# **MI SENSOR API**

Version 2.05



# **REVISION HISTORY**

<b>Revision No.</b>	Description	Date
2.03	Initial release	11/08/2018
2.04	Added MI_SNR_CustFunction api	11/08/2019
	<ul> <li>Added MI_SNR_CUST_DIR_e</li> </ul>	
	<ul> <li>Added bEarlyInit to MI_SNR_PADInfo_t</li> </ul>	
	<ul> <li>Added Shutter/Gain to MI_SNR_PlaneInfo_t</li> </ul>	
2.05	<ul> <li>Updated description of MI_SNR_SetFps and MI_SNR_GetFps</li> </ul>	12/17/2019

# **TABLE OF CONTENTS**

			DRY	
			NTS	
1.	API	参考		1
	1.1.	概述		1
	1.2.	模块 AP		1
		1.2.1	MI SNR Enable	2
		1.2.2	MI_SNR_Disable	
		1.2.3	MI_SNR_GetPadInfo	
		1.2.4	MI_SNR_GetPlaneInfo	
		1.2.5	MI_SNR_GetFps	6
		1.2.6	MI_SNR_SetFps	7
		1.2.7	MI_SNR_GetBT656SrcType	8
		1.2.8	MI_SNR_QueryResCount	9
		1.2.9	MI_SNR_GetRes	10
		1.2.10	MI_SNR_GetCurRes	14
		1.2.11	MI_SNR_SetRes	15
		1.2.12	MI_SNR_SetOrien	16
		1.2.13	MI_SNR_GetOrien	17
		1.2.14	MI_SNR_SetPlaneMode	18
			MI_SNR_GetPlaneMode	
2.	SEN	SOR 数据	类型	22
	2.1.	MI_SNR	_MAX_PADNUM	23
	2.2.	MI_SNR	_MAX_PLANENUM	23
	2.3.	_	_PAD_ID_e	
	2.4.	MI_SNR	_HDRSrc_e	24
	2.5.	MI_SNR	_HDRHWMode_e	25
	2.6.	MI_SNR	_Anadec_SrcType_e	26
	2.7.	_	_Res_t	
	2.8.	_	_Res_List_t	
		_	_AttrParallel_t	
			_MipiAttr_t	
			_AttrBt656_t	
			_IntfAttr_u	
		_	_PADInfo_t	
_		_	_PlaneInfo_t	
3	SFN	SOR 错误	······································	37

# 1. API 参考

# 1.1. 概述

SNR(sensor)实现获取摄像头接口信息、调整分辨率和帧率等功能。

# 1.2. 模块 API

API名	功能
MI_SNR_Enable	设置Sensor使能
MI SNR Disable	设置Sensor失能
MI_SNR_GetPadInfo	获取Sensor设备信息
MI_SNR_GetPlaneInfo	获取Sensor通道信息
MI_SNR_GetFps	获取Sensor当前帧率
MI_SNR_SetFps	设置Sensor帧率
MI_SNR_GetBT656SrcType	获取BT656 Sensor源输入格式
MI SNR QueryResCount	获取Sensor支持分辨率的数量
MI_SNR_GetRes	获取索引对应的sensor分辨率
MI SNR GetCurRes	获取Sensor当前分辨率
MI_SNR_SetRes	设置Sensor分辨率
MI SNR GetOrien	获取Sensor翻转属性
MI_SNR_SetOrien	设置Sensor翻转
MI SNR SetPlaneMode	设置Sensor通道模式
MI_SNR_GetPlaneMode	获取设置的Sensor通道模式
MI_SNR_CustFunction	设置Sensor客制化功能

# 1.2.1 MI\_SNR\_Enable

▶ 功能

设置 SENSOR 对应设备的使能

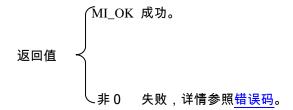
▶ 语法

MI\_S32 MI\_SNR\_Enable(MI\_SNR\_PAD\_ID\_e ePADId);

▶ 形参

参数名称	描述	输入/输出
ePADId	SENSOR 设备号。	输入
	取值范围: [0, MI_SNR_MAX_PAD_NUM]。	

▶ 返回值



▶ 依赖

• 头文件: mi\_sensor\_datatype.h、mi\_sensor.h

• 库文件: libmi\_sensor.a

※ 注意

◆ 在调用前要保证 SENSOR设备处于未初始化状态。如果SENSOR设备已处于使能状态,可以使用MI SNR Disable来去初始化设备。

▶ 相关主题

MI SNR Disable

# 1.2.2 MI\_SNR\_Disable

▶ 功能

设置 SENSOR 对应设备失能。

▶ 语法

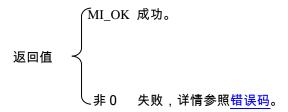
### MI SENSOR API Version 2.05

MI\_S32 MI\_SNR\_Disable(MI\_SNR\_PAD\_ID\_e ePADId);

#### ▶ 形参

参数名称	描述	输入/输出
ePADId	SENSOR设备号。	输入
	取值范围: [0, MI SNR MAX PAD NUM)。	

### ▶ 返回值



#### ▶ 依赖

• 头文件: mi\_sensor\_datatype.h、mi\_sensor.h

• 库文件: libmi\_sensor.a

#### ▶ 相关主题

MI SNR Enable

# 1.2.3 MI\_SNR\_GetPadInfo

▶ 功能

获取 SENSOR 设备信息。

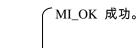
#### ▶ 语法

MI\_S32 MI\_SNR\_GetPadInfo(<u>MI\_SNR\_PAD\_ID\_e</u> ePADId, <u>MI\_SNR\_PADInfo\_t</u> \*pstPadInfo);

#### ▶ 形参

参数名称	描述	输入/输出
ePADId	SENSOR设备号。	输入
	取值范围: [0, MI_SNR_MAX_PAD_NUM)。	
pstPadInfo	SENSOR设备属性指针	输出

#### ▶ 返回值



#### 返回值

非 0 失败,详情参照错误码。

### ▶ 依赖

• 头文件: mi\_sensor\_datatype.h、mi\_sensor.h

• 库文件: libmi\_sensor.a

### > 相关主题

无。

# 1.2.4 MI\_SNR\_GetPlaneInfo

▶ 功能

获取 SENSOR 通道信息。

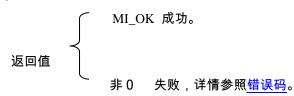
### ▶ 语法

MI\_S32 MI\_SNR\_GetPadInfo(MI\_SNR\_PAD\_ID\_e ePADId, MI\_SNR\_PlaneInfo\_t \*pstPadInfo);

#### ▶ 形参

参数名称	描述	输入/输出
ePADId	SENSOR设备号。	输入
	取值范围: [0, MI_SNR_MAX_PAD_NUM)。	
u32PlaneID	SENSOR通道号。	输入
	取值范围: [0, MI SNR MAX PLANE NUM)。	
pstChnInfo	SENSOR通道信息。	输出

### ▶ 返回值



#### ▶ 依赖

• 头文件: mi\_sensor\_datatype.h、mi\_sensor.h

• 库文件: libmi\_sensor.a

▶ 相关主题

无。

# 1.2.5 MI\_SNR\_GetFps

▶ 功能

获取 SENSOR 帧率

▶ 语法

MI\_S32 MI\_SNR\_GetFps(MI\_SNR\_PAD\_ID\_e ePADId, MI\_U32 \*pFps);

▶ 形参

参数名称	描述	输入/输出
ePADId	SENSOR设备号。 取值范围: [0, MI_SNR_MAX_PAD_NUM)。	输入
pFps	帧率指针	输出

▶ 返回值

MI\_OK 成功。

返回值

非 0 失败,详情参照错误码。

▶ 依赖

• 头文件: mi\_sensor\_datatype.h、mi\_sensor.h

• 库文件: libmi\_sensor.a

※ 注意

获取到的 fps 范围为:

min\*1000 < fps < max\*1000: 精确到小数点后 3 位。

▶ 相关主题

MI SNR SetFps

# 1.2.6 MI\_SNR\_SetFps

▶ 功能

设置 SENSOR 帧率。

▶ 语法

MI\_S32 MI\_SNR\_SetFps(<u>MI\_SNR\_PAD\_ID\_e\_</u>ePADId, MI\_U32 \*pFps);

#### ▶ 形参

参数名称	描述	输入/输出
ePADId	SENSOR设备号。 取值范围: [0, MI SNR MAX PAD NUM)。	输入
pFps	帧率指针	输出

#### ▶ 返回值

#### ▶ 依赖

• 头文件: mi\_sensor\_datatype.h、mi\_sensor.h

• 库文件: libmi\_sensor.a

#### ※ 注意

fps 有两个取值区间:

min < fps < max: 精确到个位数

min\*1000 < fps < max\*1000: 精确到小数点后 3 位。

#### ► 相关主题

MI SNR GetFps

# 1.2.7 MI\_SNR\_GetBT656SrcType

▶ 功能

获取 BT656 源输入格式。

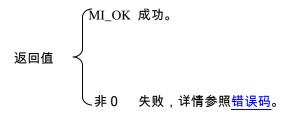
# ▶ 语法

MI\_S32 MI\_SNR\_GetBT656SrcType(<u>MI\_SNR\_PAD\_ID\_e</u> ePADId, MI\_U32 u32PlaneID, <u>MI\_SNR\_Anadec\_SrcType\_e</u> \*psttype);

#### ▶ 形参

参数名称	描述	输入/输出
ePADId	SENSOR设备号。 取值范围: [0, MI SNR MAX PAD NUM)。	输入
u32PlaneID	SENSOR通道号。 取值范围: [0, MI_SNR_MAX_PLANE_NUM)。	输入
psttype	源输入格式	输出

#### ▶ 返回值



#### ▶ 依赖

• 头文件: mi\_sensor\_datatype.h、mi\_sensor.h

• 库文件: libmi\_sensor.a

### ※ 注意

在BT656 sensor上才可以用。

#### ▶ 举例

无。

# ▶ 相关主题

MI SNR Anadec SrcType e

# 1.2.8 MI\_SNR\_QueryResCount

▶ 功能

获取 SENSOR 支持分辨率的数量。

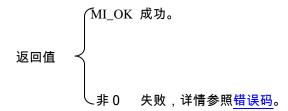
#### ▶ 语法

MI\_S32 MI\_SNR\_QueryResCount(<u>MI\_SNR\_PAD\_ID\_e</u> ePADId, MI\_U32 \*pu32ResCount);

# ▶ 形参

参数名称	描述	输入/输出
ePADId	SENSOR设备号。	输入
	取值范围: [0, MI SNR MAX PAD NUM)。	
*pu32ResCount	SENSOR设备支持的resolution 数量	输出

#### ▶ 返回值



#### ▶ 依赖

• 头文件: mi\_sensor\_datatype.h、mi\_sensor.h

• 库文件: libmi\_sensor.a

#### ※ 注意

无。

▶ 举例

无。

▶ 相关主题

MI SNR GetRes

# 1.2.9 MI\_SNR\_GetRes

▶ 功能

获取 resolution 映射表中索引对应的分辨率。

▶ 语法

MI\_S32 MI\_SNR\_GetRes(MI\_SNR\_PAD\_ID\_e ePADId, MI\_U8 u8ResIdx,

### **MI SENSOR API** Version 2.05

MI\_SNR\_Res\_t \*pstRes);

#### ▶ 形参

参数名称	描述	输入/输出
ePADId	SENSOR设备号。	输入
	取值范围: [0, MI SNR MAX PAD NUM)。	
u8ResIdx	分辨率映射表中的索引	输入
*pstRes	序号所对应的分辨率	输出

#### ▶ 返回值

#### ▶ 依赖

- 头文件: mi\_sensor\_datatype.h、mi\_sensor.h
- 库文件: libmi sensor.a

#### ※ 注意

无

#### ▶ 举例

```
MI_U32 u32ResCount = 0;
MI_U8 u8ResIndex =0;
MI_U8 u8ChocieRes =0;
MI SNR QueryResCount(E_MI_SNR_PAD_ID_0, &u32ResCount);
for(u8ResIndex=0; u8ResIndex < u32ResCount; u8ResIndex++)</pre>
{
    MI SNR GetRes(E_MI_SNR_PAD_ID_0, u8ResIndex, &stRes);
    printf("index %d, Crop(%d,%d,%d), outputsize(%d,%d), maxfps %d, minfps %d,
              ResDesc %s\n",u8ResIndex, stRes.stCropRect.u16X, stRes.stCropRect.u16Y,
         stRes.stCropRect.u16Width,stRes.stCropRect.u16Height,stRes.stOutputSize.u16Width,
         stRes.stOutputSize.u16Height, stRes.u32MaxFps,stRes.u32MinFps, stRes.strResDesc);
}
printf("select res\n");
scanf("%c", &select);
MI SNR SetPlaneMode(E_MI_SNR_PAD_ID_0, FALSE);
MI SNR SetRes(E_MI_SNR_PAD_ID_0,u8ResIdx);
MI SNR Enable(E_MI_SNR_PAD_ID_0);
```

# **MI SENSOR API**

Version 2.05

#### ▶ 相关主题

MI SNR QueryResCount MI SNR Res t

# 1.2.10 MI\_SNR\_GetCurRes

▶ 功能

获取 sensor 当前分辨率和在分辨率映射表中的位置。

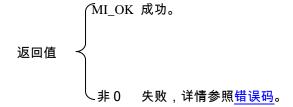
#### ▶ 语法

MI\_S32 MI\_SNR\_GetCurRes(<u>MI\_SNR\_PAD\_ID\_e</u> ePADId, MI\_U8 \*pu8CurResIdx, <u>MI\_SNR\_Res\_t</u> \*pstCurRes);

#### ▶ 形参

参数名称	描述	输入/输出
ePADId	SENSOR设备号。	输入
	取值范围: [0, MI_SNR_MAX_PAD_NUM)。	
*pu8CurResIdx	当前分辨率的索引	输出
*pstCurRes	当前分辨率信息	输出

#### ▶ 返回值



#### ▶ 依赖

• 头文件: mi\_sensor\_datatype.h、mi\_sensor.h

• 库文件: libmi\_sensor.a

#### ※ 注意

无。

# ▶ 举例

参考 MI\_SNR\_GetRes 举例

▶ 相关主题

MI SNR Res t

# 1.2.11 MI\_SNR\_SetRes

▶ 功能

设置 sensor 设备输出分辨率

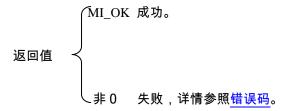
▶ 语法

MI\_S32 MI\_SNR\_SetRes(<u>MI\_SNR\_PAD\_ID\_e</u> ePADId, MI\_U8 u8ResIdx);

#### ▶ 形参

参数名称	描述	输入/输出
stVifDevMap	Dev 和SensorPad Maping 关系	输入
u8Length	Dev Num	输入

### ▶ 返回值



- ▶ 依赖
- 头文件: mi\_sensor\_datatype.h、mi\_sensor.h
- 库文件: libmi\_sensor.a
- ※ 注意

在默认情况下 vif Dev 和 SensorPad 对应关系是 vif Dev0 -> SensorPad0, vif Dev1 -> SensorPad1.

▶ 举例

参考 MI\_SNR\_GetRes 举例

▶ 相关主题

MI SNR GetRes

### MI\_SNR\_Res\_t

# 1.2.12 MI\_SNR\_SetOrien

▶ 功能

设置 sensor 图象翻转属性

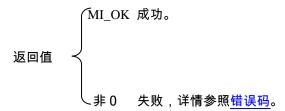
#### ▶ 语法

MI\_S32 MI\_SNR\_SetOrien(<u>MI\_SNR\_PAD\_ID\_e</u> ePADId, MI\_BOOL bMirror, MI\_BOOL bFlip);

# ▶ 形参

参数名称	描述	输入/输出
ePADId	SENSOR设备号。	输入
	取值范围: [0, MI_SNR_MAX_PAD_NUM)。	
bMirror	使能竖直镜像翻转	输入
bFlip	使能水平镜像翻转	输入

#### ▶ 返回值



# ▶ 依赖

• 头文件: mi\_sensor.h

• 库文件: libmi\_sensor.a

### ※ 注意

无。

#### ▶ 举例

无。

# ▶ 相关主题

# MI SNR GetOrien

# 1.2.13 MI\_SNR\_GetOrien

▶ 功能

获取 sensor 图象翻转属性

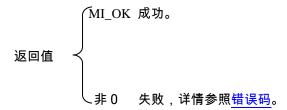
# ▶ 语法

 $\label{eq:mi_snr_pad_id_e} MI\_S32\ MI\_SNR\_GetOrien( \underline{MI\_SNR\_PAD\_ID\_e} \qquad ePADId,\ MI\_BOOL\ *pbMirror, \\ MI\_BOOL\ *pbFlip);$ 

# ▶ 形参

参数名称	描述	输入/输出
ePADId	SENSOR设备号。	输入
	取值范围: [0, MI_SNR_MAX_PAD_NUM)。	
*pbMirror	使能竖直镜像翻转	输出
*pbFlip	使能水平镜像翻转	输出

# ▶ 返回值



#### ▶ 依赖

• 头文件: mi\_sensor.h

• 库文件: libmi\_sensor.a

# ※ 注意

无。

#### ▶ 举例

无。

# ▶ 相关主题

# MI SNR SetOrien

# 1.2.14 MI\_SNR\_SetPlaneMode

▶ 功能

设置 sensor Plane 模式。

▶ 语法

MI\_S32 MI\_SNR\_SetPlaneMode(MI\_SNR\_PAD\_ID\_e ePADId, MI\_BOOL bEnable);

#### ▶ 形参

参数名称	描述	输入/输出
ePADId	SENSOR设备号。	输入
	取值范围: [0, MI SNR MAX PAD NUM)。	
bEnable	开HDR 需要置为TRUE, 否则为FALSE	输入

# ▶ 返回值

#### ▶ 依赖

头文件: mi\_sensor.h

• 库文件: libmi\_sensor.a

#### ※ 注意

无。

▶ 举例

无。

▶ 相关主题

MI SNR GetPlaneMode

# 1.2.15 MI\_SNR\_GetPlaneMode

▶ 功能

获取上层设置的 Sensor Plane 模式。

▶ 语法

MI\_S32 MI\_SNR\_GetPlaneMode(MI\_SNR\_PAD\_ID\_e ePADId, MI\_BOOL \*pbEnable);

# ▶ 形参

参数名称	描述	输入/输出
------	----	-------

ePADId	SENSOR设备号。	输入
	取值范围: [0, MI_SNR_MAX_PAD_NUM)。	
*pbEnable	开HDR 需要置为TRUE, 否则为FALSE	输出

#### ▶ 返回值

#### ▶ 依赖

头文件: mi\_sensor.h

• 库文件: libmi\_sensor.a

#### ※ 注意

无。

▶ 举例

无。

### ▶ 相关主题

MI SNR SetPlaneMode

# 1.2.16 MI\_SNR\_CustFunction

▶ 功能

设置 Sensor 客制化功能。

# ▶ 语法

MI\_S32 MI\_SNR\_CustFunction(<u>MI\_SNR\_PAD\_ID\_e</u> ePADId, MI\_U32 u32CmdId, MI\_U32 u32DataSize, void \*pCustData, MI\_SNR\_CUST\_DIR\_e eDir);;

#### ▶ 形参

参数名称	描述	输入/输出
ePADId	SENSOR设备号。	输入

	取值范围: [0, MI_SNR_MAX_PAD_NUM)。	
u32CmdId	客制化功能ID	输入
u32DataSize	客制化功能数据 buffer size	输入
pCustData	客制化功能数据 buffer	输入
eDir	客制化数据类型	输入

#### ▶ 返回值

返回值 非 0 失败,详情参照错误码。

### ▶ 依赖

• 头文件: mi\_sensor.h

• 库文件: libmi\_sensor.a

### ※ 注意

无。

▶ 举例

无。

▶ 相关主题

MI SNR CUST DIR e

# 2. SENSOR 数据类型

# 视频输入相关数据类型定义如下:

MI SNR MAX PADNUM	定义支持Sensor最大数量。
MI SNR MAX PLANENUM	定义每一个Sensor Pad 支持通道数量
MI SNR PAD ID e	定义Sensor Pad枚举类型
MI_SNR_HDRSrc_e	定义Sensor HDR 通道序号
MI SNR HDRHWMode e	定义 HDR 硬件设置模式
MI SNR Anadec SrcType e	定义Bt656 sensor 输入源格式
MI SNR Res t	定义 Sensor分辨率属性
MI_SNR_Res_List_t	定义 Sensor分辨率映射表
MI_SNR_AttrParallel_t	定义 Parallel Sensor属性
MI_SNR_MipiAttr_t	定义Mipi Sensor属性
MI_SNR_AttrBt656_t	定义Bt656 Sensor属性
MI SNR IntfAttr u	定义Sensor接口联合体
MI SNR PADInfo t	定义Sensor Pad信息
MI_SNR_PlaneInfo_t	定义Sensor 通道信息
MI SNR CUST DIR e	定义Sensor 客制化功能数据类型

# 2.1. MI\_SNR\_MAX\_PADNUM

▶ 说明

定义支持 Sensor 最大数量。

▶ 定义

#define MI\_SNR\_MAX\_PADNUM 4

※ 注意事项

无

▶ 相关数据类型及接口

无。

# 2.2. MI\_SNR\_MAX\_PLANENUM

▶ 说明

定义每一个 Sensor Pad 支持通道数量。

▶ 定义

#define MI\_SNR\_MAX\_PLANENUM 2

※ 注意事项

无。

▶ 相关数据类型及接口

无。

# 2.3. MI\_SNR\_PAD\_ID\_e

▶ 说明

Sensor Pad Id 枚举。

```
typedef enum
             {
                 E_MI_SNR_PAD_ID_0 = 0,
                 E_MI_SNR_PAD_ID_1 = 1,
                 E_MI_SNR_PAD_ID_2 = 2,
                 E_MI_SNR_PAD_ID_3 = 3,
                 E_MI_SNR_PAD_ID_MAX = 3,
                 E_MI_SNR_PAD_ID_NA = 0xFF,
              } MI_SNR_PAD_ID_e;
※
  注意事项
                无。
   相关数据类型及接口
                无。
2.4. MI_SNR_HDRSrc_e
   说明
                Sensor HDR 通道号枚举。
   定义
                typedef enum
              {
                 E_MI_SNR_HDR_SOURCE_VC0,
                 E_MI_SNR_HDR_SOURCE_VC1,
                 E_MI_SNR_HDR_SOURCE_VC2,
                 E_MI_SNR_HDR_SOURCE_VC3,
                 E_MI_SNR_HDR_SOURCE_MAX
```

} MI\_SNR\_HDRSrc\_e;

定义

无。

# ▶ 相关数据类型及接口

MI\_SNR\_PlaneInfo\_t

# 2.5. MI\_SNR\_HDRHWMode\_e

▶ 说明

定义 Sensor 硬件设置 HDR 模式。

▶ 定义

```
typedef enum

{

E_MI_SNR_HDR_HW_MODE_NONE = 0,

E_MI_SNR_HDR_HW_MODE_SONY_DOL = 1,

E_MI_SNR_HDR_HW_MODE_DCG = 2,

E_MI_SNR_HDR_HW_MODE_EMBEDDED_RAW8 = 3,

E_MI_SNR_HDR_HW_MODE_EMBEDDED_RAW10 = 4,

E_MI_SNR_HDR_HW_MODE_EMBEDDED_RAW12 = 5,

E_MI_SNR_HDR_HW_MODE_EMBEDDED_RAW16 = 6, //Only for OV2718?

} MI_SNR_HDRHWMode_e;
```

#### ▶ 成员

成员名称	描述
E_MI_SNR_HDR_HW_MODE_NONE	没有开HDR模式
E_MI_SNR_HDR_HW_MODE_SONY_DOL	Digital Overlap High Dynamic Range
E_MI_SNR_HDR_HW_MODE_DCG	
E_MI_SNR_HDR_HW_MODE_EMBEDDED _RAW8	8bit 压缩模式
E_MI_SNR_HDR_HW_MODE_EMBEDDED _RAW10	10bit 压缩模式
E_MI_SNR_HDR_HW_MODE_EMBEDDED _RAW12	12bit 压缩模式
E_MI_SNR_HDR_HW_MODE_EMBEDDED _RAW16	16bit 压缩模式

※ 注意事项

无。

▶ 相关数据类型及接口

MI SNR MipiAttr t

# 2.6. MI\_SNR\_Anadec\_SrcType\_e

▶ 说明

定义BT656 sensor 输入源格式。

▶ 定义

```
typedef enum

{

E_MI_SNR_ANADEC_SRC_NO_READY = 0,

E_MI_SNR_ANADEC_SRC_PAL,

E_MI_SNR_ANADEC_SRC_NTSC,

E_MI_SNR_ANADEC_SRC_HD,

E_MI_SNR_ANADEC_SRC_FHD,

E_MI_SNR_ANADEC_SRC_DISCNT,

E_MI_SNR_ANADEC_SRC_NUM

} MI_SNR_Anadec_SrcType_e;
```

※ 注意事项

无。

▶ 相关数据类型及接口

MI\_SNR\_GetBT656SrcType

# 2.7. MI\_SNR\_Res\_t

▶ 说明

定义 Sensor 分辨率属性

#### ▶ 定义

```
typedef struct MI_SNR_Res_s

{

MI_SYS_WindowRect_t stCropRect;

MI_SYS_WindowSize_t stOutputSize; /**< Sensor actual output size */

MI_U32 u32MaxFps; /**< Max fps in this resolution */

MI_U32 u32MinFps; /**< Min fps in this resolution*/

MI_S8 strResDesc[32];// Need to put "HDR" here if the resolution is for HDR

} __attribute__((packed, aligned(4))) MI_SNR_Res_t;
```

#### ▶ 成员

成员名称	描述
stCropRect	在output size上裁剪的区域
stOutputSize	Sensor 输出大小范围
u32MaxFps	当前分辨率下最大帧率
u32MinFps	当前分辨率下最小帧率
strResDesc	Resolution string

### ※ 注意事项

sensor 实际输出给后端的 size 是 stCropRect 的 width/height。

#### ▶ 相关数据类型及接口

MI\_SNR\_GetRes MI\_SNR\_GetCurRes MI\_SNR\_SetRes

# 2.8. MI\_SNR\_Res\_List\_t

# ▶ 说明

定义 Sensor 分辨率映射表。

▶ 定义

```
typedef struct MI_SNR_Res_List_s
{
    MI_U32 u32NumRes;/**< number of sensor resolution in list */
    MI_U32 u32CurResIndex;/**< current sensor resolution*/
    MI_SNR_Res_t stRes[12]; /**< resolution list */
} __attribute__((packed, aligned(4))) MI_SNR_Res_List_t;</pre>
```

▶ 成员

成员名称	描述	
u32NumRes	支持的resolution 数量	
u32CurResIndex	当前resolution 索引	
stRes	Resolution 映射表	

#### ※ 注意事项

此结构体为了之后扩展用,目前没有用到

▶ 相关数据类型及接口

MI\_SNR\_Res\_t

# 2.9. MI\_SNR\_AttrParallel\_t

▶ 说明

定义parallel sensor 属性。

▶ 定义

```
typedef struct MI_SNR_AttrParallel_s
{
     MI_VIF_SyncAttr_t stSyncAttr;
} MI_SNR_AttrParallel_t;
```

▶ 成员

成员名称	描述

# **MI SENSOR API**

Version 2.05

stSyncAttr	同步信号属性
------------	--------

#### 注意事项 Ж

无。

#### 相关数据类型及接口

MI SNR IntfAttr u

#### MI SNR MipiAttr t 2.10.

{

说明

定义Mipi sensor 属性。

}MI\_SNR\_MipiAttr\_t;

定义

```
typedef struct MI_SNR_MipiAttr_s
   MI_U32 u32LaneNum; // 几条信号同时传
   MI_U32 u32DataFormat;
                               //0: YUV 422 format. 1: RGB pattern.
   MI_VIF_DataYuvSeq_e
                           eDataYUVOrder;
   MI_U32 u32HsyncMode; //每条 line 的前面 hsync 还是后一般后
   MI_U32 u32Sampling_delay;
   /** < MIPI start sampling delay */ /*bit 0~7: clk_skip_ns. bit 8~15: data_skip_ns*/
   MI_SNR_HDRHWMode_e eHdrHWmode;
   MI_U32 u32Hdr_Virchn_num;
   MI_U32 u32Long_packet_type[2];
// [0]Null [1]blinking [2]embedded [14]yuv422_8b [26]RAW8 [27]RAW10 [28]RAW12 [32]UD1
   [33]UD2 [34]UD3 [35]UD4 [36]UD5 [37]UD6 [38]UD7 [39]UD8
```

#### 成员

成员名称	描述	
u32LaneNum	支持同时传输数据的信号线数量	
u32DataFormat	0: YUV 422 format. 1: RGB pattern	
eDataYUVOrder	YUV 排列顺序	
u32HsyncMode	同步前一条或者后一条line的hsync 信号	

成员名称	描述	
u32Sampling_delay	延时跳过数据头部分	
eHdrHWmode	Sensor支持的HDR mode	
u32Hdr_Virchn_num	Sensor支持的HDR 虚拟通道数量	
u32Long_packet_type[2]	Sensor支持的数据打包格式	

#### ※ 注意事项

无。

# ▶ 相关数据类型及接口

MI\_SNR\_IntfAttr\_u

# 2.11. MI\_SNR\_AttrBt656\_t

▶ 说明

定义BT656 sensor 属性。

▶ 定义

```
typedef struct MI_SNR_AttrBt656_s

{

MI_U32 u32Multiplex_num;

MI_VIF_SyncAttr_t stSyncAttr;

MI_VIF_ClkEdge_e eClkEdge;

MI_VIF_BitOrder_e eBitSwap;

} MI_SNR_AttrBt656_t;
```

# ▶ 成员

成员名称	描述
u32Multiplex_num	复合模式的路数
stSyncAttr	同步信号属性
eClkEdge	采样时钟模式
eBitSwap	数据排列方向

#### ※ 注意事项

无。

▶ 相关数据类型及接口

MI SNR IntfAttr u

# 2.12. MI\_SNR\_IntfAttr\_u

▶ 说明

定义 sensor 接口类型联合体。

▶ 定义

```
typedef union {
    MI_SNR_AttrParallel_t stParallelAttr;
    MI_SNR_MipiAttr_t stMipiAttr;
    MI_SNR_AttrBt656_t stBt656Attr;
} MI_SNR_IntfAttr_u;
```

#### ▶ 成员

成员名称	描述	
stParallelAttr	Parallel sensor 属性	
stMipiAttr	Mipi sensor属性	
stBt656Attr	Bt656 sensor属性	

#### ※ 注意事项

无。

▶ 相关数据类型及接口

MI\_SNR\_PADInfo\_t

# 2.13. MI\_SNR\_PADInfo\_t

▶ 说明

### **MI SENSOR API** Version 2.05

定义 sensor Pad 信息属性。

#### ▶ 定义

```
typedef struct MI_SNR_PADInfo_s

{

MI_U32     u32PlaneCount;

//It is different expo number for HDR. It is mux number for BT656. //??

MI_VIF_IntfMode_e    eIntfMode;

MI_VIF_HDRType_e    eHDRMode;

MI_SNR_IntfAttr_u    unIntfAttr;

MI_BOOL     bEarlyInit;

} MI_SNR_PADInfo_t;
```

#### ▶ 成员

成员名称	描述
u32PlaneCount	BT656 sensor 代表最大复合路数 Mipi sensor 代表长短曝数量
eIntfMode	Sensor 接口枚举
eHDRMode	Hdr 模式
unIntfAttr	Sensor 接口属性联合体
bEarlyInit	Sensor 是否已经提前初始化

### ※ 注意事项

无。

#### ▶ 相关数据类型及接口

MI SNR GetPadInfo

# 2.14. MI\_SNR\_PlaneInfo\_t

▶ 说明

定义 sensor 通道 信息属性。

#### 定义

{

```
typedef struct MI_SNR_PlaneInfo_s
    MI_U32
                              u32PlaneID;// For HDR long/short exposure or BT656 channel 0\sim3
    MI_S8
                              s8SensorName[32];
                               stCapRect;
    MI\_SYS\_WindowRect\_t
    MI_SYS_BayerId_e
                              eBayerId;
    MI_SYS_DataPrecision_e ePixPrecision;
                                eHdrSrc;
    MI_SNR_HDRSrc_e
    MI_U32
                              u32ShutterUs;
                              u32SensorGainX1024;
    MI_U32
    MI_U32
                              u32CompGain;
} MI_SNR_PlaneInfo_t;
```

#### 成员

成员名称	描述	
u32PlaneID	开HDR代表当前Plane是长曝还是短曝	
	BT656 代表代表当前Plane在复合路数中的通道 id	
s8SensorName	Sensor name字符串	
stCapRect	在sensor 数据上裁剪的位置	
eBayerId	RGB 排列顺序	
ePixPrecision	RGB 压缩模式	
eHdrSrc	HDR 通道号	
u32ShutterUs	Sensor Shutter	
u32SensorGainX1024	Sensor Gain	
u32CompGain	Sensor Compensate Gain	

#### Ж. 注意事项

无。

# 相关数据类型及接口

MI\_SNR\_GetPadInfo

# 2.15. MI\_SNR\_CUST\_DIR\_e

▶ 说明

定义 Sensor 客制化功能数据类型。

▶ 定义

```
typedef enum
{
    E_MI_SNR_CUSTDATA_TO_DRIVER,
    E_MI_SNR_CUSTDATA_TO_USER,
    E_MI_SNR_CUSTDATA_MAX = E_MI_SNR_CUSTDATA_TO_USER,
} MI_SNR_CUST_DIR_e;
```

#### ▶ 成员

成员名称	描述	
E_MI_SNR_CUSTDATA_TO_DRIVER	客制化buffer 数据设置给Sensor Driver	
E_MI_SNR_CUSTDATA_TO_USER	客制化buffer 数据从sensor 获取	
E_MI_SNR_CUSTDATA_MAX	数据类型MAX 选项	

# ※ 注意事项

无。

#### ▶ 相关数据类型及接口

MI\_SNR\_CustFunction

# 3. SENSOR 错误码

视频输入 API 错误码如表 所示。

# Sensor API 错误码

错误代码	宏定义	描述
0xA0032001	MI_ERR_SNR_INVALID_DEVID	设备号无效
0xA0032002	MI_ERR_SNR_INVALID_CHNID	通道号无效
0xA0032003	MI_ERR_SNR_INVALID_PARA	参数设置无效
0xA0032006	MI_ERR_SNR_INVALID_NULL_PTR	输入参数空指针错误
0xA0032007	MI_ERR_SNR_FAILED_NOTCONFIG	设备或通道属性未配置
0xA0108008	MI_ERR_SNR_NOT_SUPPORT	操作不支持
0xA0108009	MI_ERR_SNR_NOT_PERM	操作不允许
0xA0108010	MI_ERR_SNR_SYS_NOTREADY	系统未初始化
0xA0108012	MI_ERR_SNR_BUSY	系统忙
0xA0032080	MI_ERR_SNR_FAIL	端口无效