

该接口只支持 YUV420SP 或者 ARGB8888 的像素格式。

➤ 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：

<u>MI_DIVP_DiType_e</u>	定义 DIVP 的 deinterlace 的类型。
<u>MI_DIVP_TnrLevel_e</u>	定义 DIVP TNR 的等级。
<u>MI_DIVP_OutputPortAttr_t</u>	定义 DIVP 绑定的 output port 的属性参数。
<u>MI_DIVP_ChnAttr_t</u>	定义 DIVP 通道的属性参数。
<u>MI_DIVP_CHN</u>	DIVP 通道的 ID。
<u>MI_DIVP_DirectBuf_t</u>	用于图像处理的内存信息结构体

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_DVP_TNRLEVEL_OFF	DVP 通道上关闭 TNR。
E_MI_DVP_TNRLEVEL_LOW	DVP 通道上开启弱等级的 TNR。
E_MI_DVP_TNRLEVEL_MIDDLE	DVP 通道上开启中等等级的 TNR。
E_MI_DVP_TNRLEVEL_HIGH	DVP 通道上开启强等级的 TNR。
E_MI_DVP_TNRLEVEL_NUM	DVP 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_DVP_OutputPortAttr_t

➤ 说明

定义 DVP 绑定的 output port 的属性参数。

➤ 定义

```
typedef struct MI_DVP_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_DVP_OutputPortAttr_t;
```

➤ 成员

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

※ 注意事项

- MSR930 的 DVP 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

DVP 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 DVP 只支援旋转 90 度和 270 度，DVP 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。

2.5. MI_DVP_DirectBuf_t

- 说明

定义用于图像处理的内存信息

- 定义

```
typedef struct MI_DVP_DirectBuf_s
{
    MI_SYS_PixelFormat_e ePixelFormat; //YUV420SP or ARGB888 only
    MI_U32 u32Width;
    MI_U32 u32Height;
    MI_U32 u32Stride[3];
    MI_PHY phyAddr[3];
}MI_DVP_DirectBuf_t;
```

- 成员

成员名称	描述
ePixelFormat	图像的像素格式
u32Width	图像画面的宽度
u32Height	图像画面的高度
u32Stride	图像每行所占字节数
phyAddr	Buffer 的起始物理地址

- 注意事项

只支持 YUV420SP 或者 ARGB8888 的图像像素格式

- 相关数据类型及接口

[MI_DVP_StretchBuf](#)

3. DIVP 错误码

DIVP API 返回值如表 3-1 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