MI HDMI API

Version 2.05

© 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	• Initial release	04/12/2018
2.04	• Added MI_HDMI_SetAnalogDrvCurrent	06/12/2019
2.05	• Revised the Doc and Summary Section	01/12/2020

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

TABLE OF CONTENTS

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

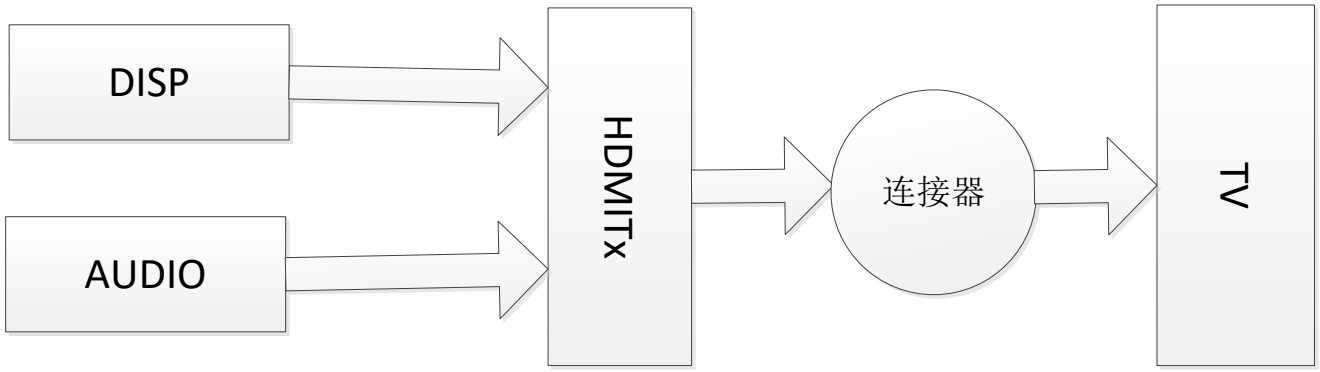
1. 概述

1.1. 模块说明

HDMI (High Definition Multimedia Interface)，即高清多媒体接口，是一种全数字化视频和声音发送接口，可以同时发送未压缩的音频及视频信号。

HDMI 可以分为 TX 端与 RX 端，TX 即 soure 设备，用来将数据传输给 RX（sink 设备）。本模块目前只支持基于 HDMI ver1.4 的 HDMITx。

1.2. 数据流框图



1.3. 各平台特性

平台 / 支持特性	SSR621/SSR623
1920x1080@24	Y
1920x1080@25	Y
1920x1080@30	Y
1920x1080@50	Y
1920x1080@60	Y
1280x720@60	Y
1280x720@50	Y
720x576@50	Y
720x480@60	Y
1280x1024@60	Y
1024x768@60	Y
1440x900@60	Y
1600x1200@60	Y
1280x800@60	Y

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

平台 / 支持特性	SSR621/SSR623
1366x768@60	Y
1680x1050@60	Y
Audio_32K	Y
Audio_48K	Y

1.4. 关键字说明

- EDID：Extended display identification data
  指屏幕分辨率的资料，包括厂商名称与序号。
- InfoFrame:
  HDMI 的规范，属于辅助数据类别，用于告知 sink 设备要传输数据的各种特质。
- DeepColor:
  即深色模式，用于表示更加精准的颜色，数据量也会随之增加。
- SinkInfo:
  sink 端的设备信息。

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

2. API 参考

该功能模块提供以下 API：

API 名	功能
MI_HDMI_Init	初始化 HDMI
MI_HDMI_DeInit	去初始化 HDMI
MI_HDMI_Open	打开 HDMI
MI_HDMI_Close	关闭 HDMI
MI_HDMI_SetAttr	设置 HDMI 属性
MI_HDMI_GetAttr	获取 HDMI 属性
MI_HDMI_Start	启动 HDMI
MI_HDMI_Stop	停止 HDMI
MI_HDMI_GetSinkInfo	获取 HDMI Sink 端的能力信息
MI_HDMI_SetAvMute	设置 HDMI 音视频静音功能
MI_HDMI_FroceGetEdid	获取 HDMI 的 EDID 信息
MI_HDMI_SetDeepColor	设置 HDMI 的 Deep Color 模式
MI_HDMI_GetDeepColor	获取 HDMI 的 Deep Color 模式
MI_HDMI_SetInfoFrame	设置 HDMI 的 Info Frame 信息
MI_HDMI_GetInfoFrame	获取 当前设置的 HDMI 的 Info Frame 信息
MI_HDMI_SetAnalogDrvCurrent	设置 HDMI 的通道驱动电流

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

2.1. MI_HDMI_Init

➢ 功能

初始化 HDMI 模块。

➢ 语法

MI_S32 MI_HDMI_Init(MI_HDMI_InitPara_t *pstHdmiPara);

➢ 形参

参数名称	描述	输入/输出
pstHdmiPara	初始化参数结构体指针， 可以配置为 NULL	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_hdmi.h、mi_hdmi_datatype.h
库文件：libmi_hdmi.a

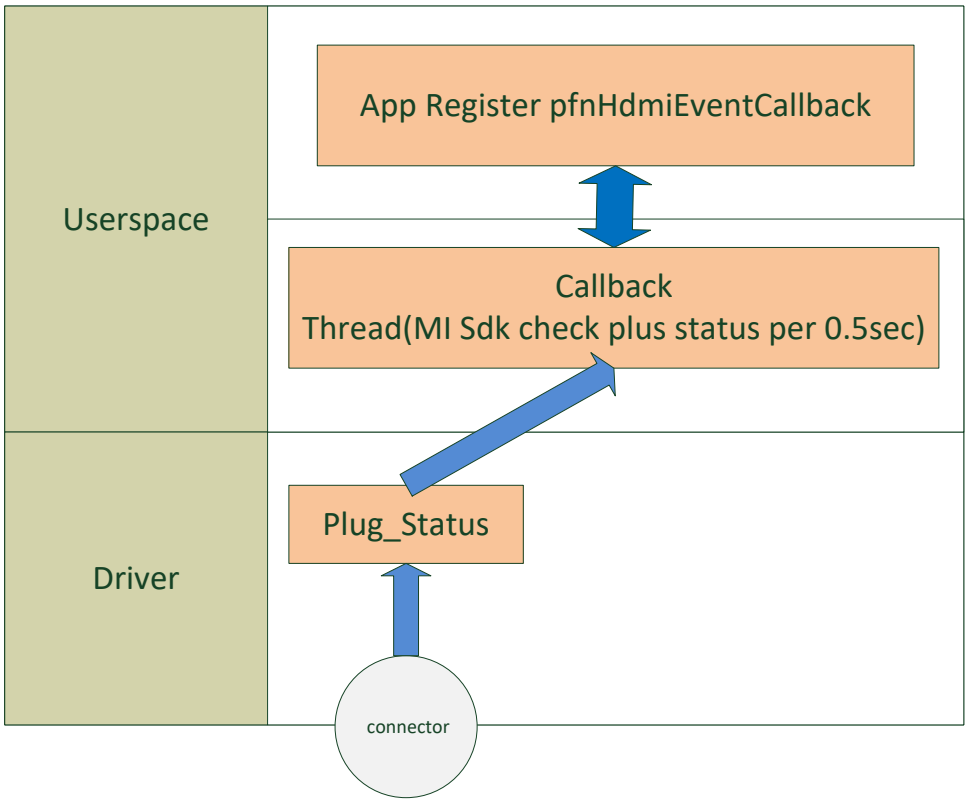
※ 注意

- HDMI 在注册事件回调后，系统会以回调的形式通知 AP 有热插拔之类的操作，如果用户不需要对相应事件做特殊响应，可以将初始化参数结构体中的函数指针置为 NULL。

➢ 特殊说明

在 Init 注册的 pfnHdmiEventCallback 回调，主要是回调插、拔事件给 App，目前设计是在将在 SDK 层开一个线程，然后检查对应的时间来上报给回调注册者

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.



➢ 举例

```
1.  MI_HDMI_InitPara_t stHdmiPara;
2.  MI_HDMI_Attr_t stAttr;
3.  MI_HDMI_TimingType_e eTimingType;
4.  MI_HDMI_Edid_t stEdid;
5.  /* 启动 DISP */
6.  eTimingType = E_MI_HDMI_TIMING_1080_60P;
7.  /* 启动 HDMI */
8.  stHdmiPara.pCallBackArgs = NULL;
9.  stHdmiPara.pfnHdmiEventCallback = NULL;
10. MI_HDMI_Init(&stHdmiPara);
11. MI_HDMI_Open(0);
12. /* 获取 sink 设备支持的 timing */
13. MI_HDMI_ForceGetEdid(0, &stEdid);
14. ...
15. /* 根据 sink 支持的 timing 进行配置 */
16. MI_MPI_HDMI_GetAttr(0, &stAttr);
17. stAttr.bEnableHdmi = MI_TRUE;
18. stAttr.bEnableVideo = MI_TRUE;
19. stAttr.eVideoFmt = enVideoFmt;
20. stAttr.eVidOutMode= E_MI_HDMI_VIDEO_MODE_YCBCR444;
21. stAttr.eDeepColorMode = E_MI_HDMI_DEEP_COLOR_24BIT;
22. stAttr.bEnableAudio = MI_FALSE;
23. stAttr.bIsMultiChannel = MI_FALSE;
24. stAttr.eBitDepth = E_MI_HDMI_BIT_DEPTH_16;
25. MI_HDMI_SetAttr(0, &stAttr);
26. MI_HDMI_Start(0);
27. /* 停止 HDMI */
28. MI_HDMI_Stop(0);
29. MI_HDMI_Close(0);
30. MI_HDMI_DeInit();
```


Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

- 相关主题

  MI _HDMI_DeInit

2.2. MI_HDMI_DeInit

- 功能

  去初始化 HDMI。

- 语法

  MI_S32 MI_HDMI_DeInit();

- 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_hdmi.h、mi_ hdmi_datatype.h
  库文件：libmi_hdmi.a

※ 注意

  无。

- 举例

  请参见 MI_HDMI_Init 的举例。

- 相关主题

  MI_HDMI_Init

2.3. MI_HDMI_Open

- 功能

  打开 HDMI。

- 语法

  MI_S32 MI_HDMI_Open(MI_HDMI_DeviceId_e eHdmi);

- 形参

参数名称	描述	输入/输出
eHdmi	HDMI 接口号。 取值范围：0	输入

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

- 返回值

返回值：
- 0 打开 HDMI 设备成功。
- 非 0 打开设备失败，详情参照错误码。

- 依赖

头文件：mi_hdmi.h、mi_ hdmi_datatype.h
库文件：libmi _hdmi.a

※ 注意

打开 HDMI 之前必须先调用 MI_HDMI_Init。
重复打开 HDMI 返回成功。

- 举例

请参见 MI_HDMI_Init 的举例。

- 相关主题

MI_HDMI_Close

2.4. MI_HDMI_Close

- 功能

关闭 HDMI。

- 语法

```
MI_S32 MI_HDMI_Close(MI_HDMI_DeviceId_e eHdmi);
```

- 形参

参数名称	描述	输入/输出
eHdmi	HDMI 接口号。 取值范围：0	输入

- 返回值

返回值：
- 0 销毁 HDMI 设备成功。
- 非 0 销毁 HDMI 设备失败，详情参照错误码。

- 依赖

头文件：mi_hdmi.h、mi_ hdmi_datatype.h
库文件：libmi_hdmi.a

※ 注意

重复关闭返回成功。

- 举例

请参见 MI_HDMI_Init 的举例。

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

➢ 相关主题

MI_HDMI_Open

2.5. MI_HDMI_SetAttr

➢ 功能

设置 HDMI 属性。

➢ 语法

MI_S32 MI_HDMI_SetAttr(MI_HDMI_DeviceId_e eHdmi, MI_HDMI_Attr_t *pstAttr);

➢ 形参

参数名称	描述	输入/输出
eHdmi	HDMI 接口号。 取值范围：0	输入
pstAttr	HDMI 属性结构体指针。	输入

➢ 返回值

返回值：
- 0 设置 HDMI 设备属性成功。
- 非 0 设置 HDMI 设备属性失败，详情参照错误码。

➢ 依赖

头文件：mi_hdmi.h、mi_ hdmi_datatype.h
库文件：libmi_hdmi.a

※ 注意

- 设置 HDMI 属性之前需要调用 MI_HDMI_Open。
- 用户需要在调用 MI_HDMI_Start 之前设置 HDMI 属性。
- 在调用 MI_HDMI_Start 之后设置 HDMI 属性，需要先调用 MI_HDMI_Stop 再设置新的 HDMI 属性。
- 传输 audio 时需要确保有传输视频数据，音频是夹到视频中传输的

➢ 举例

```
1.  MI_HDMI_InitPara_t stHdmiPara;
2.  MI_HDMI_Attr_t  stAttr;
3.  MI_HDMI_TimingType_e enVideoFmt;
4.  /* 启动 DISP */
5.  /* HDMI start 后设置 HDMI 属性 */
6.  MI_HDMI_Stop(0, MI_TRUE);
7.  MI_HDMI_GetAttr(0, &stAttr);
8.  /* 用户更改 HDMI 属性，如 timing */
9.  stAttr.eVideoFmt = E_MI_HDMI_VIDEO_FMT_720P_60;
10. MI_HDMI_SetAttr(0, &stAttr);
11. MI_HDMI_Start(0, MI_FALSE);
```


Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

➢ 相关主题

MI_HDMI_GetAttr

2.6. MI_HDMI_GetAttr

➢ 功能

获取 HDMI 属性。

➢ 语法

MI_S32 MI_HDMI_GetAttr(MI_HDMI_DeviceId_e eHdmi, MI_HDMI_Attr_t *pstAttr);

➢ 语法

参数名称	描述	输入/输出
eHdmi	HDMI 接口号。 取值范围：0	输入
pstAttr	HDMI 属性结构体指针。	输出

➢ 返回值

返回值：
- 0 获取 HDMI 设备属性成功。
- 非 0 获取 HDMI 设备属性失败，详情参照错误码。

➢ 依赖

头文件：mi_hdmi.h、mi_ hdmi_datatype.h
库文件：libmi_hdmi.a

※ 注意

无。

➢ 举例

请参见 MI_HDMI_Init 举例。

➢ 相关主题

MI_HDMI_SetAttr

2.7. MI_HDMI_Start

➢ 功能

启动 HDMI。

➢ 语法

MI_S32 MI_HDMI_Start(MI_HDMI_DeviceId_e eHdmi);

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

- 形参

参数名称	描述	输入/输出
eHdmi	HDMI 接口号。 取值范围：0	输入

- 返回值

返回值：
- 0 启动 HDMI 工作成功。
- 非 0 HDMI 设备启动失败，详情参照错误码。

- 依赖

头文件：mi_hdmi.h、mi_ hdmi_datatype.h
库文件：libmi_hdmi.a

※ 注意

- 启动 HDMI 前必须先调用 MI_HDMI_Open。

- 举例

请参见 MI_HDMI_Init 举例。

- 相关主题

MI_HDMI_Stop

2.8. MI_HDMI_Stop

- 功能

停止 HDMI 设备工作。

- 语法

MI_S32 MI_HDMI_Stop(MI_HDMI_DeviceId_e eHdmi);

- 形参

参数名称	描述	输入/输出
eHdmi	HDMI 接口号。 取值范围：0	输入

- 返回值

返回值：
- 0 停止 HDMI 设备工作。
- 非 0 停止 HDMI 设备失败，详情参照错误码。

- 依赖

头文件：mi_hdmi.h、mi_ hdmi_datatype.h
库文件：libmi_hdmi.a

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

※ 注意

无。

➢ 相关主题

MI_HDMI_Start

2.9. MI_HDMI_GetSinkInfo

➢ 功能

获取 HDMI Sink 端的能力级信息。

➢ 语法

MI_S32 MI_HDMI_GetSinkInfo(MI_HDMI_DeviceId_e eHdmi,
MI_HDMI_SinkInfo_t *pstSinkInfo);

➢ 形参

参数名称	描述	输入/输出
eHdmi	HDMI 接口号。取值范围：0	输入
pstSinkCap	HDMI Sink 端能力级结构体指针	输出

➢ 返回值

返回值：
- 0 获取 HDMI Sink 端能力成功。
- 非 0 获取 Sink 信息失败，详情参照错误码。

➢ 依赖

头文件：mi_hdmi.h、mi_ hdmi_datatype.h
库文件：libmi_hdmi.a

※ 注意

- 调用前必须先打开 HDMI。为了确保 HDMI 打开完成，建议在调用 MI_HDMI_Open 接口后等待 500ms 左右。
- 应在 HDMI 启动且插入线缆之后调用。
- 一般在插入事件处理函数中或者强制获取 EDID 之后使用。

➢ 举例

```
1.  MI_HDMI_Init_Para_t stHdmiPara;
2.  MI_HDMI_Attr_t  stAttr;
3.  MI_HDMI_TimingType_e enVideoFmt;
4.  MI_HDMI_SinkInfo_t stSinkInfo;
5.  /* 启动 DISP */
6.  /* 启动 HDMI */
7.  /* 获取 HDMI Sink 端能力级 */
8.  MI_HDMI_GetSinkInfo(0, &stSinkInfo);
```

➢ 相关主题

无。

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

2.10. MI_HDMI_SetAvMute

➢ 功能

设置 HDMI 音视频静音功能。

➢ 定义

MI_S32 MI_HDMI_SetAvMute(MI_HDMI_DeviceId_e eHdmi, MI_BOOL bAvMute);

➢ 形参

参数名称	描述	输入/输出
eHdmi	HDMI 接口号。 取值范围：0	输入
bAvMute	是否设置 HDMI 音视频静音。取值范围： MI_TRUE：设置音视频静音。 MI_FALSE：关闭音视频静音。	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_hdmi.h、mi_ hdmi_datatype.h
库文件：libmi _hdmi.a

※ 注意

需要在调用 `MI_HDMI_Start` 之后使用。

➢ 举例

```
1.  MI_HDMI_Start(0);
2.  MI_HDMI_SetMute(0, MI_TRUE);
```

➢ 相关主题

无。

2.11. MI_HDMI_ForceGetEdid

➢ 功能

获取 HDMI 的 EDID 信息。

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

➢ 语法

MI_S32 MI_HDMI_ForceGetEdid(MI_HDMI_DeviceId_e eHdmi,
MI_HDMI_Edid_t *pstEdidData);

➢ 形参

参数名称	描述	输入/输出
eHdmi	HDMI 接口号。 取值范围：0	输入
pstEdidData	HDMI 的 EDID 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_hdmi.h、mi_ hdmi_datatype.h
库文件：libmi _hdmi.a

※ 注意

- 调用前必须先打开 HDMI。

➢ 举例

```
1. MI_HDMI_InitPara_t stHdmiPara;
2. MI_HDMI_Attr_t stAttr;
3. MI_HDMI_Edid_t stEdidData;
4. /* 启动 DISP */
5. /* 启动 HDMI */
6. /* 获取 HDMI EDID 信息 */
7. MI_HDMI_ForceGetEdid(0, &stEdidData);
```

➢ 相关主题

无。

2.12. MI_HDMI_SetDeepColor

➢ 功能

设置 Deep Color 模式。

➢ 功能

MI_S32 MI_HDMI_SetDeepColor(MI_HDMI_DeviceId_e eHdmi,
MI_HDMI_DeepColor_e eDeepColor);

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

➢ 形参

参数名称	描述	输入/输出
eHdmi	HDMI 接口号。 取值范围：0	输入
eDeepColor	HDMI 的 Deep Color 模式。	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_hdmi.h、mi_ hdmi_datatype.h
库文件：libmi_hdmi.a

※ 注意

- 该接口属于高级接口，一般不需要调用。
- 调用前必须先打开 HDMI。
- 该接口可用于设置 MI_HDMI_SetAttr 不支持的某些 Deep Color 模式。
- 该接口暂未实现。

➢ 举例

```
1.  MI_HDMI_InitPara_t stHdmiPara;
2.  MI_HDMI_Attr_t  stAttr;
3.  /* 启动 DISP*/
4.  /* 启动 HDMI */
5.  /* 设置 Deep color 模式 */
6.  MI_HDMI_SetDeepColor (0, E_MI_HDMI_DEEP_COLOR_24BIT);
```

2.13. MI_HDMI_GetDeepColor

➢ 功能

获取 Deep Color 模式。

➢ 语法

MI_S32 MI_HDMI_GetDeepColor(MI_HDMI_DeviceId_e eHdmi,
MI_HDMI_DeepColor_e *peDeepColor);

➢ 形参

参数名称	描述	输入/输出
eHdmi	HDMI 接口号。取值范围：0	输入
peDeepColor	HDMI 的 Deep Color 模式指针。	输入

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

- 返回值

返回值：
- 0 成功。
- 非 0 初始化失败，详情参照错误码。

- 依赖

头文件：mi_hdmi.h、mi_ hdmi_datatype.h
库文件：libmi_hdmi.a

※ 注意

- 该接口属于高级接口，一般不需要调用。
- 调用前必须先打开 HDMI。
- 该接口暂未实现。

- 举例

无。

- 相关主题

MI_HDMI_SetDeepColor

2.14. MI_HDMI_SetInfoFrame

- 功能

设置 HDMI 的信息帧。

- 语法

```
MI_S32 MI_HDMI_SetInfoFrame(MI_HDMI_DeviceId_e eHdmi,
    MI_HDMI_InfoFrame_t *pstInfoFrame);
```

- 形参

参数名称	描述	输入/输出
eHdmi	HDMI 接口号。 取值范围：0	输入
pstInfoFrame	图像信息帧结构体指针。	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_hdmi.h、mi_ hdmi_datatype.h
库文件：libmi_hdmi.a

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

※ 注意

- 该接口一般用于用户态回调函数中配置信息帧属性。其他情况，一般不需调用本接口。
- 调用前必须先打开 HDMI。
- 本接口只支持设置 AVI、SPD 和 AUDIO 信息帧，其他类型的信息帧设置无效。
- 建议设置前先用。

➢ 举例

```
1.  MI_HDMI_DeviceId_e eHdmi = E_MI_HDMI_ID_0;
2.  MI_HDMI_InfoFrame_t stAviInfoFrame = {E_MI_HDMI_INFOFRAME_TYPE_AVI};
3.  MI_HDMI_InfoFrame_t stAudInfoFrame = {E_MI_HDMI_INFOFRAME_TYPE_AUDIO};
4.  MI_HDMI_InfoFrame_t stSpdInfoFrame = {E_MI_HDMI_INFOFRAME_TYPE_SPD};
5.
6.  /* AviInfoFrame */
7.  stAviInfoFrame.unInforUnit.stAviInfoFrame.bEnableAfdOverWrite = 0;
8.  ...
9.
10. /* AudioInfoFrame */
11. stAudInfoFrame.unInforUnit.stAudInfoFrame.u32ChannelCount = 2;
12. ...
13.
14. /* SpdInfoFrame */
15. stSpdInfoFrame.unInforUnit.stSpdInfoFrame.au8VendorName[0] = 0;
16. ...
17.
18. MI_HDMI_SetInfoFrame(eHdmi, &stAviInfoFrame);
```

➢ 相关主题

MI_HDMI_GetInfoFrame

2.15. MI_HDMI_GetInfoFrame

➢ 功能

获取 HDMI 的信息帧。

➢ 语法

MI_RESULT MI_HDMI_GetInfoFrame(MI_HDMI_DeviceId_e eHdmi,
MI_HDMI_InfoFrameType_e eInfoFrameType,
MI_HDMI_InfoFrame_t*pstInfoFrame);

➢ 形参

参数名称	描述	输入/输出
eHdmi	HDMI 接口号。 取值范围：0	输入
eInfoFrameType	信息帧类型。	输入
pstInfoFrame	图像信息帧结构体指针。	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。

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

➢ 依赖

头文件：mi_hdmi.h、mi_ hdmi_datatype.h

库文件：libmi_hdmi.a

※ 注意

- 该接口一般用于用户态回调函数中获取信息帧属性。其他情况，一般不需调用本接口。
- 调用前必须先打开 HDMI。
- 建议设置前先用。

2.16. MI_HDMI_SetAnalogDrvCurrent

➢ 功能

设置 HDMI 通道的驱动电流

➢ 语法

MI_RESULT MI_HDMI_SetAnalogDrvCurrent(MI_HDMI_DeviceId_e eHdmi,
MI_HDMI_AnalogDrvCurrent_t *pstAnalogDrvCurrent);

➢ 形参

参数名称	描述	输入/输出
eHdmi	HDMI 接口号。 取值范围：0	输入
pstAnalogDrvCurrent	HDMI 输出电流的结构体类型	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_hdmi.h、mi_ hdmi_datatype.h
库文件：libmi_hdmi.a

※ 注意

- 该接口属于高级接口，提供给用户调整 hdmi 的通道输出电流。
- 调用前必须先打开 HDMI。

➢ 举例

```
1.  MI_HDMI_AnalogDrvCurrent_t CurInfo = {};
2.
3.  memset(&CurInfo, 0xFF, sizeof(CurInfo));
4.  MI_HDMI_SetAnalogDrvCurrent(g_HdmiAttr.eHdmi, &CurInfo);
```


Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

3. HDMI 数据类型

HDMI 相关数据类型定义如下：

MI_HDMI_DeviceId_e	定义 HDMI 接口号
MI_HDMI_InitPara_t	定义 HDMI 初始化属性结构体
MI_HDMI_EventCallBack	定义 HDMI 回调函数
MI_HDMI_EventType_e	定义 HDMI 事件通知类型
MI_HDMI_Attr_t	定义 HDMI 属性结构体
MI_HDMI_TimingType_e	定义 HDMI 视频制式类型
MI_HDMI_OutputMode_e	定义 HDMI 颜色空间类型
MI_HDMI_ColorType_e	定义 HDMI 输出颜色类型
MI_HDMI_DeepColor_e	定义 HDMI 深色模式类型
MI_HDMI_SampleRate_e	定义 HDMI 音频输出采样率类型
MI_HDMI_BitDepth_e	定义 HDMI 音频输出采样位宽类型
MI_HDMI_SinkInfo_t	定义 HDMI Sink 端能力级结构体
MI_HDMI_Edid_t	定义 HDMI EDID 信息结构体
MI_HDMI_InfoFrameType_e	定义 HDMI 信息帧类型
MI_HDMI_AviInfoFrameVer_t	定义 AVI 信息帧结构体
MI_HDMI_AudInfoFrameVer_t	定义 AUDIO 信息帧结构体
MI_HDMI_SpdInfoFrame_t	定义 SPD 信息帧结构体
MI_HMDI_InfoFrameUnit_u	定义 HDMI 信息帧联合体
MI_HDMI_InfoFrame_t	定义 HDMI 信息帧结构体
MI_HDMI_AnalogDrvCurrent_t	定义 HDMI 通道驱动电流结构体

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_ hdmi_datatype.h

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

3.1. MI_HDMI_DeviceId_e

➢ 说明

定义 HDMI 接口号。

➢ 定义

```
typedef enum
{
    E_MI_HDMI_ID_0 = 0,
    E_MI_HDMI_ID_BUTT
} MI_HDMI_DeviceId_e;
```

➢ 成员

成员名称	描述
E_MI_HDMI_ID_0	HDMI 接口 0

※ 注意事项

无。

➢ 相关数据类型及接口

无。

3.2. MI_HDMI_InitPara_t

➢ 说明

定义 HDMI 初始化属性结构体。

➢ 定义

```
typedef struct MI_HDMI_InitPara_s
{
    MI_HDMI_EventCallBack    pfnHdmiEventCallback;
    void                     *pCallBackArgs;
}MI_HDMI_InitPara_t;
```

➢ 成员

成员名称	描述
pfnHdmiEventCallback	事件处理回调函数
pCallBackArgs	回调函数参数私有数据

※ 注意事项

无。

➢ 相关数据类型及接口

MI_HDMI_Init

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

3.3. MI_HDMI_EventCallBack

➢ 说明

定义 HDMI 回调函数。

➢ 定义

```
typedef MI_RESULT (*MI_HDMI_EventCallBack)
(MI_HDMI_DeviceId_e eHdmi, MI_HDMI_EventType_e event, void        entParam, void
*pUsrParam);
```

➢ 成员

成员名称	描述
MI_HDMI_EvenType_e	HDMI 事件类型。
pEventParam pUsrParam	事件私有数据。

※ 注意事项

无

➢ 相关数据类型及接口

无

3.4. MI_HDMI_EventType_e

➢ 说明

定义 HDMI 事件通知类型。

➢ 定义

```
typedef enum
{
        E_MI_HDMI_EVENT_HOTPLUG = 0,
        E_MI_HDMI_EVENT_NO_PLUG,
        E_MI_HDMI_EVENT_MAX
} MI_HDMI_EventType_e;
```

➢ 成员

成员名称	描述
E_MI_HDMI_EVENT_HOTPLUG	HDMI Cable 连接事件。
E_MI_HDMI_EVENT_NO_PLUG	HDMI Cable 断开连接事件。

※ 注意事项

无

➢ 相关数据类型及接口

MI_HDMI_Init

※ 注意事项

无。

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

3.5. MI_HDMI_Attr_t

- 说明

  定义 HDMI 属性结构体。

- 定义

```
typedef struct MI_HDMI_Attr_s
{
    MI_HDMI_VideoAttr_t stVideoAttr;
    MI_HDMI_AudioAttr_t stAudioAttr;
    MI_HDMI_EnInfoFrame_t stEnInfoFrame;
} MI_HDMI_Attr_t;

typedef struct MI_HDMI_VideoAttr_s
{
    MI_BOOL                 bEnableVideo;
    MI_HDMI_TimingType_e eTimingType;
    MI_HDMI_OutputMode_e eOutputMode;
    MI_HDMI_DeepColor_e     eDeepColorMode;
} MI_HDMI_VideoAttr_t;

typedef struct MI_HDMI_AudioAttr_s
{
    MI_BOOL                 bEnableAudio;
    MI_BOOL                 bIsMultiChannel;
    MI_HDMI_SampleRate_e    eSampleRate;
    MI_HDMI_BitDepth_e      eBitDepth;
} MI_HDMI_VideoAttr_t;

typedef struct MI_HDMI_EnInfoFrame_s
{
    MI_BOOL                 bEnableAviInfoFrame;
    MI_BOOL                 bEnableAudInfoFrame;
    MI_BOOL                 bEnableSpdInfoFrame;
    MI_BOOL                 bEnableMpegInfoFrame;
} MI_HDMI_EnInfoFrame_t;
```

- 成员

成员名称	描述
bEnableVideo	是否输出视频
eTimingType	视频制式，此参数需要与视频输出配置的制式保持一致。
eOutputMode	HDMI 输出视频模式。
eDeepColorMode	DeepColor 输出模式。 默认为 MI_HDMI_DEEP_COLOR_24BIT 不支持 MI_HDMI_DEEP_COLOR_30BIT 和 MI_HDMI_DEEP_COLOR_36BIT。 MI_HDMI_DEEP_COLOR_48BIT
bEnableAudio	是否使能音频。
bIsMultiChannel	多声道数。 0：2 声道； 1：8 声道。//目前不支持
enSampleRate	音频采样率，此参数需要与 AO 的配置保持一致
enBitDepth	音频位宽，默认为 16，此参数需要与 AO 的配置保持一致。
bEnableAviInfoFrame	是否使能 AVI InfoFrame。

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

成员名称	描述
	建议使能
bEnableAudInfoFrame	是否使能 AUDIO InfoFrame。 建议使能
bEnableSpdInfoFrame	是否使能 SPD InfoFrame。

※ 注意事项

- 用户可以根据建议值设置 HDMI 属性

➢ 相关数据类型及接口

MI_HDMI_SetAttr

3.6. MI_HDMI_TimingType_e

➢ 说明

定义 HDMI 视频制式类型。

➢ 定义

```
typedef enum
{
    E_MI_HDMI_TIMING_480_60I        = 0,
    E_MI_HDMI_TIMING_480_60P        = 1,
    E_MI_HDMI_TIMING_576_50I        = 2,
    E_MI_HDMI_TIMING_576_50P        = 3,
    E_MI_HDMI_TIMING_720_50P        = 4,
    E_MI_HDMI_TIMING_720_60P        = 5,
    E_MI_HDMI_TIMING_1080_50I       = 6,
    E_MI_HDMI_TIMING_1080_50P       = 7,
    E_MI_HDMI_TIMING_1080_60I       = 8,
    E_MI_HDMI_TIMING_1080_60P       = 9,
    E_MI_HDMI_TIMING_1080_30P       = 10,
    E_MI_HDMI_TIMING_1080_25P       = 11,
    E_MI_HDMI_TIMING_1080_24P       = 12,
    E_MI_HDMI_TIMING_4K2K_30P       = 13,
    E_MI_HDMI_TIMING_1440_50P       = 14,
    E_MI_HDMI_TIMING_1440_60P       = 15,
    E_MI_HDMI_TIMING_1440_24P       = 16,
    E_MI_HDMI_TIMING_1440_30P       = 17,
    E_MI_HDMI_TIMING_1470_50P       = 18,
    E_MI_HDMI_TIMING_1470_60P       = 19,
    E_MI_HDMI_TIMING_1470_24P       = 20,
    E_MI_HDMI_TIMING_1470_30P       = 21,
    E_MI_HDMI_TIMING_1920x2205_24P   = 22,
    E_MI_HDMI_TIMING_1920x2205_30P   = 23,
    E_MI_HDMI_TIMING_4K2K_25P       = 24,
    E_MI_HDMI_TIMING_4K1K_60P       = 25,
    E_MI_HDMI_TIMING_4K2K_60P       = 26,
    E_MI_HDMI_TIMING_4K2K_24P       = 27,
    E_MI_HDMI_TIMING_4K2K_50P       = 28,
    E_MI_HDMI_TIMING_2205_24P       = 29,
    E_MI_HDMI_TIMING_4K1K_120P      = 30,
    E_MI_HDMI_TIMING_4096x2160_24P   = 31,
    E_MI_HDMI_TIMING_4096x2160_25P   = 32,
    E_MI_HDMI_TIMING_4096x2160_30P   = 33,
```


Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

```
    E_MI_HDMI_TIMING_4096x2160_50P   = 34,
    E_MI_HDMI_TIMING_4096x2160_60P   = 35,
    E_MI_HDMI_TIMING_1024x768_60P    = 36,
    E_MI_HDMI_TIMING_1280x1024_60P   = 37,
    E_MI_HDMI_TIMING_1440x900_60P    = 38,
    E_MI_HDMI_TIMING_1600x1200_60P   = 39,
    E_MI_HDMI_TIMING_1280x800_60P    = 40,
    E_MI_HDMI_TIMING_1366x768_60P    = 41,
    E_MI_HDMI_TIMING_1680x1050_60P   = 42,
    E_MI_HDMI_TIMING_MAX,
} MI_HDMI_TimingType_e;
```

➢ 成员

无。

※ 注意事项

- 需要根据视频输出的制式设置相应的 HDMI 的制式。
- 当 AP 在设置 timing 的时候，需要匹配 DISP 模块的 timing，HDMI 的 timing 需和 DISP 模块的 timing 保持一致。

➢ 相关数据类型及接口

MI_HDMI_SetAttr

3.7. MI_HDMI_OutputMode_e

➢ 说明

定义 HDMI 输出模式类型

➢ 定义

```
typedef enum
{
        E_MI_HDMI_OUTPUT_MODE_HDMI = 0,
        E_MI_HDMI_OUTPUT_MODE_HDMI_HDCP,
        E_MI_HDMI_OUTPUT_MODE_DVI,
        E_MI_HDMI_OUTPUT_MODE_DVI_HDCP,
        E_MI_HDMI_OUTPUT_MODE_MAX,
} MI_HDMI_OutputMode_e;
```

➢ 成员

成员名称	描述
E_MI_HDMI_OUTPUT_MODE_HDMI	HDMI 输出模式
E_MI_HDMI_OUTPUT_MODE_HDMI_HDCP	HDMI + DHCP 输出模式
E_MI_HDMI_OUTPUT_MODE_DVI	DVI 输出模式
E_MI_HDMI_OUTPUT_MODE_DVI_HDCP	DVI + DHCP 输出模式

※ 注意事项

无。

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

➢ 相关数据类型及接口

MI_HDMI_SetAttr

3.8. MI_HDMI_ColorType_e

➢ 说明

定义 HDMI 颜色空间类型。

➢ 定义

```
typedef enum
{
        E_MI_HDMI_COLOR_TYPE_RGB444 = 0,
        E_MI_HDMI_COLOR_TYPE_YCBCR422,
        E_MI_HDMI_COLOR_TYPE_YCBCR444,
        E_MI_HDMI_COLOR_TYPE_YCBCR420,
        E_MI_HDMI_COLOR_TYPE_MAX
} MI_HDMI_ColorType_e;
```

➢ 成员

成员名称	描述
E_MI_HDMI_COLOR_TYPE_RGB444	RGB444 输出模式
E_MI_HDMI_COLOR_TYPE_YCBCR422	YCBCR422 输出模式
E_MI_HDMI_COLOR_TYPE_YCBCR444	YCBCR444 输出模式
E_MI_HDMI_COLOR_TYPE_YCBCR420	YCBCR420 输出模式

※ 注意事项

无。

➢ 相关数据类型及接口

MI_HDMI_SetAttr

3.9. MI_HDMI_DeepColor_e

➢ 说明

定义 HDMI 深色模式类型。

➢ 定义

```
typedef enum
{
        E_MI_HDMI_DEEP_COLOR_24BIT = 0x00,
        E_MI_HDMI_DEEP_COLOR_30BIT,
        E_MI_HDMI_DEEP_COLOR_36BIT,
        E_MI_HDMI_DEEP_COLOR_48BIT,
        E_MI_HDMI_DEEP_COLOR_MAX,
} MI_HDMI_DeepColor_e;
```


Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

➢ 成员

成员名称	描述
E_MI_HDMI_DEEP_COLOR_24BIT	HDMI Deep Color 24bit 模式
E_MI_HDMI_DEEP_COLOR_30BIT	HDMI Deep Color 30bit 模式
E_MI_HDMI_DEEP_COLOR_36BIT	HDMI Deep Color 36bit 模式
E_MI_HDMI_DEEP_COLOR_48BIT	HDMI Deep Color 48bit 模式

※ 注意事项

无。

➢ 相关数据类型及接口

MI_HDMI_SetAttr

3.10. MI_HDMI_SampleRate_e

➢ 说明

定义 HDMI 音频输出采样率类型。

➢ 定义

```
typedef enum
{
    E_MI_HDMI_AUDIO_SAMPLERATE_UNKNOWN      = 0,
    E_MI_HDMI_AUDIO_SAMPLERATE_32K          = 1,
    E_MI_HDMI_AUDIO_SAMPLERATE_44K          = 2,
    E_MI_HDMI_AUDIO_SAMPLERATE_48K          = 3,
    E_MI_HDMI_AUDIO_SAMPLERATE_88K          = 4,
    E_MI_HDMI_AUDIO_SAMPLERATE_96K          = 5,
    E_MI_HDMI_AUDIO_SAMPLERATE_176K         = 6,
    E_MI_HDMI_AUDIO_SAMPLERATE_192K         = 7,
    E_MI_HDMI_AUDIO_SAMPLERATE_MAX,
} MI_HDMI_SampleRate_e;
```

➢ 成员

无。

※ 注意事项

- 当 AP 在设置采样率的时候，需要匹配 Audio 模块的采样率，HDMI 的采样率需和 Audio 模块的采样率保持一致。

➢ 相关数据类型及接口

MI_HDMI_SetAttr

3.11. MI_HDMI_BitDepth_e

➢ 说明

定义 HDMI 音频输出采样位宽类型。

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

➢ 定义

```
typedef enum
{
        E_MI_HDMI_BIT_DEPTH_UNKNOWN =0,
        E_MI_HDMI_BIT_DEPTH_8       = 8,
        E_MI_HDMI_BIT_DEPTH_16    = 16,
        E_MI_HDMI_BIT_DEPTH_18    = 18,
        E_MI_HDMI_BIT_DEPTH_20    = 20,
        E_MI_HDMI_BIT_DEPTH_24    = 24,
        E_MI_HDMI_BIT_DEPTH_32    = 32,
        E_MI_HDMI_BIT_DEPTH_BUTT
}MI_HDMI_BitDepth_e;
```

➢ 成员

无。

※ 注意事项

无。

➢ 相关数据类型及接口

MI_HDMI_SetAttr

3.12. MI_HDMI_SinkInfo_t

➢ 说明

定义 HDMI Sink 端能力级结构体。

➢ 定义

```
typedef struct MI_HDMI_Sink_Info_s
{
    MI_BOOL bConnected;
    MI_BOOL bSupportHdmi;
    MI_HDMI_TimingType_e eNativeTimingType;
    MI_BOOL abVideoFmtSupported[E_MI_HDMI_TIMING_MAX];
    MI_BOOL bSupportYCbCr444;
    MI_BOOL bSupportYCbCr422;
    MI_BOOL bSupportYCbCr;
    MI_BOOL bSupportxvYcc601;
    MI_BOOL bSupportxvYcc709;
    MI_U8 u8MdBit;
    MI_BOOL abAudioFmtSupported[MI_HDMI_MAX_ACAP_CNT];
    MI_U32 au32AudioSampleRateSupported[MI_HDMI_MAX_AUDIO_SAMPLE_RATE_CNT];
    MI_U32 u32MaxPcmChannels;
    MI_U8 u8Speaker;
    MI_U8 au8IdManufactureName[4];
    MI_U32 u32IdProductCode;
    MI_U32 u32IdSerialNumber;
    MI_U32 u32WeekOfManufacture;
    MI_U32 u32YearOfManufacture;
    MI_U8 u8Version;
    MI_U8 u8Revision;
    MI_U8 u8EdidExternBlockNum;
    MI_U8 au8IeeRegId[3];//IEEE registeration identifier
```


Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

```
    MI_U8 u8PhyAddr_A;
    MI_U8 u8PhyAddr_B;
    MI_U8 u8PhyAddr_C;
    MI_U8 u8PhyAddr_D;
    MI_BOOL bSupportDviDual;
    MI_BOOL bSupportDeepColorYcbcr444;
    MI_BOOL bSupportDeepColor30Bit;
    MI_BOOL bSupportDeepColor36Bit;
    MI_BOOL bSupportDeepColor48Bit;
    MI_BOOL bSupportAi;
    MI_U32 u8MaxTmdsClock;
    MI_BOOL bILatencyFieldsPresent;
    MI_BOOL bLatencyFieldsPresent;
    MI_BOOL bHdmiVideoPresent;
    MI_U8 u8VideoLatency;
    MI_U8 u8AudioLatency;
    MI_U8 u8InterlacedVideoLatency;
    MI_U8 u8InterlacedAudioLatency;
} MI_HDMI_SinkInfo_t;
```

- 成员

成员名称	描述
bConnected	设备是否连接
bSupportHdmi	设备是否支持 HDMI，如果不支持，则为 DVI 设备
eNativeVideoFormat	显示设备物理分辨率。
abVideoFmtSupported	视频能力集。 MI_TRUE：支持这种显示格式； MI_FALSE：不支持。
bSupportYCbCr	是否支持 YCBCR 显示。 MI_TRUE：支持 YCBCR 显示； MI_FALSE：只支持 RGB 显示。
bSupportxvYCC601	是否支持 xvYCC601 颜色格式。
bSupportxvYCC709	是否支持 xvYCC709 颜色格式
u8MDBit	xvYCC601 支持的传输 profile:1:P0,2:P1,4:P2
bAudioFmtSupported	音频能力集，请参考 EIA-CEA-861-D 表 37 MI_TRUE：支持这种显示格式； MI_FALSE：不支持。
u32AudioSampleRateSupported	音频采样率能力集，0 为非法值，其他为支持的音频采样率。
u32MaxPcmChannels	音频最大的 PCM 通道数。
u8Speaker	扬声器位置，请参考 EIA-CEA-861-D 中 SpeakerDATABlock 的定义。
u8IDManufactureName	设备厂商标识。
u32IDProductCode	设备 ID
u32IDSerialNumber	设备序列号
u32WeekOfManufacture	设备生产日期(周)。
u32YearOfManufacture	设备生产日期(年)。
u8Version	设备版本号
u8Revision	设备子版本号
au8EDIDExternBlockNum	EDID 扩展块数目
au8IEERegId	24-bit IEEE Registration Identifier (0x000C03)。
u8PhyAddr_A	CEC 物理地址 A。

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

成员名称	描述
u8PhyAddr_B	CEC 物理地址 B。
u8PhyAddr_C	CEC 物理地址 C。
u8PhyAddr_D	CEC 物理地址 D。
bSupportDVIDual	是否支持 DVI dual-link 操作。
bSupportDeepColorYCBCR444	是否支持 YCBCR 4:4:4 Deep Color 模式。
bSupportDeepColor36Bit	是否支持 Deep Color 36bit 模式
bSupportDeepColor48Bit	是否支持 Deep Color 48bit 模式
bSupportAI	是否支持 Supports_AI 模式
u8MaxTMDSClock	最大 TMDS 时钟
bLatencyFieldsPresent	延时标志位。
bHDMIVideoPresent	Video_Latency 和 Audio_Latency fields 是否存在。
u8VideoLatency	视频延时。
u8AudioLatency	音频延时。
u8InterlacedVideoLatency	隔行视频模式下的视频延时。
u8InterlacedAudioLatency	隔行视频模式下的音频延时。

※ 注意事项

无。

➢ 相关数据类型及接口

MI_HDMI_GetSinkInfo

3.13. MI_HDMI_Edid_t

➢ 说明

定义 HDMI EDID 信息结构体

➢ 定义

```
typedef struct MI_HDMI_Edid_s
{
        MI_BOOL bEdidValid;
        MI_U32 u32Edidlength;
        MI_U8 au8Edid[512];
} MI_HDMI_Edid_t;
```

➢ 成员

成员名称	描述
bEdidValid	EDID 信息是否有效。
u32Edidlength	EDID 信息数据长度，一般为 256 Byte
au8Edid	au8Edid

※ 注意事项

无。

➢ 相关数据类型及接口

MI_HDMI_ForceGetEdid

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

3.14. MI_HDMI_InfoFrameType_e

➢ 说明

定义 HDMI 信息帧类型。

➢ 定义

```
typedef enum
{
        E_MI_INFOFRAME_TYPE_AVI = 0,
        E_MI_INFOFRAME_TYPE_SPD,
        E_MI_INFOFRAME_TYPE_AUDIO,
        E_MI_INFOFRAME_TYPE_MAX
} MI_HDMI_InfoFrameType_e;
```

➢ 成员

成员名称	描述
E_MI_INFOFRAME_TYPE_AVI	AVI 信息帧类型
E_MI_INFOFRAME_TYPE_SPD	SPD 信息帧类型。
E_MI_INFOFRAME_TYPE_AUDIO	AUDIO 信息帧类型。

※ 注意事项

无。

➢ 相关数据类型及接口

MI _HDMI_DeInit
MI_HDMI_SetInfoFrame

3.15. MI_HDMI_AviInfoFrameVer_t

➢ 说明

定义 AVI 信息帧结构体。

➢ 定义

```
typedef struct MI_HDMIAviInforFrameVer_s
{
    MI_BOOL  enableAFDoverWrite;
    MI_U8 A0Value;
    MI_HDMI_ColorType_e eColorType;
    MI_HDMI_Colorimetry_e eColorimetry;
    MI_HDMI_ExtColorimetry_e eExtColorimetry;
    MI_HDMI_YccQuantRange_e eYccQuantRange;
    MI_HDMI_TimingType_e eTimingType; //trans to MS_VIDEO_TIMING in impl
    MI_HDMI_VideoAfdRatio_e eAfdRatio;
    MI_HDMI_ScanInfo_e eScanInfo;
    MI_HDMI_AspectRatio_e eAspectRatio;
} MI_HDMI_AviInfoFrameVer_t;
```


Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

➢ 成员

成员名称	描述
eColorType	颜色空间类型。
eScanInfo	扫描信息枚举变量。
eColorimetry	色彩空间矩阵。
eAspectRatio	幅形比枚举变量。
eExtColorimetry	扩展色彩空间矩阵。
eTimingType	视频格式类型。
eYccQuantRange	YCC 色彩空间范围。

※ 注意事项

- 各成员变量的枚举成员详见 mi_ hdmi_datatype.h 文件。

➢ 相关数据类型及接口

MI_HDMI_SetInfoFrame
MI_HDMI_GetInfoFrame

3.16. MI_HDMI_AudInfoFrameVer_t

➢ 说明

定义 AUDIO 信息帧结构体。

➢ 定义

```
typedef struct MI_HDMI_AudInfoFrameVer_s
{
        MI_U32 u32ChannelCount;
        MI_HDMI_AudioCodeType_e eAudioCodeType;
        MI_HDMI_SampleRate_e eSampleRate;
} MI_HDMI_AudInfoFrameVer_t;
```

➢ 成员

成员名称	描述
u32ChannelCount	通道个数。
eAudioCodeType	音频数据编码类型。
eSampleRate	采样频率。推荐设 0.
	0：表由码流头决定。
	1：32kHz
	2：44.1 kHz
	3：48 kHz
	4：88.2 kHz
	5：96 kHz
	6：176.4 kHz
	7：192 kHz

※ 注意事项

无。

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

➢ 相关数据类型及接口

MI_HDMI_SetInfoFrame
MI_HDMI_GetInfoFrame

3.17. MI_HDMI_SpdInfoFrame_t

➢ 说明

定义 SPD 信息帧结构体。

➢ 定义

```
typedef struct MI_HDMI_SpdInfoFrame_s
{
        MI_U8 au8VendorName[8];
        MI_U8 au8ProductDescription[16];
} MI_HDMI_SpdInfoFrame_t;
```

➢ 成员

成员名称	描述
au8VendorName[8]	卖方名称。
au8ProductDescription[16]	产品描述符。

※ 注意事项

无。

➢ 相关数据类型及接口

MI_HDMI_SetInfoFrame
MI_HDMI_GetInfoFrame

3.18. MI_HMDI_InfoFrame_Unit_u

➢ 说明

定义 HDMI 信息帧联合体。

➢ 定义

```
typedef union
{
        MI_HDMI_Avi InfoFrameVer_t      stAviInfoFrame;
        MI_HDMI_AudInfoFrameVer_t       stAudInfoFrame;
        MI_HDMI_SpdInfoFrame_t          stSpdInfoFrame;
} MI_HDMI_InfoFrameUnit_u;
```

➢ 成员

成员名称	描述
stAviInfoFrame	AVI 信息帧
stAudInfoFrame	AUD 信息帧
stSpdInfoFrame	SPD 信息帧

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

※ 注意事项

无。

➢ 相关数据类型及接口

MI_HDMI_SetInfoFrame
MI_HDMI_GetInfoFrame

3.19. MI_HDMI_AnalogDrvCurrent_t

➢ 说明

定义 HDMI 通道驱动电流结构体。

➢ 定义

```
typedef struct MI_HDMI_AnalogDrvCurrent_s
{
    MI_U8 u8DrvCurTap1Ch0;
    MI_U8 u8DrvCurTap1Ch1;
    MI_U8 u8DrvCurTap1Ch2;
    MI_U8 u8DrvCurTap1Ch3;
    MI_U8 u8DrvCurTap2Ch0;
    MI_U8 u8DrvCurTap2Ch1;
    MI_U8 u8DrvCurTap2Ch2;
    MI_U8 u8DrvCurTap2Ch3;
} MI_HDMI_AnalogDrvCurrent_t;
```

➢ 成员

成员名称	描述
u8DrvCurTap1Ch0	Tap1 通道 0 驱动电流配置
u8DrvCurTap1Ch1	Tap1 通道 1 驱动电流配置
u8DrvCurTap1Ch2	Tap1 通道 2 驱动电流配置
u8DrvCurTap1Ch3	Tap1 通道 3 驱动电流配置
u8DrvCurTap2Ch0	Tap2 通道 0 驱动电流配置
u8DrvCurTap2Ch1	Tap2 通道 1 驱动电流配置
u8DrvCurTap2Ch2	Tap2 通道 2 驱动电流配置
u8DrvCurTap2Ch3	Tap2 通道 3 驱动电流配置

※ 注意事项

无。

➢ 相关数据类型及接口

MI_HDMI_SetAnalogDrvCurrent

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.

4. HDMI 错误码

HDMI API 错误码如表所示：

HDMI API 错误码

错误代码	宏定义	描述
0xA00B2011	MI_ERR_HDMI_INVALID_PARAM	非法参数
0xA00B2001	MI_ERR_HDMI_INVALID_DEVID	无效设备 ID
0xA00B2018	MI_ERR_HDMI_DRV_FAILED	HDMI 驱动返回失败
0xA00B2015	MI_ERR_HDMI_NOT_INIT	HDMI 未初始化
0xA00B2008	MI_ERR_HDMI_NOT_SUPPORT	不支持操作
0xA00B21E0	MI_ERR_HDMI_UNSUPPORT_TIMING	不支持输出 timing
0xA00B21E1	MI_ERR_HDMI_UNSUPPORT_COLORTYPE	不支持颜色类型
0xA00B21E2	MI_ERR_HDMI_UNSUPPORT_ACODETYPE	不支持的音频编码类型
0xA00B21E3	MI_ERR_HDMI_UNSUPPORT_AFREQ	不支持的音频频率
0xA00B21E4	MI_ERR_HDMI_EDID_HEADER_ERR	EDID 头信息解析失败
0xA00B21E5	MI_ERR_HDMI_EDID_DATA_ERR	EDID 数据错误

Copyright © 2020 SigmaStar Technology Corp. All rights reserved.