

MI PANEL API

Version 2.08

© 2020 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: MI_PANEL_SetSscConfig 	06/06/2019
2.05	<ul style="list-style-type: none"> Added new mipi dsi config parameters: MI_U8 u8SyncCalibrate MI_U16 u16VirHsyncSt MI_U16 u16VirHsyncEnd MI_U16 u16VsyncRef MI_U16 u16DataClkSkew 	08/09/2019
2.06	<ul style="list-style-type: none"> Added new mipi dsi config parameters: MI_U8 u8PolCh0 MI_U8 u8PolCh1 MI_U8 u8PolCh2 MI_U8 u8PolCh3 MI_U8 u8PolCh4 	10/14/2019
2.07	<ul style="list-style-type: none"> Added output pixel format enum E_MI_PNL_OUTPUT_565BIT_MODE 	12/09/2019
2.08	<ul style="list-style-type: none"> Redefined SSC parameter MI_PANEL_SscConfig_t 	03/13/2020

TABLE OF CONTENTS

REVISION HISTORY	i
TABLE OF CONTENTS.....	ii
1. API 参考	1
1.1. MI_PANEL_Init	2
1.2. MI_PANEL_GPIO_Init.....	2
1.3. MI_PANEL_DeInit	3
1.4. MI_PANEL_SetIndex	3
1.5. MI_PANEL_GetTotalNum	4
1.6. MI_PANEL_SetPowerOn	5
1.7. MI_PANEL_GetPowerOn	5
1.8. MI_PANEL_SetBacklight	6
1.9. MI_PANEL_GetBacklight	7
1.10. MI_PANEL_SetBacklightLevel.....	7
1.11. MI_PANEL_GetBacklightLevel	8
1.12. MI_PANEL_SetSscConfig	9
1.13. MI_PANEL_SetMipiDsiConfig.....	9
1.14. MI_PANEL_SetTimingConfig	10
1.15. MI_PANEL_SetDrvCurrentConfig	11
1.16. MI_PANEL_SetOutputPattern.....	11
1.17. MI_PANEL_SetPanelParam	12
1.18. MI_PANEL_GetPanelParam	13
1.19. MI_PANEL_GetTimingConfig.....	13
2. PANEL 数据类型	15
2.1. MI_PANEL_INDEX_e	16
2.2. MI_PANEL_GpioConfig_t	16
2.3. MI_PANEL_SpiCmdTable_t	16
2.4. MI_PANEL_PowerConfig_t	17
2.5. MI_PANEL_BackLightConfig_t.....	17
2.6. MI_PANEL_SscConfig_t	18
2.7. MI_PANEL_MipiDsiConfig_t	18
2.8. MI_PANEL_MipiDsiLaneMode_e	20
2.9. MI_PANEL_MipiDsiFormat_e.....	20
2.10. MI_PANEL_MipiDsiCtrlMode_e	21
2.11. MI_PANEL_TimingConfig_t.....	21
2.12. MI_PANEL_DrvCurrentConfig_t.....	22
2.13. MI_PANEL_TestPatternConfig_t	22
2.14. MI_PANEL_ParamConfig_t.....	22
2.15. MI_PANEL_LinkType_e	25
2.16. MI_PANEL_AspectRatio_e	26
2.17. MI_PANEL_TiBitMode_e	26
2.18. MI_PANEL_OutputFormatBitMode_e	27
2.19. MI_PANEL_OutputTimingMode_e.....	27
3. PANEL 错误码	29

1. API 参考

该功能模块提供以下 API：

API 名	功能
MI_PANEL_Init	初始化 PANEL
MI_PANEL_GPIO_Init	PANEL 背光、复位、Init Cmd 相关 GPIO 初始化
MI_PANEL_DeInit	去初始化 PANEL
MI_PANEL_SetIndex	设置 Index 对应屏参
MI_PANEL_GetTotalNum	获取总的屏参个数
MI_PANEL_SetPowerOn	设置屏供电使能
MI_PANEL_GetPowerOn	获取屏供电状态
MI_PANEL_SetBacklight	设置背光使能
MI_PANEL_GetBacklight	获取背光使能状态
MI_PANEL_SetBacklightLevel	设置背光亮度值
MI_PANEL_GetBacklightLevel	获取背光亮度值
MI_PANEL_SetSscConfig	设置展频相关信息
MI_PANEL_SetMipiDsiConfig	设置 MIPI 相关信息
MI_PANEL_SetTimingConfig	设置 Timing 相关信息
MI_PANEL_SetDrvCurrentConfig	设置驱动能力
MI_PANEL_SetOutputPattern	设置 PANEL 输出 pattern
MI_PANEL_SetPanelParam	设置 PANEL INFO
MI_PANEL_GetPanelParam	获取当前 PANEL 配置信息
MI_PANEL_GetTimingConfig	获取 Timing 相关信息

1.1. MI_PANEL_Init

➤ 功能

初始化 PANEL 模块。

➤ 语法

MI_S32 MI_PANEL_Init([MI_PANEL_LinkType_e](#) eLinkType);

➤ 形参

参数名称	描述	输入/输出
eLinkType	PANEL 类型。	输入

➤ 返回值

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

➤ 依赖

头文件: mi_panel.h
库文件: libmi_panel.a

➤ 举例

NA

➤ 相关主题

[MI_PANEL_DeInit](#)

1.2. MI_PANEL_GPIO_Init

➤ 功能

初始化 PANEL 复位、背光、command init 相关的 gpio。

➤ 功能

MI_S32 MI_PANEL_GPIO_Init([MI_PANEL_GpioConfig_t](#) *pstGpioCfg);

➤ 形参

参数名称	描述	输入/输出
pstGpioCfg	GPIO 配置信息结构体指针。	输入

➤ 返回值

返回值 {
0 GPIO 初始化成功。
非 0 GPIO 初始化失败，详情参照[错误码](#)。

- 依赖
头文件: `mi_panel.h`、`mi_panel_datatype.h`
库文件: `libmi_panel.a`
- 举例
NA
- 相关主题
NA

1.3. MI_PANEL_DeInit

- 功能
去初始化 PANEL。
- 语法
`MI_S32 MI_PANEL_DeInit(void);`
- 形参
无。
- 返回值
返回值 {
0 去初始化成功。
非 0 去初始化失败, 详情参照[错误码](#)。
- 依赖
头文件: `mi_panel.h`
库文件: `libmi_panel.a`
- 举例
NA
- 相关主题
[MI_PANEL_Init](#)

1.4. MI_PANEL_SetIndex

- 功能
设置屏参。
- 语法
`MI_S32 MI_PANEL_SetIndex(MI_PANEL_INDEX_e eIndex)`

➤ 形参

参数名称	描述	输入/输出
eIndex	屏参索引值。	输入

➤ 返回值

返回值 {
0 设置屏参成功。
非 0 设置屏参失败，详情参照[错误码](#)。

➤ 依赖

头文件：mi_panel.h
库文件：libmi_panel.a

※ 注意

暂不支持。

➤ 举例

NA

➤ 相关主题

[MI_PANEL_GetTotalNum](#)

1.5. MI_PANEL_GetTotalNum

➤ 功能

获取总的屏参个数。

➤ 语法

```
MI_S32 MI_PANEL_GetTotalNum(MI_U32 *pu32TotalNum);
```

➤ 形参

参数名称	描述	输入/输出
pu32TotalNum	该指针接收屏参个数	输出

➤ 返回值

返回值 {
0 获取屏参个数成功。
非 0 获取屏参个数失败，详情参照[错误码](#)。

➤ 依赖

头文件：mi_panel.h
库文件：libmi_panel.a

※ 注意
暂不支持。

➤ 举例
NA

➤ 相关主题
[MI_PANEL_SetIndex](#)

1.6. MI_PANEL_SetPowerOn

➤ 功能
设置屏供电使能。

➤ 语法
MI_S32 MI_PANEL_SetPowerOn([MI_PANEL_PowerConfig_t](#) *pstPowerCfg);

➤ 形参

参数名称	描述	输入/输出
pstPowerCfg	屏供电使能状态	输入

➤ 返回值

返回值 {
 0 设置屏供电使能成功。
 非 0 设置屏供电使能失败，详情参照[错误码](#)。

➤ 依赖
头文件: mi_panel.h、mi_panel_datatype.h
库文件: libmi_panel.a

※ 注意
设置屏供电使能之前需要初始化 PANEL。

➤ 举例
NA

➤ 相关主题
[MI_PANEL_GetPowerOn](#)

1.7. MI_PANEL_GetPowerOn

➤ 功能
获取屏供电使能状态。

➤ 语法
MI_S32 MI_PANEL_GetPowerOn([MI_PANEL_PowerConfig_t](#) *pstPowerCfg);

➤ 语法

参数名称	描述	输入/输出
pstPowerCfg	该指针接收屏供电使能状态。	输出

➤ 返回值

返回值 { 0 获取屏供电使能状态成功。
非 0 获取屏供电使能状态失败，详情参照[错误码](#)。

➤ 依赖

头文件：mi_panel.h、mi_panel_datatype.h
库文件：libmi_panel.a

➤ 举例

NA

➤ 相关主题

[MI_PANEL_SetPowerOn](#)

1.8. MI_PANEL_SetBacklight

➤ 功能

设置屏背光使能。

➤ 语法

MI_S32 MI_PANEL_SetBacklight([MI_PANEL_BackLightConfig_t](#) *pstBackLightCfg);

➤ 形参

参数名称	描述	输入/输出
pstBackLightCfg	屏背光使能状态。	输入

➤ 返回值

返回值 { 0 设置屏背光使能成功。
非 0 设置屏背光使能失败，详情参照[错误码](#)。

➤ 依赖

头文件：mi_panel.h、mi_panel_datatype.h
库文件：libmi_panel.a

➤ 举例

NA

➤ 相关主题

[MI_PANEL_GetBackLight](#)

1.9. MI_PANEL_GetBacklight

➤ 功能

获取屏背光使能状态。

➤ 语法

MI_S32 MI_PANEL_GetBacklight([MI_PANEL_BackLightConfig_t](#) *pstBackLightCfg);

➤ 形参

参数名称	描述	输入/输出
pstBackLightCfg	该指针接收屏背光使能状态。	输出

➤ 返回值

返回值 { 0 获取屏背光使能状态成功。
非 0 获取屏背光使能状态失败，详情参照[错误码](#)。

➤ 依赖

头文件：mi_panel.h、mi_panel_datatype.h
库文件：libmi_panel.a

※ 注意

无。

➤ 举例

NA

➤ 相关主题

[MI_PANEL_SetBackLight](#)

1.10. MI_PANEL_SetBacklightLevel

➤ 功能

设置背光亮度值。

➤ 语法

MI_S32 MI_PANEL_SetBacklightLevel([MI_PANEL_BackLightConfig_t](#) *pstBackLightCfg);

➤ 形参

参数名称	描述	输入/输出
pstBackLightCfg	背光亮度值	输入

- 返回值
 - 返回值 {
 - 0 设置屏背光亮度值成功。
 - 非 0 设置屏背光亮度值失败，详情参照[错误码](#)。
- 依赖
 - 头文件：mi_panel.h、mi_panel_datatype.h
 - 库文件：libmi_panel.a
- ※ 注意
 - 调用前需先设置屏背光使能
- 举例
 - NA
- 相关主题
 - [MI_PANEL_GetBacklightLevel](#)。

1.11. MI_PANEL_GetBacklightLevel

- 功能
 - 获取屏背光亮度值。
- 定义
 - MI_S32 MI_PANEL_GetBacklightLevel([MI_PANEL_BackLightConfig_t](#) *pstBackLightCfg);
- 形参

参数名称	描述	输入/输出
pstBackLightCfg	该指针接收屏背光亮度值。	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_panel.h、mi_panel_datatype.h
 - 库文件：libmi_panel.a
- ※ 注意
 - 调用前需要先设置背光亮度值。
- 举例
 - NA

➤ 相关主题

[MI_PANEL_SetBacklightLevel](#)。

1.12. MI_PANEL_SetSscConfig

➤ 功能

设置展频相关信息。

➤ 定义

MI_S32 MI_PANEL_SetSscConfig([MI_PANEL_SscConfig_t](#) *pstSscCfg);

➤ 形参

参数名称	描述	输入/输出
pstSscCfg	展频相关配置信息	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_panel.h 、 mi_panel_datatype.h
库文件：libmi_panel.a

➤ 注意

无

➤ 举例

NA

➤ 相关主题

无

1.13. MI_PANEL_SetMipiDsiConfig

➤ 功能

设置 MIPI 接口相关配置

➤ 定义

MI_S32 MI_PANEL_SetMipiDsiConfig([MI_PANEL_MipiDsiConfig_t](#) *pstMipiDsiCfg);

➤ 形参

参数名称	描述	输入/输出
pstMipiDsiCfg	MIPI 接口相关配置	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_panel.h 、 mi_panel_datatype.h
库文件：libmi_panel.a
- 注意

只有 mipi panel 才需要调用，同时要调用 [MI_PANEL_SetPanelParam](#)。
- 举例

NA
- 相关主题

无

1.14. MI_PANEL_SetTimingConfig

- 功能

设置 Timing 相关信息
- 定义

MI_S32 MI_PANEL_SetTimingConfig([MI_PANEL_TimingConfig_t](#) *pstTimingCfg);
- 形参

参数名称	描述	输入/输出
pstTimingCfg	Timing 相关配置信息	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_panel.h 、 mi_panel_datatype.h
库文件：libmi_panel.a
- 注意

无
- 举例

NA

➤ 相关主题

[MI_PANEL_GetTimingConfig](#)

1.15. MI_PANEL_SetDrvCurrentConfig

➤ 功能

设置驱动能力相关信息

➤ 定义

MI_S32 MI_PANEL_SetDrvCurrentConfig([MI_PANEL_DrvCurrentConfig_t](#) *pstDrvCurrentCfg);

➤ 形参

参数名称	描述	输入/输出
pstDrvCurrentCfg	驱动能力相关配置信息	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_panel.h 、 mi_panel_datatype.h
库文件: libmi_panel.a

➤ 注意

无

➤ 举例

NA

➤ 相关主题

无

1.16. MI_PANEL_SetOutputPattern

➤ 功能

设置输出 pattern

➤ 定义

MI_S32 MI_PANEL_SetOutputPattern([MI_PANEL_TestPatternConfig_t](#) *pstTestPatternCfg);

➤ 形参

参数名称	描述	输入/输出
pstTestPatternCfg	输出 pattern 相关配置信息	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_panel.h 、 mi_panel_datatype.h
库文件：libmi_panel.a
- 注意

调试使用（目前只针对 LVDS）
- 举例

NA
- 相关主题

无

1.17. MI_PANEL_SetPanelParam

- 功能

设置 PANEL INFO
- 定义

MI_S32 MI_PANEL_SetPanelParam([MI_PANEL_ParamConfig_t](#) *pstParamCfg);
- 形参

参数名称	描述	输入/输出
pstParamCfg	屏参信息	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_panel.h 、 mi_panel_datatype.h
库文件：libmi_panel.a
- 注意

无
- 举例

NA

➤ 相关主题

[MI_PANEL_GetPanelParam](#)

1.18. MI_PANEL_GetPanelParam

➤ 功能

获取屏参

➤ 定义

MI_S32 MI_PANEL_GetPanelParam([MI_PANEL_ParamConfig_t](#) *pstParamCfg);

➤ 形参

参数名称	描述	输入/输出
pstParamCfg	屏参信息	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_panel.h 、 mi_panel_datatype.h
库文件: libmi_panel.a

➤ 注意

需先调用 [MI_PANEL_SetPanelParam](#)。

➤ 举例

NA

➤ 相关主题

[MI_PANEL_SetPanelParam](#)

1.19. MI_PANEL_GetTimingConfig

➤ 功能

获取 Timing 相关信息

➤ 定义

MI_S32 MI_PANEL_GetTimingConfig([MI_PANEL_TimingConfig_t](#) *pstTimingCfg);

➤ 形参

参数名称	描述	输入/输出
pstTimingCfg	Timing 相关信息	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_panel.h 、 mi_panel_datatype.h
 - 库文件：libmi_panel.a
- 注意
 - 无
- 举例
 - NA
- 相关主题
 - [MI_PANEL_SetTimingConfig](#)

2. PANEL 数据类型

PANEL 相关数据类型定义如下：

MI_PANEL_INDEX_e	定义 PANEL INFO INDEX 枚举
MI_PANEL_GpioConfig_t	定义 GPIO 配置信息结构体
MI_PANEL_SpiCmdTable_t	定义 PANEL initcmd table 结构体指针
MI_PANEL_PowerConfig_t	定义 PANEL 供电使能状态结构体
MI_PANEL_BackLightConfig_t	定义 PANEL 背光信息结构体
MI_PANEL_SscConfig_t	定义 PANEL 展频配置信息结构体
MI_PANEL_MipiDsiConfig_t	定义 MIPI 接口配置信息结构体
MI_PANEL_MipiDsiLaneMode_e	定义 MIPI 接口 Lane mode 枚举
MI_PANEL_MipiDsiFormat_e	定义 MIPI 接口 Format 枚举
MI_PANEL_MipiDsiCtrlMode_e	定义 MIPI 接口 CtrlMode 枚举
MI_PANEL_TimingConfig_t	定义 PANEL Timing 信息结构体
MI_PANEL_DrvCurrentConfig_t	定义驱动能力配置结构体
MI_PANEL_TestPatternConfig_t	定义输出 pattern 信息结构体
MI_PANEL_ParamConfig_t	定义屏参信息结构体
MI_PANEL_LinkType_e	定义 Link 类型枚举
MI_PANEL_AspectRatio_e	定义 AspectRatio 枚举
MI_PANEL_TiBitMode_e	定义 TiBitMode 枚举
MI_PANEL_OutputFormatBitMode_e	定义输出 FormatBitMode 枚举
MI_PANEL_OutputTimingMode_e	定义 panel output timing change mode 枚举

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_panel_datatype.h`

2.1. MI_PANEL_INDEX_e

➤ 说明

定义 PANEL INFO INDEX 枚举。

➤ 定义

```
typedef enum
{
    MI_PANEL_AUO_21_1920x1080_60Hz = 0,
    MI_PANEL_LG_37_1920x1080_60Hz,
    MI_PANEL_INX_10_1280x800_60Hz,
    MI_PANEL_MAX
} MI_PANEL_INDEX_e;
```

➤ 相关数据类型及接口

[MI_PANEL_SetIndex](#)

2.2. MI_PANEL_GpioConfig_t

➤ 说明

定义 GPIO 配置信息结构体

➤ 定义

```
typedef struct
{
    MI_BOOL bCtlBL;    \\是否控制背光
    MI_BOOL bInvBL;    \\背光极性是否反转
    MI_U16 u16GpioBL;  \\背光 GPIO NUM
    MI_U16 u16GpioRST; \\复位 GPIO NUM
    MI_U16 u16GpioSCL; \\模拟 SPI SCL 信号 GPIO NUM
    MI_U16 u16GpioSDI; \\模拟 SPI SDI 信号 GPIO NUM
    MI_U16 u16GpioCS;  \\模拟 SPI CS 信号 GPIO NUM
    MI_U32 u32CmdTblSize; \\initcmd table size
    MI_U8 u8PerCmdLength; \\每个 cmd 的字节数
    MI_PANEL_SpiCmdTable_t *pSpiCmdTbl; \\initcmd table 结构体指针
}MI_PANEL_GpioConfig_t;
```

➤ 相关数据类型及接口

[MI_PANEL_SpiCmdTable_t](#)
[MI_PANEL_GPIO_Init](#)

2.3. MI_PANEL_SpiCmdTable_t

➤ 说明

定义 PANEL initcmd table 结构体类型

➤ 定义

```
typedef struct
{
    unsigned int cmd; \\cmd value
    unsigned char count; \\NA
    unsigned char para_list[64]; \\NA
}MI_PANEL_SpiCmdTable_t;
```

➤ 相关数据类型及接口

[MI_PANEL_GpioConfig_t](#)
[MI_PANEL_GPIO_Init](#)

2.4. MI_PANEL_PowerConfig_t

➤ 说明

定义 PANEL 供电使能状态结构体

➤ 定义

```
typedef struct
{
    MI_BOOL bEn;
}MI_PANEL_PowerConfig_t;
```

➤ 相关数据类型及接口

[MI_PANEL_SetPowerOn](#)
[MI_PANEL_GetPowerOn](#)

2.5. MI_PANEL_BackLightConfig_t

➤ 说明

定义 PANEL 背光信息结构体

➤ 定义

```
typedef struct
{
    MI_BOOL bEn;
    MI_U16 u16Level;
}MI_PANEL_BackLightConfig_t;
```

➤ 成员

成员名称	描述
bEn	是否使能 panel 背光。
u16Level	如果 bEn = TRUE, 该参数设置背光 level。

- 相关数据类型及接口

[MI_PANEL_SetBacklight](#)

[MI_PANEL_GetBacklight](#)

[MI_PANEL_SetBacklightLevel](#)

[MI_PANEL_GetBacklightLevel](#)

2.6. MI_PANEL_SscConfig_t

- 说明

定义 PANEL 展频配置信息结构体

- 定义

```
typedef struct
{
    MI_BOOL bEn;
    MI_U16 u16Freq;
    MI_U16 u16Ratio;
}MI_PANEL_SscConfig_t;
```

- 成员

成员名称	描述
bEn	panel 展频功能使能状态
u16Freq	展频频率（单位 Khz）
u16Ratio	展频幅度（单位 1/1000）

- 注意

u16Freq 不能小于 15Khz.

- 相关数据类型及接口

[MI_PANEL_SetSscConfig](#)

2.7. MI_PANEL_MipiDsiConfig_t

- 说明

定义 MIPI 接口配置信息结构体

- 定义

```
typedef struct
{
    MI_U8 u8HsTrail;
    MI_U8 u8HsPrpr;
    MI_U8 u8HsZero;
    MI_U8 u8ClkHsPrpr;
    MI_U8 u8ClkHsExit;
```

```
MI_U8 u8ClkTrail;
MI_U8 u8ClkZero;
MI_U8 u8ClkHsPost;
MI_U8 u8DaHsExit;
MI_U8 u8ContDet;

MI_U8 u8Lpx;
MI_U8 u8TaGet;
MI_U8 u8TaSure;
MI_U8 u8TaGo;

MI_U16 u16Hactive;
MI_U16 u16Hpw;
MI_U16 u16Hbp;
MI_U16 u16Hfp;
MI_U16 u16Vactive;
MI_U16 u16Vpw;
MI_U16 u16Vbp;
MI_U16 u16Vfp;

MI_U16 u16BlIp;
MI_U16 u16Fps;

MI_PANEL_MipiDsiLaneMode_e enLaneNum;
MI_PANEL_MipiDsiFormat_e enformat;
MI_PANEL_MipiDsiCtrlMode_e enCtrl;

MI_U8 *pu8CmdBuf;
MI_U32 u32CmdBufSize;

MI_U8 u8SyncCalibrate;
MI_U16 u16VirHsyncSt;
MI_U16 u16VirHsyncEnd;
MI_U16 u16VsyncRef;
MI_U16 u16DataClkSkew;

MI_U8 u8PolCh0;
MI_U8 u8PolCh1;
MI_U8 u8PolCh2;
MI_U8 u8PolCh3;
MI_U8 u8PolCh4;
}MI_PANEL_MipiDsiConfig_t;
```

- 相关数据类型及接口

[MI_PANEL_MipiDsiLaneMode_e](#)
[MI_PANEL_MipiDsiFormat_e](#)
[MI_PANEL_MipiDsiCtrlMode_e](#)
[MI_PANEL_SetMipiDsiConfig](#)

2.8. MI_PANEL_MipiDsiLaneMode_e

- 说明

定义 MIPI 接口 Lane mode 枚举

- 定义

```
typedef enum
{
    E_MI_PNL_MIPI_DSI_LANE_NONE    = 0,
    E_MI_PNL_MIPI_DSI_LANE_1      = 1,
    E_MI_PNL_MIPI_DSI_LANE_2      = 2,
    E_MI_PNL_MIPI_DSI_LANE_3      = 3,
    E_MI_PNL_MIPI_DSI_LANE_4      = 4,
} MI_PANEL_MipiDsiLaneMode_e;
```

- 相关数据类型及接口

[MI_PANEL_MipiDsiConfig_t](#)
[MI_PANEL_SetMipiDsiConfig](#)

2.9. MI_PANEL_MipiDsiFormat_e

- 说明

定义 MIPI 接口 Format 枚举

- 定义

```
typedef enum
{
    E_MI_PNL_MIPI_DSI_RGB565      = 0,
    E_MI_PNL_MIPI_DSI_RGB666      = 1,
    E_MI_PNL_MIPI_DSI_LOOSELY_RGB666 = 2,
    E_MI_PNL_MIPI_DSI_RGB888      = 3,
} MI_PANEL_MipiDsiFormat_e;
```

- 相关数据类型及接口

[MI_PANEL_MipiDsiConfig_t](#)
[MI_PANEL_SetMipiDsiConfig](#)

2.10. MI_PANEL_MipiDsiCtrlMode_e

➤ 说明

定义 MIPI 接口 CtrlMode 枚举

➤ 定义

```
typedef enum
{
    E_MI_PNL_MIPI_DSI_CTRL_CMD_MODE = 0,
    E_MI_PNL_MIPI_DSI_CTRL_SYNC_PULSE = 1,
    E_MI_PNL_MIPI_DSI_CTRL_SYNC_EVENT = 2,
    E_MI_PNL_MIPI_DSI_BURST_MODE = 3,
} MI_PANEL_MipiDsiCtrlMode_e;
```

➤ 相关数据类型及接口

[MI_PANEL_MipiDsiConfig_t](#)

[MI_PANEL_SetMipiDsiConfig](#)

2.11. MI_PANEL_TimingConfig_t

➤ 说明

定义 PANEL Timing 信息结构体

➤ 定义

```
typedef struct
{
    MI_U16 u16HSyncWidth;
    MI_U16 u16HSyncBackPorch;
    MI_U16 u16HSyncFrontPorch;

    MI_U16 u16VSyncWidth;
    MI_U16 u16VSyncBackPorch;
    MI_U16 u16VSyncFrontPorch;

    MI_U16 u16HStart;
    MI_U16 u16VStart;

    MI_U16 u16Height;
    MI_U16 u16Width;

    MI_U16 u16Htotal;
    MI_U16 u16Vtotal;
}MI_PANEL_TimingConfig_t;
```

- 相关数据类型及接口

[MI_PANEL_SetTimingConfig](#)

[MI_PANEL_GetTimingConfig](#)

2.12. MI_PANEL_DrvCurrentConfig_t

- 说明

定义驱动能力配置结构体

- 定义

```
typedef struct
{
    MI_U16 u16DrvCurrent;
}MI_PANEL_DrvCurrentConfig_t;
```

- 相关数据类型及接口

[MI_PANEL_SetDrvCurrentConfig](#)

2.13. MI_PANEL_TestPatternConfig_t

- 说明

定义输出 pattern 结构体

- 定义

```
typedef struct
{
    MI_BOOL bEn;
    MI_U16 u16R;
    MI_U16 u16G;
    MI_U16 u16B;
}MI_PANEL_TestPatternConfig_t;
```

- 相关数据类型及接口

[MI_PANEL_SetOutputPattern](#)

2.14. MI_PANEL_ParamConfig_t

- 说明

定义屏参结构体

➤ 定义

```
typedef struct
{
    const char pPanelName[48];          ///< PanelName
    MI_U8      u8Dither;                ///< PANEL_DITHER, keep the setting
    MI_PANEL_LinkType_e eLinkType;      ///< PANEL_LINK

    ////////////////////////////////////////////////////////////////////
    // Board related setting
    ////////////////////////////////////////////////////////////////////
    MI_U8 u8DualPort;                  ///< VOP_21[8], MOD_4A[1], PANEL_DUAL_PORT, refer
to u8DoubleClk
    MI_U8 u8SwapPort;                  ///< MOD_4A[0], PANEL_SWAP_PORT, refer to "LVDS
output app note" A/B channel swap
    MI_U8 u8SwapOdd_ML;                ///< PANEL_SWAP_ODD_ML
    MI_U8 u8SwapEven_ML;               ///< PANEL_SWAP_EVEN_ML
    MI_U8 u8SwapOdd_RB;                ///< PANEL_SWAP_ODD_RB
    MI_U8 u8SwapEven_RB;               ///< PANEL_SWAP_EVEN_RB

    MI_U8 u8SwapLVDS_POL;              ///< MOD_40[5], PANEL_SWAP_LVDS_POL, for
differential P/N swap
    MI_U8 u8SwapLVDS_CH;               ///< MOD_40[6], PANEL_SWAP_LVDS_CH, for pair
swap
    MI_U8 u8PDP10BIT;                  ///< MOD_40[3], PANEL_PDP_10BIT ,for pair swap
    MI_U8 u8LVDS_TI_MODE;              ///< MOD_40[2], PANEL_LVDS_TI_MODE, refer to
"LVDS output app note"

    ////////////////////////////////////////////////////////////////////
    // For TTL Only
    ////////////////////////////////////////////////////////////////////
    MI_U8 u8DCLKDelay;                 ///< PANEL_DCLK_DELAY
    MI_U8 u8InvDCLK;                   ///< MOD_4A[4], PANEL_INV_DCLK
    MI_U8 u8InvDE;                      ///< MOD_4A[2], PANEL_INV_DE
    MI_U8 u8InvHSync;                  ///< MOD_4A[12], PANEL_INV_HSYNC
    MI_U8 u8InvVSync;                  ///< MOD_4A[3], PANEL_INV_VSYNC

    ////////////////////////////////////////////////////////////////////
    // Output driving current setting
    ////////////////////////////////////////////////////////////////////
    // driving current setting (0x00=4mA, 0x01=6mA, 0x02=8mA, 0x03=12mA)
    MI_U8 u8DCKLCurrent;                ///< define PANEL_DCLK_CURRENT
    MI_U8 u8DECurrent;                  ///< define PANEL_DE_CURRENT
    MI_U8 u8ODDDataCurrent;             ///< define PANEL_ODD_DATA_CURRENT
    MI_U8 u8EvenDataCurrent;           ///< define PANEL_EVEN_DATA_CURRENT
}
```

```

////////////////////////////////////
// panel on/off timing
////////////////////////////////////
MI_U16 u16OnTiming1;      ///< time between panel & data while turn on power
MI_U16 u16OnTiming2;      ///< time between data & back light while turn on power
MI_U16 u16OffTiming1;     ///< time between back light & data while turn off power
MI_U16 u16OffTiming2;    ///< time between data & panel while turn off power

////////////////////////////////////
// panel timing spec.
////////////////////////////////////
// sync related
MI_U16 u16HSyncWidth;     ///< VOP_01[7:0], PANEL_HSYNC_WIDTH
MI_U16 u16HSyncBackPorch; ///< PANEL_HSYNC_BACK_PORCH, no register setting,
provide value for query only
MI_U16 u16VSyncWidth;     ///< define PANEL_VSYNC_WIDTH
MI_U16 u16VSyncBackPorch; ///< define PANEL_VSYNC_BACK_PORCH

// DE related
MI_U16 u16HStart;         ///< VOP_04[11:0], PANEL_HSTART, DE H Start
(PANEL_HSYNC_WIDTH + PANEL_HSYNC_BACK_PORCH)
MI_U16 u16VStart;         ///< VOP_06[11:0], PANEL_VSTART, DE V Start
MI_U16 u16Width;          ///< PANEL_WIDTH, DE width (VOP_05[11:0] = HEnd =
HStart + Width - 1)
MI_U16 u16Height;         ///< PANEL_HEIGHT, DE height (VOP_07[11:0], = Vend =
VStart + Height - 1)

// DClk related
MI_U16 u16MaxHTotal;      ///< PANEL_MAX_HTOTAL. Reserved for future using.
MI_U16 u16HTotal;         ///< VOP_0C[11:0], PANEL_HTOTAL
MI_U16 u16MinHTotal;      ///< PANEL_MIN_HTOTAL. Reserved for future using.
MI_U16 u16MaxVTTotal;     ///< PANEL_MAX_VTOTAL. Reserved for future using.
MI_U16 u16VTTotal;        ///< VOP_0D[11:0], PANEL_VTOTAL
MI_U16 u16MinVTTotal;     ///< PANEL_MIN_VTOTAL. Reserved for future using.
MI_U16 u16MaxDCLK;        ///< PANEL_MAX_DCLK. Reserved for future using.
MI_U16 u16DCLK;           ///< LPLL_0F[23:0], PANEL_DCLK
MI_U16 u16MinDCLK;        ///< PANEL_MIN_DCLK. Reserved for future using.

                                 ///< spread spectrum
MI_U16 u16SpreadSpectrumStep; ///< move to board define, no use now.
MI_U16 u16SpreadSpectrumSpan; ///< move to board define, no use now.

MI_U8 u8DimmingCtl;       ///< Initial Dimming Value

```

```

MI_U8 u8MaxPWMVal;          ///< Max Dimming Value
MI_U8 u8MinPWMVal;         ///< Min Dimming Value

MI_U8 u8DeinterMode;       ///< define PANEL_DEINTER_MODE, no use now
MI_PANEL_AspectRatio_e ePanelAspectRatio; ///< Panel Aspect Ratio, provide information to
upper layer application for aspect ratio setting.

MI_U16 u16LVDSTxSwapValue;
MI_PANEL_TiBitMode_e eTiBitMode;          ///< MOD_4B[1:0], refer to
"LVDS output app note"
MI_PANEL_OutputFormatBitMode_e eOutputFormatBitMode;

MI_U8 u8SwapOdd_RG;        ///< define PANEL_SWAP_ODD_RG
MI_U8 u8SwapEven_RG;       ///< define PANEL_SWAP_EVEN_RG
MI_U8 u8SwapOdd_GB;        ///< define PANEL_SWAP_ODD_GB
MI_U8 u8SwapEven_GB;       ///< define PANEL_SWAP_EVEN_GB

/**
 * Others
 */
MI_U8 u8DoubleClk;         ///< LPLL_03[7], define Double Clock ,LVDS dual mode
MI_U32 u32MaxSET;          ///< define PANEL_MAX_SET
MI_U32 u32MinSET;          ///< define PANEL_MIN_SET
MI_PANEL_OutputTimingMode_e eOutTimingMode; ///< Define which panel output timing
change mode is used to change VFreq for same panel
MI_U8 u8NoiseDith;         ///< PAFRC mixed with noise dither disable
}MI_PANEL_ParamConfig_t;

```

➤ 相关数据类型及接口

[MI_PANEL_LinkType_e](#)
[MI_PANEL_AspectRatio_e](#)
[MI_PANEL_TiBitMode_e](#)
[MI_PANEL_OutputFormatBitMode_e](#)
[MI_PANEL_OutputTimingMode_e](#)
[MI_PANEL_GetPanelParam](#)

2.15. MI_PANEL_LinkType_e

➤ 说明

定义 PANEL LinkType 枚举

➤ 定义

```
typedef enum
{
    E_MI_PNL_LINK_TTL,                ///< TTL type
    E_MI_PNL_LINK_LVDS,              ///< LVDS type
    E_MI_PNL_LINK_RSDS,              ///< RSDS type
    E_MI_PNL_LINK_MINILVDS,          ///< TCON
    E_MI_PNL_LINK_ANALOG_MINILVDS,   ///< Analog TCON
    E_MI_PNL_LINK_DIGITAL_MINILVDS,  ///< Digital TCON
    E_MI_PNL_LINK_MFC,               ///< Ursa (TTL output to Ursa)
    E_MI_PNL_LINK_DAC_I,             ///< DAC output
    E_MI_PNL_LINK_DAC_P,             ///< DAC output
    E_MI_PNL_LINK_PDPLVDS,           ///< For PDP(Vsync use Manually MODE)
    E_MI_PNL_LINK_EXT,               ///< EXT LPLL TYPE
    E_MI_PNL_LINK_MIPI_DSI,          ///< Mipi DSI
}MI_PANEL_LinkType_e;
```

➤ 相关数据类型及接口

[MI_PANEL_ParamConfig_t](#)

[MI_PANEL_GetPanelParam](#)

2.16. MI_PANEL_AspectRatio_e

➤ 说明

定义 PANEL AspectRatio 枚举

➤ 定义

```
typedef enum
{
    E_MI_PNL_ASPECT_RATIO_4_3 = 0,    ///< set aspect ratio to 4 : 3
    E_MI_PNL_ASPECT_RATIO_WIDE,       ///< set aspect ratio to 16 : 9
    E_MI_PNL_ASPECT_RATIO_OTHER,      ///< reserved for other aspect ratio other than
    4:3/ 16:9
}MI_PANEL_AspectRatio_e;
```

➤ 相关数据类型及接口

[MI_PANEL_ParamConfig_t](#)

[MI_PANEL_GetPanelParam](#)

2.17. MI_PANEL_TiBitMode_e

➤ 说明

定义 PANEL TiBitMode 枚举

➤ 定义

```
typedef enum
{
    E_MI_PNL_TI_10BIT_MODE = 0,
    E_MI_PNL_TI_8BIT_MODE = 2,
    E_MI_PNL_TI_6BIT_MODE = 3,
} MI_PANEL_TiBitMode_e;
```

➤ 相关数据类型及接口

[MI_PANEL_ParamConfig_t](#)

[MI_PANEL_GetPanelParam](#)

2.18. MI_PANEL_OutputFormatBitMode_e

➤ 说明

定义 PANEL OutputFormatBitMode 枚举

➤ 定义

```
typedef enum
{
    E_MI_PNL_OUTPUT_10BIT_MODE = 0, //default is 10bit, because 8bit panel can use 10bit
    config and 8bit config.
    E_MI_PNL_OUTPUT_6BIT_MODE = 1, //but 10bit panel(like PDP panel) can only use 10bit
    config.
    E_MI_PNL_OUTPUT_8BIT_MODE = 2, //and some PDA panel is 6bit.
    E_MI_PNL_OUTPUT_565BIT_MODE = 3,
} MI_PANEL_OutputFormatBitMode_e;
```

➤ 相关数据类型及接口

[MI_PANEL_ParamConfig_t](#)

[MI_PANEL_GetPanelParam](#)

2.19. MI_PANEL_OutputTimingMode_e

➤ 说明

定义 panel output timing change mode

➤ 定义

```
typedef enum
{
    E_MI_PNL_CHG_DCLK = 0, //<change output DClk to change Vfreq.
    E_MI_PNL_CHG_HTOTAL = 1, //<change H total to change Vfreq.
    E_MI_PNL_CHG_VTOTAL = 2, //<change V total to change Vfreq.
} MI_PANEL_OutputTimingMode_e;
```

➤ 相关数据类型及接口

[MI_PANEL_ParamConfig_t](#)

[MI_PANEL_GetPanelParam](#)

3. PANEL 错误码

错误代码	宏定义	描述
0x0	MI_SUCCESS	succeeded
0xA00B2011	MI_ERR_PANEL_INVALID_PARAM	非法参数
0xA00B2006	MI_ERR_PANEL_NULL_PTR	空指针异常
0xA00B2001	MI_ERR_PANEL_INVALID_DEVID	无效设备 ID
0xA00B2018	MI_ERR_PANEL_DRV_FAILED	PANEL 驱动返回失败
0xA00B2015	MI_ERR_PANEL_NOT_INIT	PANEL 未初始化
0xA00B2008	MI_ERR_PANEL_NOT_SUPPORT	不支持操作