



MACARON High-Integrated IP Camera SoC Processor

ISP 軟件開發參考 Version 0.6



SigmaStar Confidential

© 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
0.1 (APIVer 0.1)	<ul style="list-style-type: none">Initial release	01/25/2019
0.2 (APIVer 0.2)	<ul style="list-style-type: none">Modified YEE API (by adding overshoot/undershoot limit)	02/19/2019
0.3 (APIVer 0.3)	<ul style="list-style-type: none">Added R2Y APIUpdated ColorTrans API description	04/09/2019
0.4 (APIVer 0.4)	<ul style="list-style-type: none">Added AE auto flickerAdded LSC/LSC_CTRL apiAdded ALSC_CTRL apiAdded DynamicDP_CLUSTER apiUpdated WDR api (by adding gamma sync enable)	04/23/2019
0.5 (APIVer 0.4)	<ul style="list-style-type: none">Updated ColorTrans API Parameter	05/17/2019
0.6 (APIVer 0.5)	<ul style="list-style-type: none">Updated AWB API	05/21/2019

TABLE OF CONTENTS

1. 概述.....	1
2. MI ISP API 参考.....	2
2.1. MI ISP API 概述.....	2
2.2. 功能函数参考.....	2
2.2.1 MI_ISP_IQ_GetVersionInfo.....	2
2.2.2 MI_ISP_IQ_GetParaInitStatus.....	3
2.2.3 MI_ISP_IQ_GetIQInd.....	4
2.2.4 MI_ISP_IQ_SetColorToGray.....	5
2.2.5 MI_ISP_IQ_GetColorToGray.....	6
2.2.6 MI_ISP_IQ_SetContrast.....	6
2.2.7 MI_ISP_IQ_GetContrast.....	7
2.2.8 MI_ISP_IQ_SetBrightness.....	8
2.2.9 MI_ISP_IQ_GetBrightness.....	9
2.2.10 MI_ISP_IQ_SetLightness.....	10
2.2.11 MI_ISP_IQ_GetLightness.....	11
2.2.12 MI_ISP_IQ_SetRGBGamma.....	12
2.2.13 MI_ISP_IQ_GetRGBGamma.....	12
2.2.14 MI_ISP_IQ_SetYUVGamma.....	13
2.2.15 MI_ISP_IQ_GetYUVGamma.....	14
2.2.16 MI_ISP_IQ_SetSaturation.....	15
2.2.17 MI_ISP_IQ_GetSaturation.....	16
2.2.18 MI_ISP_IQ_SetRGBMatrix.....	17
2.2.19 MI_ISP_IQ_GetRGBMatrix.....	17
2.2.20 MI_ISP_IQ_SetFalseColor.....	18
2.2.21 MI_ISP_IQ_GetFalseColor.....	19
2.2.22 MI_ISP_IQ_SetNR3D.....	20
2.2.23 MI_ISP_IQ_GetNR3D.....	21
2.2.24 MI_ISP_IQ_SetNRDeSpike.....	22
2.2.25 MI_ISP_IQ_GetNRDeSpike.....	22
2.2.26 MI_ISP_IQ_SetNRLuma.....	23
2.2.27 MI_ISP_IQ_GetNRLuma.....	24
2.2.28 MI_ISP_IQ_SetNRChroma.....	25
2.2.29 MI_ISP_IQ_GetNRChroma.....	26
2.2.30 MI_ISP_IQ_SetSharpness.....	27
2.2.31 MI_ISP_IQ_GetSharpness.....	27
2.2.32 MI_ISP_IQ_SetCrossTalk.....	28
2.2.33 MI_ISP_IQ_GetCrossTalk.....	29
2.2.34 MI_ISP_IQ_SetOBC.....	30
2.2.35 MI_ISP_IQ_GetOBC.....	31
2.2.36 MI_ISP_IQ_SetOBC_P1.....	32
2.2.37 MI_ISP_IQ_GetOBC_P1.....	32
2.2.38 MI_ISP_IQ_SetWDR.....	33
2.2.39 MI_ISP_IQ_GetWDR.....	34

2.2.40	MI_ISP_IQ_SetDynamicDP	35
2.2.41	MI_ISP_IQ_GetDynamicDP	36
2.2.42	MI_ISP_IQ_SetDynamicDP_CLUSTER	37
2.2.43	MI_ISP_IQ_GetDynamicDP_CLUSTER	37
2.2.44	MI_ISP_IQ_SetHSV	38
2.2.45	MI_ISP_IQ_GetHSV	39
2.2.46	MI_ISP_IQ_SetRGBIR	40
2.2.47	MI_ISP_IQ_GetRGBIR	41
2.2.48	MI_ISP_IQ_SetDEMOSAIC	42
2.2.49	MI_ISP_IQ_GetDEMOSAIC	43
2.2.50	MI_ISP_IQ_SetR2Y	43
2.2.51	MI_ISP_IQ_GetR2Y	44
2.2.52	MI_ISP_IQ_SetCOLORTTRANS	45
2.2.53	MI_ISP_IQ_GetCOLORTTRANS	46
2.2.54	MI_ISP_IQ_SetHDR	47
2.2.55	MI_ISP_IQ_GetHDR	48
2.2.56	MI_ISP_IQ_SetLSC	48
2.2.57	MI_ISP_IQ_GetLSC	49
2.2.58	MI_ISP_IQ_SetLSC_CTRL	50
2.2.59	MI_ISP_IQ_GetLSC_CTRL	51
2.2.60	MI_ISP_IQ_SetALSC	52
2.2.61	MI_ISP_IQ_GetALSC	53
2.2.62	MI_ISP_IQ_SetALSC_CTRL	53
2.2.63	MI_ISP_IQ_GetALSC_CTRL	54
2.2.64	MI_ISP_IQ_SetIQMode	55
2.2.65	MI_ISP_IQ_GetIQMode	56
2.2.66	MI_ISP_IQ_SetAPIBypassMode	57
2.2.67	MI_ISP_IQ_GetAPIBypassMode	58
2.2.68	MI_ISP_AE_QueryExposureInfo	58
2.2.69	MI_ISP_AE_GetHistoWghtY	59
2.2.70	MI_ISP_AE_SetEVComp	60
2.2.71	MI_ISP_AE_GetEVComp	61
2.2.72	MI_ISP_AE_SetExpoMode	62
2.2.73	MI_ISP_AE_GetExpoMode	63
2.2.74	MI_ISP_AE_SetManualExpo	63
2.2.75	MI_ISP_AE_GetManualExpo	64
2.2.76	MI_ISP_AE_SetState	65
2.2.77	MI_ISP_AE_GetState	66
2.2.78	MI_ISP_AE_SetTarget	67
2.2.79	MI_ISP_AE_GetTarget	68
2.2.80	MI_ISP_AE_SetConverge	68
2.2.81	MI_ISP_AE_GetConverge	69
2.2.82	MI_ISP_AE_SetExposureLimit	70
2.2.83	MI_ISP_AE_GetExposureLimit	71

2.2.84	MI_ISP_AE_SetPlainLongExpoTable	72
2.2.85	MI_ISP_AE_GetPlainLongExpoTable	73
2.2.86	MI_ISP_AE_SetPlainShortExpoTable	73
2.2.87	MI_ISP_AE_GetPlainShortExpoTable	74
2.2.88	MI_ISP_AE_SetWinWgtType	75
2.2.89	MI_ISP_AE_GetWinWgtType	76
2.2.90	MI_ISP_AE_SetWinWgt	77
2.2.91	MI_ISP_AE_GetWinWgt	78
2.2.92	MI_ISP_AE_SetFlicker	78
2.2.93	MI_ISP_AE_GetFlicker	79
2.2.94	MI_ISP_AE_SetStrategy	80
2.2.95	MI_ISP_AE_GetStrategy	81
2.2.96	MI_ISP_AWB_QueryInfo	82
2.2.97	MI_ISP_AWB_GetCTStats	83
2.2.98	MI_ISP_AWB_GetHWStats	83
2.2.99	MI_ISP_AWB_SetAttr	84
2.2.100	MI_ISP_AWB_GetAttr	85
2.2.101	MI_ISP_AWB_SetAttrEx	86
2.2.102	MI_ISP_AWB_GetAttrEx	87
2.2.103	MI_ISP_AWB_SetMultiLSAttr	88
2.2.104	MI_ISP_AWB_GetMultiLSAttr	88
2.2.105	MI_ISP_AWB_SetCTCaliAttr	89
2.2.106	MI_ISP_AWB_GetCTCaliAttr	90
2.2.107	MI_ISP_AF_QueryInfo	91
2.2.108	MI_ISP_AF_SetWindow	92
2.2.109	MI_ISP_AF_GetWindow	93
2.2.110	MI_ISP_AF_SetFilter	93
2.2.111	MI_ISP_AF_GetFilter	94
2.2.112	MI_ISP_API_CmdLoadBinFile	95
2.2.113	MI_ISP_API_CmdLoadCaliData	96
3.	MI_ISP_API 数据类型	98
3.1.	MI_ISP_API 结构类型详细描述	98
3.1.1	MI_ISP_BOOL 结构体	98
3.1.2	MI_ISP_OP_TYPE 结构体	98
3.1.3	MI_ISP_SM_STATE_TYPE 结构体	99
3.1.4	MI_ISP_IO_PARAM_MODE 结构体	100
3.1.5	MI_ISP_BYPASS_MODE 结构体	100
3.1.6	MI_ISP_IO_INDEX 结构体	101
3.1.7	MI_ISP_API_ID 结构体	102
3.1.8	MI_ISP_AWB_ALG_TYPE 结构体	104
3.1.9	MI_ISP_CALI_ITEM 结构体	104
3.1.10	MI_ISP_IO_VERSION_INFO 结构体	105
3.1.11	MI_ISP_IO_PARAM_INIT_INFO 结构体	106
3.1.12	MI_ISP_IO_COLOR_TO_GRAY 结构体	107

3.1.13	MI_ISP_IQ_CONTRAST 结构体	108
3.1.14	MI_ISP_IQ_BRIGHTNESS 结构体	110
3.1.15	MI_ISP_IQ_LIGHTNESS 结构体	112
3.1.16	MI_ISP_IQ_RGBGAMMA 结构体	114
3.1.17	MI_ISP_IQ_YUVGAMMA 结构体	117
3.1.18	MI_ISP_IQ_SATURATION 结构体	119
3.1.19	MI_ISP_IQ_RGBMATRIX 结构体	122
3.1.20	MI_ISP_IQ_FALSECOLOR 结构体	124
3.1.21	MI_ISP_IQ_NR3D 结构体	127
3.1.22	MI_ISP_IQ_NRDESPIKE 结构体	130
3.1.23	MI_ISP_IQ_NRLUMA 结构体	133
3.1.24	MI_ISP_IQ_NRChroma 结构体	136
3.1.25	MI_ISP_IQ_SHARPNESS 结构体	138
3.1.26	MI_ISP_IQ_CROSSTALK 结构体	142
3.1.27	MI_ISP_IQ_OBC 结构体	144
3.1.28	MI_ISP_IQ_WDR 结构体	147
3.1.29	MI_ISP_IQ_DYNAMIC_DP 结构体	150
3.1.30	MI_ISP_IQ_DYNAMIC_DP_CLUSTER 结构体	152
3.1.31	MI_ISP_IQ_HSV 结构体	155
3.1.32	MI_ISP_IQ_RGBIR 结构体	158
3.1.33	MI_ISP_IQ_DEMOSAIC 结构体	161
3.1.34	MI_ISP_IQ_R2Y 结构体	163
3.1.35	MI_ISP_IQ_COLORTRANS 结构体	164
3.1.36	MI_ISP_IQ_HDR 结构体	167
3.1.37	MI_ISP_IQ_LSC 结构体	169
3.1.38	MI_ISP_IQ_LSC_CTRL 结构体	170
3.1.39	MI_ISP_IQ_ALSC 结构体	171
3.1.40	MI_ISP_IQ_ALSC_CTRL 结构体	172
3.1.41	MI_ISP_API_BYPASS 结构体	173
3.1.42	MI_ISP_AE_EXPO_VALUE 结构体	173
3.1.43	MI_ISP_AE_HIST_WEIGHT_Y 结构体	174
3.1.44	MI_ISP_AE_EXPO_INFO 结构体	175
3.1.45	MI_ISP_AE_EV_COMP 结构体	176
3.1.46	MI_ISP_AE_MODE 结构体	176
3.1.47	MI_ISP_AE_INTP_LUT 结构体	177
3.1.48	MI_ISP_AE_CONV_CONDITON 结构体	178
3.1.49	MI_ISP_AE_EXPO_LIMIT 结构体	180
3.1.50	MI_ISP_AE_EXPO_TABLE 结构体	180
3.1.51	MI_ISP_AE_WIN_WEIGHT_MODE 结构体	182
3.1.52	MI_ISP_AE_WIN_WEIGHT 结构体	183
3.1.53	MI_ISP_AE_FLICKER 结构体	184
3.1.54	MI_ISP_AE_STRATEGY 结构体	185
3.1.55	MI_ISP_AWB_QUERY_INFO 结构体	187
3.1.56	MI_ISP_AWB_CT_STATISTICS 结构体	188

3.1.57 MI_ISP_AWB_HW_STATISTICS 结构体.....189

3.1.58 MI_ISP_AWB_ATTR 结构体.....189

3.1.59 MI_ISP_AWB_ATTR_EX 结构体.....194

3.1.60 MI_ISP_AWB_MULTILS_LS 结构体.....196

3.1.61 MI_ISP_AWB_CT_CALI 结构体.....197

3.1.62 MI_ISP_AF_QUERY_INFO 结构体.....198

3.1.63 MI_ISP_AF_WINDOW 结构体.....199

3.1.64 MI_ISP_AF_FILTER 结构体.....201

3.2. 错误码.....203

SigmaStar Confidential

LIST OF FIGURES

Figure 1: ISP Block Diagram	1
Figure 2: IQTool 接口设定 Magic key 示意图	96
Figure 3: RGB Gamma curve	116
Figure 4: Chroma 强度的阈值限制条件示意图	127
Figure 5: IIR/FIR 系数	202

SigmaStar Confidential

1. 概述

ISP 模块是对 Video source 输入的数据进行分析、处理，设定相关视频参数以及进行 Camera 的调整，以实现黑电平校正、镜头校正、3A、2D/3D 降噪、CCM、Gamma...等功能。ISP 框图处理如下：

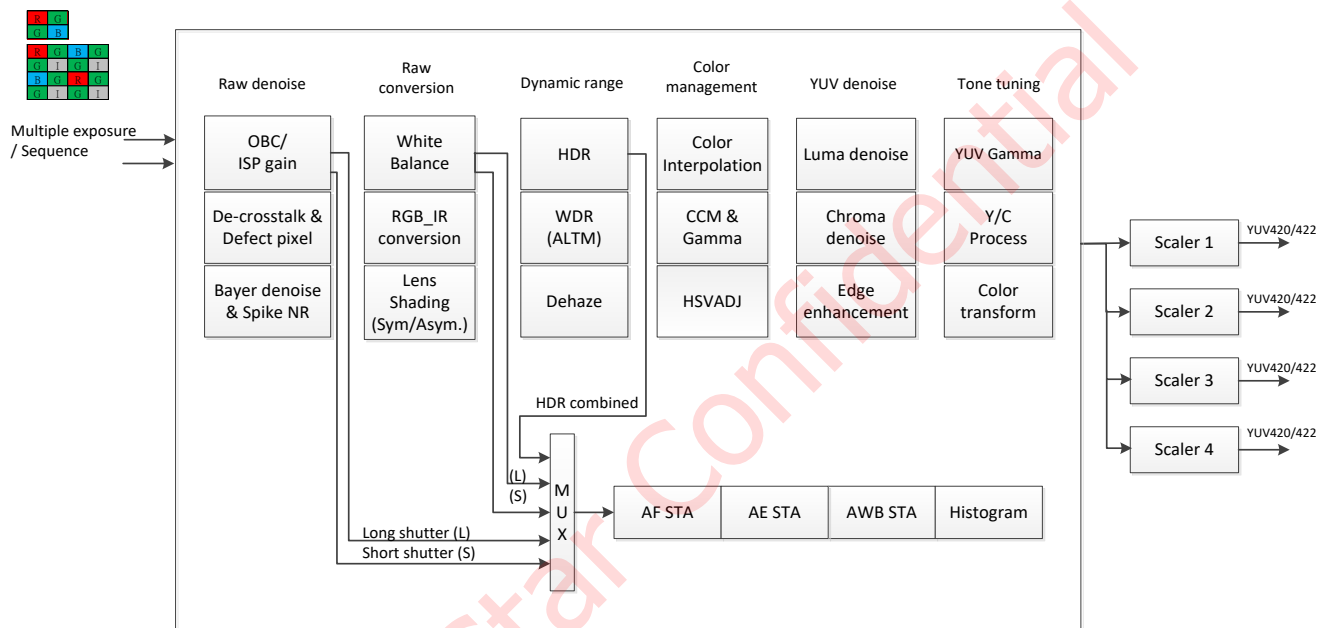


Figure 1: ISP Block Diagram

2. MI ISP API 参考

2.1. MI ISP API 概述

2.2. 功能函数参考

2.2.1 MI_ISP_IQ_GetVersionInfo

2.2.1.1. 目的

取得 IQ 版本号信息。

2.2.1.2. 语法

MI_S32 MI_ISP_IQ_GetVersionInfo(MI_U32 Channel, [MI_ISP_IQ_VERSION_INFO_TYPE_t](#) *data);

2.2.1.3. 描述

调用此函数取得 IQ 版本号信息。

2.2.1.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	IQ 版本号信息的指针。

2.2.1.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.1.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.1.7. 注意

无。

2.2.1.8. 举例

无。

2.2.2 MI_ISP_IQ_GetParaInitStatus

2.2.2.1. 目的

取得 ISP 初始化 ready 状态的参数值。

2.2.2.2. 语法

```
MI_S32 MI_ISP_IQ_GetParaInitStatus(MI_U32 Channel, MI\_ISP\_IQ\_PARAM\_INIT\_INFO\_TYPE\_t *data);
```

2.2.2.3. 描述

调用此函数取得 ISP 初始化 ready 状态的参数值，只有查询到状态为 ready (1)时，才能进行 load ISP api bin file 或者 call ISP api function。

2.2.2.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	ISP 初始化参数值的指标。

2.2.2.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.2.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.2.7. 注意

无。

2.2.2.8. 举例

```

MI_SYS_INIT();
While(1)
{
    MI_ISP_IQ_GetParaInitStatus (0, &FLAG);
    if (FLAG == 1)
    {
        break;
    }
}

```

2.2.3 MI_ISP_IQ_GetIQind

2.2.3.1. 目的

取得曝光指数 (ISO index) 参数值。

2.2.3.2. 语法

MI_S32 MI_ISP_IQ_GetIQind(MI_U32 Channel, [MI_ISP_IQ_INDEX_e](#) *data);

2.2.3.3. 描述

调用此函数取得曝光指数 (ISO index) 参数值。

2.2.3.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	曝光指数 (ISO index) 参数值的指针。

2.2.3.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.3.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.3.7. 注意

无。

2.2.3.8. 举例

无。

2.2.4 MI_ISP_IQ_SetColorToGray

2.2.4.1. 目的

设定彩色转灰阶影像参数值。

2.2.4.2. 语法

MI_S32 MI_ISP_IQ_SetColorToGray(MI_U32 Channel, [MI_ISP_IQ_COLORTOGRAY_TYPE](#) t *data);

2.2.4.3. 描述

调用此函式设定彩色转灰阶影像参数值。

2.2.4.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	彩色转灰阶影像参数值的指针。

2.2.4.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.4.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.4.7. 注意

无。

2.2.4.8. 举例

无。

2.2.5 MI_ISP_IQ_GetColorToGray

2.2.5.1 目的

取得彩色转灰阶影像参数值。

2.2.5.2 语法

MI_S32 MI_ISP_IQ_GetColorToGray(MI_U32 Channel, [MI_ISP_IQ_COLORTOGRAY_TYPE_t](#) *data);

2.2.5.3 描述

调用此函式取得彩色转灰阶影像参数值。

2.2.5.4 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	彩色转灰阶影像参数值的指针。

2.2.5.5 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.5.6 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.5.7 注意

无。

2.2.5.8 举例

无。

2.2.6 MI_ISP_IQ_SetContrast

2.2.6.1 目的

设定对比度参数值。

2.2.6.2 语法

MI_S32 MI_ISP_IQ_SetContrast(MI_U32 Channel, [MI_ISP_IQ_CONTRAST_TYPE_t](#) *data);

2.2.6.3. 描述

调用此函式设定对比度参数值。

2.2.6.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	对比度参数值的指针。

2.2.6.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.6.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.6.7. 注意

无。

2.2.6.8. 举例

无。

2.2.7 MI_ISP_IQ_GetContrast

2.2.7.1. 目的

取得对比度参数值。

2.2.7.2. 语法

MI_S32 MI_ISP_IQ_GetContrast(MI_U32 Channel, [MI_ISP_IQ_CONTRAST_TYPE_t](#) *data);

2.2.7.3. 描述

调用此函式取得对比度参数值。

2.2.7.5. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	对比度参数值的指针。

2.2.7.6. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.7.7. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.7.8. 注意

无。

2.2.7.9. 举例

无。

2.2.8 MI_ISP_IQ_SetBrightness

2.2.8.1. 目的

设定亮度参数值。

2.2.8.2. 语法

MI_S32 MI_ISP_IQ_SetBrightness(MI_U32 Channel, [MI_ISP_IQ_BRIGHTNESS_TYPE_t](#) *data);

2.2.8.3. 描述

调用此函式设定亮度参数值。

2.2.8.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	亮度参数值的指针。

2.2.8.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.8.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.8.7. 注意

无。

2.2.8.8. 举例

无。

2.2.9 MI_ISP_IO_GetBrightness

2.2.9.1. 目的

取得亮度参数值。

2.2.9.2. 语法

MI_S32 MI_ISP_IO_GetBrightness(MI_U32 Channel, [MI_ISP_IO_BRIGHTNESS_TYPE t](#) *data);

2.2.9.3. 描述

调用此函式取得亮度参数值。

2.2.9.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	亮度参数值的指针。

2.2.9.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.9.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.9.7. 注意

无。

2.2.9.8. 举例

无。

2.2.10 MI_ISP_IQ_SetLightness

2.2.10.1. 目的

设定灰度参数值。

2.2.10.2. 语法

MI_S32 MI_ISP_IQ_SetLightness(MI_U32 Channel, [MI_ISP_IQ_LIGHTNESS_TYPE t](#) *data);

2.2.10.3. 描述

调用此函式设定灰度参数值。

2.2.10.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	灰度参数值的指针。

2.2.10.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.10.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.10.7. 注意

无。

2.2.10.8. 举例

无。

2.2.11 MI_ISP_IQ_GetLightness

2.2.11.1. 目的

取得灰度参数值。

2.2.11.2. 语法

MI_S32 MI_ISP_IQ_GetLightness(MI_U32 Channel, [MI_ISP_IQ_LIGHTNESS_TYPE_t](#) *data);

2.2.11.3. 描述

调用此函式取得灰度参数值。

2.2.11.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	灰度参数值的指针。

2.2.11.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.11.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.11.7. 注意

无。

2.2.11.8. 举例

无。

2.2.12 MI_ISP_IQ_SetRGBGamma

2.2.12.1 目的

设定 RGB Gamma curve 参数值。

2.2.12.2 语法

MI_S32 MI_ISP_IQ_SetRGBGamma(MI_U32 Channel, [MI_ISP_IQ_RGBGAMMA_TYPE_t](#) *data);

2.2.12.3 描述

调用此函式设定 RGB Gamma curve 参数值。

2.2.12.4 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	RGB Gamma curve 参数值的指针。

2.2.12.5 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.12.6 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.12.7 注意

无。

2.2.12.8 举例

无。

2.2.13 MI_ISP_IQ_GetRGBGamma

2.2.13.1 目的

取得 RGB Gamma curve 参数值。

2.2.13.2 语法

MI_S32 MI_ISP_IQ_GetRGBGamma(MI_U32 Channel, [MI_ISP_IQ_RGBGAMMA_TYPE_t](#) *data);

2.2.13.3. 描述

调用此函式取得 RGB Gamma curve 参数值。

2.2.13.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	RGB Gamma curve 参数值的指针。

2.2.13.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.13.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.13.7. 注意

无。

2.2.13.8. 举例

无。

2.2.14 MI_ISP_IQ_SetYUVGamma

2.2.14.1. 目的

设定 YUV Gamma curve 参数值。

2.2.14.2. 语法

MI_S32 MI_ISP_IQ_SetYUVGamma(MI_U32 Channel, [MI_ISP_IQ_YUVGAMMA_TYPE_t](#) *data);

2.2.14.3. 描述

调用此函式设定 YUV Gamma curve 参数值。

2.2.14.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	YUV Gamma curve 参数值的指针。

2.2.14.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.14.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.14.7. 注意

无。

2.2.14.8. 举例

无。

2.2.15 MI_ISP_IQ_GetYUVGamma

2.2.15.1. 目的

取得 YUV Gamma curve 参数值。

2.2.15.2. 语法

MI_S32 MI_ISP_IQ_GetYUVGamma(MI_U32 Channel, [MI_ISP_IQ_YUVGAMMA_TYPE_t](#) *data);

2.2.15.3. 描述

调用此函式取得 YUV Gamma curve 参数值。

2.2.15.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	YUV Gamma curve 参数值的指针。

2.2.15.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.15.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.15.7. 注意

无。

2.2.15.8. 举例

无。

2.2.16 MI_ISP_IQ_SetSaturation

2.2.16.1. 目的

设定饱和度参数值。

2.2.16.2. 语法

MI_S32 MI_ISP_IQ_SetSaturation(MI_U32 Channel, [MI_ISP_IQ_SATURATION_TYPE t](#) *data);

2.2.16.3. 描述

调用此函式设定饱和度参数值。

2.2.16.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	饱和度参数值的指针。

2.2.16.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.16.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.16.7. 注意

无。

2.2.16.8. 举例

无。

2.2.17 MI_ISP_IQ_GetSaturation

2.2.17.1. 目的

取得饱和度参数值。

2.2.17.2. 语法

MI_S32 MI_ISP_IQ_GetSaturation(MI_U32 Channel, [MI_ISP_IQ_SATURATION_TYPE_t](#) *data);

2.2.17.3. 描述

调用此函式取得饱和度参数值。

2.2.17.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	饱和度参数值的指针。

2.2.17.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.17.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.17.7. 注意

无。

2.2.17.8. 举例

无。

2.2.18 MI_ISP_IQ_SetRGBMatrix

2.2.18.1. 目的

设定色彩校正矩阵参数值。

2.2.18.2. 语法

MI_S32 MI_ISP_IQ_SetRGBMatrix(MI_U32 Channel, [MI_ISP_IQ_RGBMATRIX_TYPE_t](#) *data);

2.2.18.3. 描述

调用此函式设定色彩校正矩阵参数值。

2.2.18.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	色彩校正矩阵参数值的指针。

2.2.18.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.18.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.18.7. 注意

无。

2.2.18.8. 举例

无。

2.2.19 MI_ISP_IQ_GetRGBMatrix

2.2.19.1. 目的

取得色彩校正矩阵参数值。

2.2.19.2. 语法

MI_S32 MI_ISP_IQ_GetRGBMatrix(MI_U32 Channel, [MI_ISP_IQ_RGBMATRIX_TYPE t](#) *data);

2.2.19.3. 描述

调用此函式取得色彩校正矩阵参数值。

2.2.19.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	色彩校正矩阵参数值的指针。

2.2.19.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.19.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.19.7. 注意

无。

2.2.19.8. 举例

无。

2.2.20 MI_ISP_IQ_SetFalseColor

2.2.20.1. 目的

设定去伪彩参数值。

2.2.20.2. 语法

MI_S32 MI_ISP_IQ_SetFalseColor(MI_U32 Channel, [MI_ISP_IQ_FALSECOLOR_TYPE t](#) *data);

2.2.20.3. 描述

调用此函式设定去伪彩参数值。

2.2.20.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	去伪彩参数值的指针。

2.2.20.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.20.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.20.7. 注意

无。

2.2.20.8. 举例

无。

2.2.21 MI_ISP_IQ_GetFalseColor

2.2.21.1. 目的

取得去伪彩参数值。

2.2.21.2. 语法

MI_S32 MI_ISP_IQ_GetFalseColor(MI_U32 Channel, [MI_ISP_IQ_FALSECOLOR_TYPE_t](#) *data);

2.2.21.3. 描述

调用此函式取得去伪彩参数值。

2.2.21.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	去伪彩参数值的指针。

2.2.21.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.21.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.21.7. 注意

无。

2.2.21.8. 举例

无。

2.2.22 MI_ISP_IO_SetNR3D

2.2.22.1. 目的

设定 3D 降噪参数值。

2.2.22.2. 语法

MI_S32 MI_ISP_IO_SetNR3D(MI_U32 Channel, [MI_ISP_IO_NR3D_TYPE_t](#) *data);

2.2.22.3. 描述

调用此函式设定 3D 降噪参数值。

2.2.22.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	3D 降噪参数值的指针。

2.2.22.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.22.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.22.7. 注意

无。

2.2.22.8. 举例

无。

2.2.23 MI_ISP_IQ_GetNR3D

2.2.23.1. 目的

取得 3D 降噪参数值。

2.2.23.2. 语法

MI_S32 MI_ISP_IQ_GetNR3D(MI_U32 Channel, [MI_ISP_IQ_NR3D_TYPE t](#) *data);

2.2.23.3. 描述

调用此函式取得 3D 降噪参数值。

2.2.23.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	3D 降噪参数值的指针。

2.2.23.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.23.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.23.7. 注意

无。

2.2.23.8. 举例

无。

2.2.24 MI_ISP_IQ_SetNRDeSpike

2.2.24.1. 目的

设定 DeSpike 降噪参数值。

2.2.24.2. 语法

MI_S32 MI_ISP_IQ_SetNRDeSpike(MI_U32 Channel, [MI_ISP_IQ_NRDESPIKE_TYPE_t](#) *data);

2.2.24.3. 描述

调用此函式设定 DeSpike 降噪参数值。

2.2.24.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	DeSpike 降噪参数值的指针。

2.2.24.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.24.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.24.7. 注意

无。

2.2.24.8. 举例

无。

2.2.25 MI_ISP_IQ_GetNRDeSpike

2.2.25.1. 目的

取得 DeSpike 降噪参数值。

2.2.25.2. 语法

MI_S32 MI_ISP_IQ_GetNRDeSpike(MI_U32 Channel, [MI_ISP_IQ_NRDESPIKE_TYPE_t](#) *data);

2.2.25.3. 描述

调用此函式取得 DeSpike 降噪参数值。

2.2.25.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	DeSpike 降噪参数值的指针。

2.2.25.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.25.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.25.7. 注意

无。

2.2.25.8. 举例

无。

2.2.26 MI_ISP_IQ_SetNRLuma

2.2.26.1. 目的

设定 Luma 降噪参数值。

2.2.26.2. 语法

MI_S32 MI_ISP_IQ_SetNRLuma(MI_U32 Channel, [MI_ISP_IQ_NRLUMA_TYPE_t](#) *data);

2.2.26.3. 描述

调用此函式设定 Luma 降噪参数值。

2.2.26.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	Luma 降噪参数值的指针。

2.2.26.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.26.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.26.7. 注意

无。

2.2.26.8. 举例

无。

2.2.27 MI_ISP_IQ_GetNRLuma

2.2.27.1. 目的

取得 Luma 降噪参数值。

2.2.27.2. 语法

MI_S32 MI_ISP_IQ_GetNRLuma(MI_U32 Channel, [MI_ISP_IQ_NRLUMA_TYPE_t](#) *data);

2.2.27.3. 描述

调用此函式取得 Luma 降噪参数值。

2.2.27.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	Luma 降噪参数值的指针。

2.2.27.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.27.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.27.7. 注意

无。

2.2.27.8. 举例

无。

2.2.28 MI_ISP_IO_SetNRChroma

2.2.28.1. 目的

设定颜色降噪参数值。

2.2.28.2. 语法

MI_S32 MI_ISP_IO_SetNRChroma(MI_U32 Channel, [MI_ISP_IO_NRChroma_TYPE_t](#) *data);

2.2.28.3. 描述

调用此函式设定颜色降噪参数值。

2.2.28.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	颜色降噪参数值的指针。

2.2.28.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.28.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.28.7. 注意

无。

2.2.28.8. 举例

无。

2.2.29 MI_ISP_IQ_GetNRChroma

2.2.29.1. 目的

取得颜色降噪参数值。

2.2.29.2. 语法

MI_S32 MI_ISP_IQ_GetNRChroma(MI_U32 Channel, [MI_ISP_IQ_NRChroma_TYPE_t](#) *data);

2.2.29.3. 描述

调用此函式取得颜色降噪参数值。

2.2.29.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	颜色降噪参数值的指针。

2.2.29.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.29.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.29.7. 注意

无。

2.2.29.8. 举例

无。

2.2.30 MI_ISP_IQ_SetSharpness

2.2.30.1. 目的

设定锐度参数值。

2.2.30.2. 语法

MI_S32 MI_ISP_IQ_SetSharpness(MI_U32 Channel, [MI_ISP_IQ_SHARPNESS_TYPE_t](#) *data);

2.2.30.3. 描述

调用此函式设定锐度参数值。

2.2.30.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	锐度参数值的指针。

2.2.30.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.30.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.30.7. 注意

无。

2.2.30.8. 举例

无。

2.2.31 MI_ISP_IQ_GetSharpness

2.2.31.1. 目的

取得锐度参数值。

2.2.31.2. 语法

MI_S32 MI_ISP_IQ_GetSharpness(MI_U32 Channel, [MI_ISP_IQ_SHARPNESS_TYPE_t](#) *data);

2.2.31.3. 描述

调用此函式取得锐度参数值。

2.2.31.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	锐度参数值的指针。

2.2.31.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.31.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.31.7. 注意

无。

2.2.31.8. 举例

无。

2.2.32 MI_ISP_IQ_SetCrossTalk

2.2.32.1. 目的

设定串扰参数值。

2.2.32.2. 语法

MI_S32 MI_ISP_IQ_SetCrossTalk(MI_U32 Channel, [MI_ISP_IQ_CROSSTALK_TYPE_t](#) *data);

2.2.32.3. 描述

调用此函式设定串扰参数值。

2.2.32.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	串扰参数值的指针。

2.2.32.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.32.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.32.7. 注意

无。

2.2.32.8. 举例

无。

2.2.33 MI_ISP_IQ_GetCrossTalk

2.2.33.1. 目的

取得串扰参数值。

2.2.33.2. 语法

MI_S32 MI_ISP_IQ_GetCrossTalk(MI_U32 Channel, [MI_ISP_IQ_CROSSTALK_TYPE_t](#) *data);

2.2.33.3. 描述

调用此函式取得串扰参数值。

2.2.33.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	串扰参数值的指针。

2.2.33.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.33.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.33.7. 注意

无。

2.2.33.8. 举例

无。

2.2.34 MI_ISP_IO_SetOBC

2.2.34.1. 目的

设定黑电平参数值。

2.2.34.2. 语法

MI_S32 MI_ISP_IO_SetOBC(MI_U32 Channel, [MI_ISP_IO_OBC_TYPE_t](#) *data);

2.2.34.3. 描述

调用此函式设定黑电平参数值。

2.2.34.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	黑电平参数值的指针。

2.2.34.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.34.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.34.7. 注意

无。

2.2.34.8. 举例

无。

2.2.35 MI_ISP_IQ_GetOBC

2.2.35.1. 目的

取得黑电平参数值。

2.2.35.2. 语法

MI_S32 MI_ISP_IQ_GetOBC(MI_U32 Channel, [MI_ISP_IQ_OBC_TYPE t](#) *data);

2.2.35.3. 描述

调用此函式取得黑电平参数值。

2.2.35.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	黑电平参数值的指针。

2.2.35.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.35.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.35.7. 注意

无。

2.2.35.8. 举例

无。

2.2.36 MI_ISP_IQ_SetOBC_P1

2.2.36.1. 目的

设定 pipeline 1 黑电平参数值。

2.2.36.2. 语法

MI_S32 MI_ISP_IQ_SetOBC_P1(MI_U32 Channel, [MI_ISP_IQ_OBC_TYPE_t](#) *data);

2.2.36.3. 描述

调用此函数设定 pipeline 1 黑电平参数值。

2.2.36.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	Pipeline 1 黑电平参数值的指针。

2.2.36.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.36.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.36.7. 注意

无。

2.2.36.8. 举例

无。

2.2.37 MI_ISP_IQ_GetOBC_P1

2.2.37.1. 目的

取得 pipeline 1 黑电平参数值。

2.2.37.2. 语法

MI_S32 MI_ISP_IQ_GetOBC_P1(MI_U32 Channel, [MI_ISP_IQ_OBC_TYPE t](#) *data);

2.2.37.3. 描述

调用此函式取得 pipeline 1 黑电平参数值。

2.2.37.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	Pipeline 1 黑电平参数值的指针。

2.2.37.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.37.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.37.7. 注意

无。

2.2.37.8. 举例

无。

2.2.38 MI_ISP_IQ_SetWDR

2.2.38.1. 目的

设定宽动态参数值。

2.2.38.2. 语法

MI_S32 MI_ISP_IQ_SetWDR(MI_U32 Channel, [MI_ISP_IQ_WDR_TYPE t](#) *data);

2.2.38.3. 描述

调用此函式设定宽动态参数值。

2.2.38.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	宽动态参数值的指针。

2.2.38.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.38.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.38.7. 注意

无。

2.2.38.8. 举例

无。

2.2.39 MI_ISP_IQ_GetWDR

2.2.39.1. 目的

取得宽动态参数值。

2.2.39.2. 语法

MI_S32 MI_ISP_IQ_GetWDR(MI_U32 Channel, [MI_ISP_IQ_WDR_TYPE_t](#) *data);

2.2.39.3. 描述

调用此函式取得宽动态参数值。

2.2.39.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	宽动态参数值的指针。

2.2.39.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.39.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.39.7. 注意

无。

2.2.39.8. 举例

无。

2.2.40 MI_ISP_IQ_SetDynamicDP

2.2.40.1. 目的

设定动态补偿坏点参数值。

2.2.40.2. 语法

MI_S32 MI_ISP_IQ_SetDynamicDP(MI_U32 Channel, [MI_ISP_IQ_DYNAMIC_DP_TYPE_t](#) *data);

2.2.40.3. 描述

调用此函式设定动态补偿坏点参数值。

2.2.40.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	动态补偿坏点参数值的指针。

2.2.40.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.40.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.40.7. 注意

无。

2.2.40.8. 举例

无。

2.2.41 MI_ISP_IQ_GetDynamicDP

2.2.41.1. 目的

取得动态补偿坏点参数值。

2.2.41.2. 语法

MI_S32 MI_ISP_IQ_GetDynamicDP(MI_U32 Channel, [MI_ISP_IQ_DYNAMIC_DP_TYPE_t](#) *data);

2.2.41.3. 描述

调用此函式取得动态补偿坏点参数值。

2.2.41.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	动态补偿坏点参数值的指针。

2.2.41.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.41.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.41.7. 注意

无。

2.2.41.8. 举例

无。

2.2.42 MI_ISP_IQ_SetDynamicDP_CLUSTER

2.2.42.1. 目的

设定动态补偿坏点参数值。

2.2.42.2. 语法

MI_S32 MI_ISP_IQ_SetDynamicDP_CLUSTER(MI_U32 Channel, [MI_ISP_IQ_DYNAMIC_DP_CLUSTER_TYPE_t](#) *data);

2.2.42.3. 描述

调用此函式设定动态补偿坏点参数值。

2.2.42.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	动态补偿坏点参数值的指针。

2.2.42.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.42.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.42.7. 注意

无。

2.2.42.8. 举例

无。

2.2.43 MI_ISP_IQ_GetDynamicDP_CLUSTER

2.2.43.1. 目的

取得动态补偿坏点参数值。

2.2.43.2. 语法

MI_S32 MI_ISP_IQ_GetDynamicDP_CLUSTER(MI_U32 Channel, [MI_ISP_IQ_DYNAMIC_DP_CLUSTER_TYPE_t](#) *data);

2.2.43.3. 描述

调用此函数取得动态补偿坏点参数值。

2.2.43.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	动态补偿坏点参数值的指针。

2.2.43.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.43.6. 需求

- 头文件: mi_isp.h
- 函数库: libmi_isp.so

2.2.43.7. 注意

无。

2.2.43.8. 举例

无。

2.2.44 MI_ISP_IQ_SetHSV

2.2.44.1. 目的

设定 HSV 参数值, 可做局部色相的旋转与饱和度的增减。

2.2.44.2. 语法

MI_S32 MI_ISP_IQ_SetHSV(MI_U32 Channel, [MI_ISP_IQ_HSV_TYPE_t](#) *data);

2.2.44.3. 描述

调用此函式设定 HSV 参数值。

2.2.44.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	HSV 参数值的指针。

2.2.44.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.44.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.44.7. 注意

无。

2.2.44.8. 举例

无。

2.2.45 MI_ISP_IQ_GetHSV

2.2.45.1. 目的

取得 HSV 参数值。

2.2.45.2. 语法

MI_S32 MI_ISP_IQ_GetHSV(MI_U32 Channel, [MI_ISP_IQ_HSV_TYPE_t](#) *data);

2.2.45.3. 描述

调用此函式取得 HSV 参数值。

2.2.45.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	HSV 参数值的指针。

2.2.45.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.45.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.45.7. 注意

无。

2.2.45.8. 举例

无。

2.2.46 MI_ISP_IQ_SetRGBIR

2.2.46.1. 目的

设定 RGBIR 参数值。

2.2.46.2. 语法

MI_S32 MI_ISP_IQ_SetRGBIR(MI_U32 Channel, [MI_ISP_IQ_RGBIR_TYPE_t](#) *data);

2.2.46.3. 描述

调用此函式设定 RGBIR 参数值。

2.2.46.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	RGBIR 参数值的指针。

2.2.46.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.46.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.46.7. 注意

无。

2.2.46.8. 举例

无。

2.2.47 MI_ISP_IO_GetRGBIR

2.2.47.1. 目的

取得 RGBIR 参数值。

2.2.47.2. 语法

MI_S32 MI_ISP_IO_GetRGBIR(MI_U32 Channel, [MI_ISP_IO_RGBIR_TYPE_t](#) *data);

2.2.47.3. 描述

调用此函式取得 RGBIR 参数值。

2.2.47.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	RGBIR 参数值的指针。

2.2.47.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.47.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.47.7. 注意

无。

2.2.47.8. 举例

无。

2.2.48 MI_ISP_IQ_SetDEMOSAIC

2.2.48.1. 目的

设定去马赛克参数值。

2.2.48.2. 语法

MI_S32 MI_ISP_IQ_SetDEMOSAIC(MI_U32 Channel, [MI_ISP_IQ_DEMOSAIC_TYPE_t](#) *data);

2.2.48.3. 描述

调用此函式设定去马赛克参数值。

2.2.48.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	去马赛克参数值的指针。

2.2.48.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.48.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.48.7. 注意

无。

2.2.48.8. 举例

无。

2.2.49 MI_ISP_IQ_GetDEMOSAIC

2.2.49.1. 目的

取得去马赛克参数值。

2.2.49.2. 语法

MI_S32 MI_ISP_IQ_GetDEMOSAIC(MI_U32 Channel, [MI_ISP_IQ_DEMOSAIC_TYPE_t](#) *data);

2.2.49.3. 描述

调用此函数取得去马赛克参数值。

2.2.49.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	去马赛克参数值的指针。

2.2.49.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.49.6. 需求

- 头文件: mi_isp.h
- 函数库: libmi_isp.so

2.2.49.7. 注意

无。

2.2.49.8. 举例

无。

2.2.50 MI_ISP_IQ_SetR2Y

2.2.50.1. 目的

设定色彩转换参数值。

2.2.50.2. 语法

MI_S32 MI_ISP_IQ_SetR2Y(MI_U32 Channel, [MI_ISP_IQ_R2Y_TYPE_t](#) *data);

2.2.50.3. 描述

调用此函式设定色彩转换参数值。

2.2.50.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	色彩转换参数值的指针。

2.2.50.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.50.6. 需求

- 头文件 : *mi_isp.h*
- 函式庫 : *libmi_isp.so*

2.2.50.7. 注意

无。

2.2.50.8. 举例

无。

2.2.51 MI_ISP_IQ_GetR2Y

2.2.51.1. 目的

设定色彩转换参数值。

2.2.51.2. 语法

MI_S32 MI_ISP_IQ_GetR2Y(MI_U32 Channel, [MI_ISP_IQ_R2Y_TYPE_t](#) *data);

2.2.51.3. 描述

调用此函式设定色彩转换参数值。

2.2.51.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	色彩转换参数值的指针。

2.2.51.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.51.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.51.7. 注意

无。

2.2.51.8. 举例

无。

2.2.52 MI_ISP_IQ_SetCOLORTTRANS

2.2.52.1. 目的

设定色彩转换参数值。

2.2.52.2. 语法

MI_S32 MI_ISP_IQ_SetCOLORTTRANS(MI_U32 Channel, [MI_ISP_IQ_COLORTTRANS_TYPE_t](#) *data);

2.2.52.3. 描述

调用此函式设定色彩转换参数值。

2.2.52.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	色彩转换参数值的指针。

2.2.52.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.52.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.52.7. 注意

无。

2.2.52.8. 举例

无。

2.2.53 MI_ISP_IQ_GetCOLORTTRANS

2.2.53.1. 目的

取得色彩转换参数值。

2.2.53.2. 语法

MI_S32 MI_ISP_IQ_GetCOLORTTRANS(MI_U32 Channel, [MI_ISP_IQ_COLORTTRANS_TYPE_t](#) *data);

2.2.53.3. 描述

调用此函式取得色彩转换参数值。

2.2.53.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	色彩转换参数值的指针。

2.2.53.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.53.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.53.7. 注意

无。

2.2.53.8. 举例

无。

2.2.54 MI_ISP_IQ_SetHDR

2.2.54.1. 目的

设定高动态范围 (High Dynamic Range) 参数值。

2.2.54.2. 语法

MI_S32 MI_ISP_IQ_SetHDR(MI_U32 Channel, [MI_ISP_IQ_HDR_TYPE_t](#) *data);

2.2.54.3. 描述

调用此函式设定高动态范围 (High Dynamic Range) 参数值。

2.2.54.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	高动态范围 (High Dynamic Range) 参数值的指针。

2.2.54.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.54.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.54.7. 注意

无。

2.2.54.8. 举例

无。

2.2.55 MI_ISP_IQ_GetHDR

2.2.55.1. 目的

取得高动态范围 (High Dynamic Range) 参数值。

2.2.55.2. 语法

MI_S32 MI_ISP_IQ_GetHDR(MI_U32 Channel, [MI_ISP_IQ_HDR_TYPE_t](#) *data);

2.2.55.3. 描述

调用此函式取得高动态范围 (High Dynamic Range) 参数值。

2.2.55.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	高动态范围 (High Dynamic Range) 参数值的指针。

2.2.55.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.55.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.55.7. 注意

无。

2.2.55.8. 举例

无。

2.2.56 MI_ISP_IQ_SetLSC

2.2.56.1. 目的

设定 LSC 参数值。

2.2.56.2. 语法

MI_S32 MI_ISP_IQ_SetLSC(MI_U32 Channel, [MI_ISP_IQ_LSC_TYPE_t](#) *data);

2.2.56.3. 描述

调用此函式设定 LSC 参数值。

2.2.56.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	LSC 参数值的指针。

2.2.56.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.56.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.56.7. 注意

无。

2.2.56.8. 举例

无。

2.2.57 MI_ISP_IQ_GetLSC

2.2.57.1. 目的

取得 LSC 参数值。

2.2.57.2. 语法

MI_S32 MI_ISP_IQ_GetLSC(MI_U32 Channel, [MI_ISP_IQ_LSC_TYPE_t](#) *data);

2.2.57.3. 描述

调用此函式取得 LSC 参数值。

2.2.57.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	LSC 参数值的指针。

2.2.57.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.57.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.57.7. 注意

无。

2.2.57.8. 举例

无。

2.2.58 MI_ISP_IQ_SetLSC_CTRL

2.2.58.1. 目的

设定 LSC CTRL 参数值。

2.2.58.2. 语法

MI_S32 MI_ISP_IQ_SetLSC_CTRL(MI_U32 Channel, [MI_ISP_IQ_LSC_CTRL_TYPE_t](#) *data);

2.2.58.3. 描述

调用此函式设定 LSC CTRL 参数值。

2.2.58.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	LSC CTRL 参数值的指针。

2.2.58.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.58.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.58.7. 注意

无。

2.2.58.8. 举例

无。

2.2.59 MI_ISP_IO_GetLSC_CTRL

2.2.59.1. 目的

取得 LSC CTRL 参数值。

2.2.59.2. 语法

MI_S32 MI_ISP_IO_GetLSC_CTRL(MI_U32 Channel, [MI_ISP_IO_LSC_CTRL_TYPE_t](#) *data);

2.2.59.3. 描述

调用此函式取得 LSC CTRL 参数值。

2.2.59.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	LSC CTRL 参数值的指针。

2.2.59.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.59.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.59.7. 注意

无。

2.2.59.8. 举例

无。

2.2.60 MI_ISP_IQ_SetALSC

2.2.60.1. 目的

设定 ALSC 参数值。

2.2.60.2. 语法

MI_S32 MI_ISP_IQ_SetALSC(MI_U32 Channel, [MI_ISP_IQ_ALSC_TYPE t](#) *data);

2.2.60.3. 描述

调用此函式设定 ALSC 参数值。

2.2.60.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	ALSC 参数值的指针。

2.2.60.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.60.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.60.7. 注意

无。

2.2.60.8. 举例

无。

2.2.61 MI_ISP_IQ_GetALSC

2.2.61.1. 目的

取得 ALSC 参数值。

2.2.61.2. 语法

MI_S32 MI_ISP_IQ_GetALSC(MI_U32 Channel, [MI_ISP_IQ_ALSC_TYPE_t](#) *data);

2.2.61.3. 描述

调用此函式取得 ALSC 参数值。

2.2.61.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	ALSC 参数值的指针。

2.2.61.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.61.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.61.7. 注意

无。

2.2.61.8. 举例

无。

2.2.62 MI_ISP_IQ_SetALSC_CTRL

2.2.62.1. 目的

设定 ALSC CTRL 参数值。

2.2.62.2. 语法

MI_S32 MI_ISP_IQ_SetALSC_CTRL(MI_U32 Channel, [MI_ISP_IQ_ALSC_CTRL_TYPE t](#) *data);

2.2.62.3. 描述

调用此函式设定 ALSC CTRL 参数值。

2.2.62.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	ALSC CTRL 参数值的指针。

2.2.62.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.62.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.62.7. 注意

无。

2.2.62.8. 举例

无。

2.2.63 MI_ISP_IQ_GetALSC_CTRL

2.2.63.1. 目的

取得 ALSC CTRL 参数值。

2.2.63.2. 语法

MI_S32 MI_ISP_IQ_GetALSC_CTRL(MI_U32 Channel, [MI_ISP_IQ_ALSC_CTRL_TYPE t](#) *data);

2.2.63.3. 描述

调用此函式取得 ALSC 参数值。

2.2.63.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	ALSC 参数值的指针。

2.2.63.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.63.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.63.7. 注意

无。

2.2.63.8. 举例

无。

2.2.64 MI_ISP_IQ_SetIQMode

2.2.64.1. 目的

设定 IQ 模式参数值, 目前仅提供 Day / Night Mode。

2.2.64.2. 语法

MI_S32 MI_ISP_IQ_SetIQMode(MI_U32 Channel, [MI_ISP_IQ_PARAM_MODE_e](#) *data);

2.2.64.3. 描述

调用此函式设定 IQ 模式参数值。

2.2.64.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	IQ 模式参数值的指针。

2.2.64.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.64.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.64.7. 注意

无。

2.2.64.8. 举例

无。

2.2.65 MI_ISP_IQ_GetIQMode

2.2.65.1. 目的

取得 IQ 模式参数值, 目前仅提供 Day / Night Mode。

2.2.65.2. 语法

MI_S32 MI_ISP_IQ_GetIQMode(MI_U32 Channel, [MI_ISP_IQ_PARAM_MODE_e](#) *data);

2.2.65.3. 描述

调用此函式取得 IQ 模式参数值。

2.2.65.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	IQ 模式参数值的指针。

2.2.65.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.65.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.65.7. 注意

无。

2.2.65.8. 举例

无。

2.2.66 MI_ISP_IQ_SetAPIBypassMode

2.2.66.1. 目的

设定 ISP 对应 IP 模块的略过模式(ByPass Mode)参数值。

2.2.66.2. 语法

MI_S32 MI_ISP_IQ_SetAPIBypassMode(MI_U32 Channel, [MI_ISP_API_BYPASS_TYPE_t](#) *data);

2.2.66.3. 描述

调用此函式设定 ISP 对应 IP 模块的略过模式参数值。

2.2.66.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	ISP 对应 IP 模块的略过模式参数值的指针。

2.2.66.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.66.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.66.7. 注意

无。

2.2.66.8. 举例

无。

2.2.67 MI_ISP_IQ_GetAPIBypassMode

2.2.67.1. 目的

取得 ISP 对应 IP 模块的略过模式(ByPass Mode)参数值。

2.2.67.2. 语法

MI_S32 MI_ISP_IQ_GetAPIBypassMode(MI_U32 Channel, [MI_ISP_API_BYPASS_TYPE](#) t *data);

2.2.67.3. 描述

调用此函数取得 ISP 对应 IP 模块的略过模式参数值。

2.2.67.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	ISP 对应 IP 模块的略过模式参数值的指针。

2.2.67.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.67.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.67.7. 注意

无。

2.2.67.8. 举例

无。

2.2.68 MI_ISP_AE_QueryExposureInfo

2.2.68.1. 目的

取得自动曝光参数值。

2.2.68.2. 语法

MI_S32 MI_ISP_AE_QueryExposureInfo(MI_U32 Channel, [MI_ISP_AE_EXPO_INFO_TYPE_t](#) *data);

2.2.68.3. 描述

调用此函式取得自动曝光参数值。

2.2.68.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动曝光参数值的指针。

2.2.68.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.68.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.68.7. 注意

无。

2.2.68.8. 举例

无。

2.2.69 MI_ISP_AE_GetHistoWghtY

2.2.69.1. 目的

取得当前画面亮度及直方图统计资料数值。

2.2.69.2. 语法

MI_S32 MI_ISP_AE_GetHistoWghtY(MI_U32 Channel, [MI_ISP_AE_HIST_WEIGHT_Y_TYPE_t](#) *data);

2.2.69.3. 描述

调用此函式取得当前画面亮度及直方图统计资料数值。

2.2.69.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	当前画面亮度及直方图统计资料数值的指针。

2.2.69.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.69.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.69.7. 注意

无。

2.2.69.8. 举例

无。

2.2.70 MI_ISP_AE_SetEVComp

2.2.70.1. 目的

设定曝光补偿参数值。

2.2.70.2. 语法

MI_S32 MI_ISP_AE_SetEVComp(MI_U32 Channel, [MI_ISP_AE_EV_COMP_TYPE_t](#) *data);

2.2.70.3. 描述

调用此函式设定曝光补偿参数值。

2.2.70.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	曝光补偿参数值的指针。

2.2.70.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.70.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.70.7. 注意

无。

2.2.70.8. 举例

无。

2.2.71 MI_ISP_AE_GetEVComp

2.2.71.1. 目的

取得曝光补偿参数值。

2.2.71.2. 语法

MI_S32 MI_ISP_AE_GetEVComp(MI_U32 Channel, [MI_ISP_AE_EV_COMP_TYPE_t](#) *data);

2.2.71.3. 描述

调用此函式取得曝光补偿参数值。

2.2.71.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	曝光补偿参数值的指针。

2.2.71.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.71.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.71.7. 注意

无。

2.2.71.8. 举例

无。

2.2.72 MI_ISP_AE_SetExpoMode

2.2.72.1. 目的

设定曝光模式参数值。

2.2.72.2. 语法

MI_S32 MI_ISP_AE_SetExpoMode(MI_U32 Channel, [MI_ISP_AE_MODE_TYPE_e](#) *data);

2.2.72.3. 描述

调用此函式设定曝光模式参数值。

2.2.72.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	曝光模式参数值的指针。

2.2.72.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.72.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.72.7. 注意

无。

2.2.72.8. 举例

无。

2.2.73 MI_ISP_AE_GetExpoMode

2.2.73.1. 目的

取得曝光模式参数值。

2.2.73.2. 语法

MI_S32 MI_ISP_AE_GetExpoMode(MI_U32 Channel, [MI_ISP_AE_MODE_TYPE_e](#) *data);

2.2.73.3. 描述

调用此函数取得曝光模式参数值。

2.2.73.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	曝光模式参数值的指针。

2.2.73.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.73.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.73.7. 注意

无。

2.2.73.8. 举例

无。

2.2.74 MI_ISP_AE_SetManualExpo

2.2.74.1. 目的

设定手动曝光参数值。

2.2.74.2. 语法

MI_S32 MI_ISP_AE_SetManualExpo(MI_U32 Channel, [MI_ISP_AE_EXPO_VALUE_TYPE_t](#) *data);

2.2.74.3. 描述

调用此函式设定手动曝光参数值。

2.2.74.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	手动曝光参数值的指针。

2.2.74.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.74.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.74.7. 注意

无。

2.2.74.8. 举例

无。

2.2.75 MI_ISP_AE_GetManualExpo

2.2.75.1. 目的

取得手动曝光参数值。

2.2.75.2. 语法

MI_S32 MI_ISP_AE_GetManualExpo(MI_U32 Channel, [MI_ISP_AE_EXPO_VALUE_TYPE_t](#) *data);

2.2.75.3. 描述

调用此函式取得手动曝光参数值。

2.2.75.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	手动曝光参数值的指针。

2.2.75.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.75.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.75.7. 注意

无。

2.2.75.8. 举例

无。

2.2.76 MI_ISP_AE_SetState

2.2.76.1. 目的

设定自动曝光运作状态参数值。

2.2.76.2. 语法

MI_S32 MI_ISP_AE_SetState(MI_U32 Channel, [MI_ISP_SM_STATE_TYPE_e](#) *data);

2.2.76.3. 描述

调用此函式设定自动曝光运作状态参数值。

2.2.76.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动曝光运作状态参数值的指针。

2.2.76.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.76.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.76.7. 注意

无。

2.2.76.8. 举例

无。

2.2.77 MI_ISP_AE_GetState

2.2.77.1. 目的

取得自动曝光运作状态参数值。

2.2.77.2. 语法

MI_S32 MI_ISP_AE_GetState(MI_U32 Channel, [MI_ISP_SM_STATE_TYPE_e](#) *data);

2.2.77.3. 描述

调用此函式取得自动曝光运作状态参数值。

2.2.77.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动曝光运作状态参数值的指针。

2.2.77.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.77.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.77.7. 注意

无。

2.2.77.8. 举例

无。

2.2.78 MI_ISP_AE_SetTarget

2.2.78.1. 目的

设定不同环境亮度(BV) 值下的画面曝光目标参数值。

2.2.78.2. 语法

MI_S32 MI_ISP_AE_SetTarget(MI_U32 Channel, [MI_ISP_AE_INTP_LUT_TYPE t](#) *data);

2.2.78.3. 描述

调用此函式设定曝光目标参数值。

2.2.78.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	曝光目标参数值的指针。

2.2.78.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.78.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.78.7. 注意

无。

2.2.78.8. 举例

无。

2.2.79 MI_ISP_AE_GetTarget

2.2.79.1. 目的

取得不同环境亮度(BV) 值下的画面曝光目标参数值。

2.2.79.2. 语法

MI_S32 MI_ISP_AE_GetTarget(MI_U32 Channel, [MI_ISP_AE_INTP_LUT_TYPE_t](#) *data);

2.2.79.3. 描述

调用此函式取得曝光目标参数值。

2.2.79.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	曝光目标参数值的指针。

2.2.79.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.79.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.79.7. 注意

无。

2.2.79.8. 举例

无。

2.2.80 MI_ISP_AE_SetConverge

2.2.80.1. 目的

设定曝光收敛参数值。

2.2.80.2. 语法

MI_S32 MI_ISP_AE_SetConverge(MI_U32 Channel, [MI_ISP_AE_CONV_CONDITON_TYPE_t](#) *data);

2.2.80.3. 描述

调用此函式设定曝光收敛参数值。

2.2.80.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	曝光收敛参数值的指针。

2.2.80.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.80.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.80.7. 注意

无。

2.2.80.8. 举例

无。

2.2.81 MI_ISP_AE_GetConverge

2.2.81.1. 目的

取得曝光收敛参数值。

2.2.81.2. 语法

MI_S32 MI_ISP_AE_GetConverge(MI_U32 Channel, [MI_ISP_AE_CONV_CONDITON_TYPE_t](#) *data);

2.2.81.3. 描述

调用此函式取得曝光收敛参数值。

2.2.81.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	曝光收敛参数值的指针。

2.2.81.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.81.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.81.7. 注意

无。

2.2.81.8. 举例

无。

2.2.82 MI_ISP_AE_SetExposureLimit

2.2.82.1. 目的

设定自动曝光变化范围限制参数值。

2.2.82.2. 语法

MI_S32 MI_ISP_AE_SetExposureLimit(MI_U32 Channel, [MI_ISP_AE_EXPO_LIMIT_TYPE_t](#) *data);

2.2.82.3. 描述

调用此函式设定自动曝光变化范围限制参数值。

2.2.82.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动曝光变化范围限制参数值的指针。

2.2.82.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.82.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.82.7. 注意

无。

2.2.82.8. 举例

无。

2.2.83 MI_ISP_AE_GetExposureLimit

2.2.83.1. 目的

取得自动曝光变化范围限制参数值。

2.2.83.2. 语法

MI_S32 MI_ISP_AE_GetExposureLimit(MI_U32 Channel, [MI_ISP_AE_EXPO_LIMIT_TYPE_t](#) *data);

2.2.83.3. 描述

调用此函式取得自动曝光变化范围限制参数值。

2.2.83.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动曝光变化范围限制参数值的指针。

2.2.83.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.83.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.83.7. 注意

无。

2.2.83.8. 举例

无。

2.2.84 MI_ISP_AE_SetPlainLongExpoTable

2.2.84.1. 目的

设定长曝曝光表参数值。

2.2.84.2. 语法

MI_S32 MI_ISP_AE_SetPlainLongExpoTable(MI_U32 Channel, [MI_ISP_AE_EXPO_TABLE_TYPE t](#) *data);

2.2.84.3. 描述

调用此函式设定长曝曝光表参数值。

2.2.84.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	长曝曝光表参数值的指针。

2.2.84.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.84.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.84.7. 注意

无。

2.2.84.8. 举例

无。

2.2.85 MI_ISP_AE_GetPlainLongExpoTable

2.2.85.1. 目的

取得长曝光表参数值。

2.2.85.2. 语法

MI_S32 MI_ISP_AE_GetPlainLongExpoTable(MI_U32 Channel, [MI_ISP_AE_EXPO_TABLE_TYPE_t](#) *data);

2.2.85.3. 描述

调用此函式取得长曝光表参数值。

2.2.85.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	长曝光表参数值的指针。

2.2.85.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.85.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.85.7. 注意

无。

2.2.85.8. 举例

无。

2.2.86 MI_ISP_AE_SetPlainShortExpoTable

2.2.86.1. 目的

设定短曝光表参数值。

2.2.86.2. 语法

MI_S32 MI_ISP_AE_SetPlainShortExpoTable(MI_U32 Channel, [MI_ISP_AE_EXPO_TABLE_TYPE t](#) *data);

2.2.86.3. 描述

调用此函式设定短曝曝光表参数值。

2.2.86.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	短曝曝光表参数值的指针。

2.2.86.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.86.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.86.7. 注意

无。

2.2.86.8. 举例

无。

2.2.87 MI_ISP_AE_GetPlainShortExpoTable

2.2.87.1. 目的

取得短曝曝光表参数值。

2.2.87.2. 语法

MI_S32 MI_ISP_AE_GetPlainShortExpoTable(MI_U32 Channel, [MI_ISP_AE_EXPO_TABLE_TYPE t](#) *data);

2.2.87.3. 描述

调用此函式取得短曝曝光表参数值。

2.2.87.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	短曝曝光表参数值的指针。

2.2.87.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.87.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.87.7. 注意

无。

2.2.87.8. 举例

无。

2.2.88 MI_ISP_AE_SetWinWgtType

2.2.88.1. 目的

设定曝光权重表种类参数值。

2.2.88.2. 语法

MI_S32 MI_ISP_AE_SetWinWgtType(MI_U32 Channel, [MI_ISP_AE_WIN_WEIGHT_MODE_TYPE_e](#) *data);

2.2.88.3. 描述

调用此函式设定曝光权重表种类参数值。

2.2.88.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	曝光权重表种类参数值的指针。

2.2.88.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.88.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.88.7. 注意

无。

2.2.88.8. 举例

无。

2.2.89 MI_ISP_AE_GetWinWgtType

2.2.89.1. 目的

取得曝光权重表种类参数值。

2.2.89.2. 语法

MI_S32 MI_ISP_AE_GetWinWgtType(MI_U32 Channel, [MI_ISP_AE_WIN_WEIGHT_MODE_TYPE_e](#) *data);

2.2.89.3. 描述

调用此函式取得曝光权重表种类参数值。

2.2.89.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	曝光权重表种类参数值的指针。

2.2.89.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.89.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.89.7. 注意

无。

2.2.89.8. 举例

无。

2.2.90 MI_ISP_AE_SetWinWgt

2.2.90.1. 目的

设定曝光权重表参数值。

2.2.90.2. 语法

MI_S32 MI_ISP_AE_SetWinWgt(MI_U32 Channel, [MI_ISP_AE_WIN_WEIGHT_TYPE t](#) *data);

2.2.90.3. 描述

调用此函式设定曝光权重表参数值。

2.2.90.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	曝光权重表参数值的指针。

2.2.90.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.90.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.90.7. 注意

无。

2.2.90.8. 举例

无。

2.2.91 MI_ISP_AE_GetWinWgt

2.2.91.1. 目的

取得曝光权重表参数值。

2.2.91.2. 语法

MI_S32 MI_ISP_AE_GetWinWgt(MI_U32 Channel, [MI_ISP_AE_WIN_WEIGHT_TYPE_t](#) *data);

2.2.91.3. 描述

调用此函式取得曝光权重表参数值。

2.2.91.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	曝光权重表参数值的指针。

2.2.91.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.91.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.91.7. 注意

无。

2.2.91.8. 举例

无。

2.2.92 MI_ISP_AE_SetFlicker

2.2.92.1. 目的

设定抗闪烁参数值。

2.2.92.2. 语法

MI_S32 MI_ISP_AE_SetFlicker(MI_U32 Channel, [MI_ISP_AE_FLICKER_TYPE_e](#) *data);

2.2.92.3. 描述

调用此函数设定抗闪烁参数值，支持 50Hz 和 60Hz 抗闪烁。

2.2.92.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	抗闪烁参数值的指针。

2.2.92.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.92.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.92.7. 注意

支持 50Hz 和 60Hz 抗闪烁。

2.2.92.8. 举例

无。

2.2.93 MI_ISP_AE_GetFlicker

2.2.93.1. 目的

取得抗闪烁参数值。

2.2.93.2. 语法

MI_S32 MI_ISP_AE_GetFlicker(MI_U32 Channel, [MI_ISP_AE_FLICKER_TYPE_e](#) *data);

2.2.93.3. 描述

调用此函数取得抗闪烁参数值，支持 50Hz 和 60Hz 抗闪烁。

2.2.93.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	抗闪烁参数值的指针。

2.2.93.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.93.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.93.7. 注意

无。

2.2.93.8. 举例

无。

2.2.94 MI_ISP_AE_SetStrategy

2.2.94.1. 目的

设定曝光策略参数值。

2.2.94.2. 语法

MI_S32 MI_ISP_AE_SetStrategy(MI_U32 Channel, [MI_ISP_AE_STRATEGY_TYPE_t](#) *data);

2.2.94.3. 描述

调用此函式设定曝光策略参数值。

2.2.94.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	曝光策略参数值的指针。

2.2.94.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.94.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.94.7. 注意

无。

2.2.94.8. 举例

无。

2.2.95 MI_ISP_AE_GetStrategy

2.2.95.1. 目的

取得曝光策略参数值。

2.2.95.2. 语法

MI_S32 MI_ISP_AE_GetStrategy(MI_U32 Channel, [MI_ISP_AE_STRATEGY_TYPE_t](#)*data);

2.2.95.3. 描述

调用此函式取得曝光策略参数值。

2.2.95.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	曝光策略参数值的指针。

2.2.95.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.95.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.95.7. 注意

无。

2.2.95.8. 举例

无。

2.2.96 MI_ISP_AWB_QueryInfo

2.2.96.1. 目的

取得自动白平衡参数值。

2.2.96.2. 语法

MI_S32 MI_ISP_AWB_QueryInfo(MI_U32 Channel, [MI_ISP_AWB_QUERY_INFO_TYPE_t](#) *data);

2.2.96.3. 描述

调用此函式取得自动白平衡参数值。

2.2.96.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动白平衡参数值的指针。

2.2.96.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.96.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.96.7. 注意

无。

2.2.96.8. 举例

无。

2.2.97 MI_ISP_AWB_GetCTStats

2.2.97.1. 目的

取得自动白平衡校正统计值。

2.2.97.2. 语法

MI_S32 MI_ISP_AWB_GetCTStats(MI_U32 Channel, [MI_ISP_AWB_CT_STATISTICS_TYPE](#) t *data);

2.2.97.3. 描述

调用此函数取得自动白平衡校正统计值。

2.2.97.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动白平衡校正统计值的指针。

2.2.97.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.97.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.97.7. 注意

无。

2.2.97.8. 举例

无。

2.2.98 MI_ISP_AWB_GetHWStats

2.2.98.1. 目的

取得自动白平衡硬件统计值。

2.2.98.2. 语法

MI_S32 MI_ISP_AWB_GetHWStats(MI_U32 Channel, [MI_ISP_AWB_HW_STATISTICS_TYPE_t](#) *data);

2.2.98.3. 描述

调用此函式取得自动白平衡硬件统计值。

2.2.98.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动白平衡硬件统计值的指针。

2.2.98.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.98.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.98.7. 注意

无。

2.2.98.8. 举例

无。

2.2.99 MI_ISP_AWB_SetAttr

2.2.99.1. 目的

设定自动白平衡属性参数值。

2.2.99.2. 语法

MI_S32 MI_ISP_AWB_SetAttr(MI_U32 Channel, [MI_ISP_AWB_ATTR_TYPE_t](#) *data);

2.2.99.3. 描述

调用此函式设定自动白平衡属性参数值。

2.2.99.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动白平衡属性参数值的指针。

2.2.99.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.99.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.99.7. 注意

无。

2.2.99.8. 举例

无。

2.2.100 MI_ISP_AWB_GetAttr

2.2.100.1. 目的

取得自动白平衡属性参数值。

2.2.100.2. 语法

MI_S32 MI_ISP_AWB_GetAttr(MI_U32 Channel, [MI_ISP_AWB_ATTR_TYPE_t](#) *data);

2.2.100.3. 描述

调用此函式取得自动白平衡属性参数值。

2.2.100.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动白平衡属性参数值的指针。

2.2.100.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.100.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.100.7. 注意

无。

2.2.100.8. 举例

无。

2.2.101 MI_ISP_AWB_SetAttrEx

2.2.101.1. 目的

设定自动白平衡增强属性参数值。

2.2.101.2. 语法

MI_S32 MI_ISP_AWB_SetAttrEx(MI_U32 Channel, [MI_ISP_AWB_ATTR_EX_TYPE_t](#) *data);

2.2.101.3. 描述

调用此函式设定自动白平衡增强属性参数值。

2.2.101.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动白平衡增强属性参数值的指针。

2.2.101.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.101.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.101.7. 注意

只有当 AWB_ATTR_PARAM_t 中的 eAlgType = SS_AWB_ALG_ADVANCE 时，此函式设定的参数才会生效。

2.2.101.8. 举例

无。

2.2.102 MI_ISP_AWB_GetAttrEx

2.2.102.1. 目的

取得自动白平衡增强属性参数值。

2.2.102.2. 语法

MI_S32 MI_ISP_AWB_GetAttrEx(MI_U32 Channel, [MI_ISP_AWB_ATTR_EX_TYPE_t](#) *data);

2.2.102.3. 描述

调用此函式取得自动白平衡增强属性参数值。

2.2.102.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动白平衡增强属性参数值的指针。

2.2.102.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.102.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.102.7. 注意

无。

2.2.102.8. 举例

无。

2.2.103 MI_ISP_AWB_SetMultiLSAttr

2.2.103.1. 目的

设定自动白平衡混光偏校正参数值。

2.2.103.2. 语法

MI_S32 MI_ISP_AWB_SetMultiLSAttr(MI_U32 Channel, [MI_ISP_AWB_MULTILS_LS_TYPE_t](#) *data);

2.2.103.3. 描述

调用此函式设定自动白平衡混光偏校正参数值。

2.2.103.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动白平衡混光偏校正参数值的指针。

2.2.103.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.103.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.103.7. 注意

只有当 AWB_ATTR_PARAM_t 中的 bWpWeightEnable = 1 时，，此函式设定的参数才会生效。

2.2.103.8. 举例

无。

2.2.104 MI_ISP_AWB_GetMultiLSAttr

2.2.104.1. 目的

取得自动白平衡混光偏校正参数值。

2.2.104.2. 语法

MI_S32 MI_ISP_AWB_GetMultiLSAttr(MI_U32 Channel, [MI_ISP_AWB_MULTILS_LS_TYPE_t](#) *data);

2.2.104.3. 描述

调用此函式取得自动白平衡混光偏校正参数值。

2.2.104.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动白平衡混光偏校正参数值的指针。

2.2.104.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.104.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.104.7. 注意

无。

2.2.104.8. 举例

无。

2.2.105 MI_ISP_AWB_SetCTCaliAttr

2.2.105.1. 目的

设定自动白平衡校正参数值。

2.2.105.2. 语法

MI_S32 MI_ISP_AWB_SetCTCaliAttr(MI_U32 Channel, [MI_ISP_AWB_CT_CALI_TYPE_t](#) *data);

2.2.105.3. 描述

调用此函式设定自动白平衡校正参数值。

2.2.105.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动白平衡校正参数值的指针。

2.2.105.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.105.6. 需求

- 头文件: mi_isp.h
- 函式库: libmi_isp.so

2.2.105.7. 注意

无。

2.2.105.8. 举例

无。

2.2.106 MI_ISP_AWB_GetCTCaliAttr

2.2.106.1. 目的

取得自动白平衡校正参数值。

2.2.106.2. 语法

MI_S32 MI_ISP_AWB_GetCTCaliAttr(MI_U32 Channel, [MI_ISP_AWB_CT_CALI_TYPE_t](#) *data);

2.2.106.3. 描述

调用此函式取得自动白平衡校正参数值。

2.2.106.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动白平衡校正参数值的指针。

2.2.106.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.106.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.106.7. 注意

无。

2.2.106.8. 举例

无。

2.2.107 MI_ISP_AF_QueryInfo

2.2.107.1. 目的

取得自动对焦参数值。

2.2.107.2. 语法

MI_S32 MI_ISP_AF_QueryInfo(MI_U32 Channel, [MI_ISP_AF_QUERY_INFO_TYPE_t](#) *data);

2.2.107.3. 描述

调用此函式取得自动对焦参数值。

2.2.107.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动对焦参数值的指针。

2.2.107.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.107.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.107.7. 注意

无。

2.2.107.8. 举例

无。

2.2.108 MI_ISP_AF_SetWindow

2.2.108.1. 目的

设定自动对焦窗口参数值。

2.2.108.2. 语法

MI_S32 MI_ISP_AF_SetWindow(MI_U32 Channel, [MI_ISP_AF_WINDOW_TYPE_t](#) *data);

2.2.108.3. 描述

调用此函式设定自动对焦窗口参数值。

2.2.108.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动对焦窗口参数值的指针。

2.2.108.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.108.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.108.7. 注意

无。

2.2.108.8. 举例

无。

2.2.109 MI_ISP_AF_GetWindow

2.2.109.1. 目的

取得自动对焦窗口参数值。

2.2.109.2. 语法

MI_S32 MI_ISP_AF_GetWindow(MI_U32 Channel, [MI_ISP_AF_WINDOW_TYPE_t](#) *data);

2.2.109.3. 描述

调用此函数取得自动对焦窗口参数值。

2.2.109.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动对焦窗口参数值的指针。

2.2.109.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.109.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.109.7. 注意

无。

2.2.109.8. 举例

无。

2.2.110 MI_ISP_AF_SetFilter

2.2.110.1. 目的

设定 ISP 自动对焦硬件统计值的 IIR/FIR 系数。

2.2.110.2. 语法

MI_S32 MI_ISP_AF_SetFilter(MI_U32 Channel, [MI_ISP_AF_FILTER_TYPE_t](#) *data);

2.2.110.3. 描述

调用此函数设定 ISP 自动对焦硬件统计值的 IIR/FIR 系数。

2.2.110.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动对焦硬件统计值的指针。

2.2.110.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.110.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.110.7. 注意

无。

2.2.110.8. 举例

无。

2.2.111 MI_ISP_AF_GetFilter

2.2.111.1. 目的

取得 ISP 自动对焦硬件统计值的 IIR/FIR 系数。

2.2.111.2. 语法

MI_S32 MI_ISP_AF_GetFilter(MI_U32 Channel, [MI_ISP_AF_FILTER_TYPE_t](#) *data);

2.2.111.3. 描述

调用此函数取得 ISP 自动对焦硬件统计值的 IIR/FIR 系数。

2.2.111.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
*data	自动对焦硬件统计值的指针。

2.2.111.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.111.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.111.7. 注意

无。

2.2.111.8. 举例

无。

2.2.112 MI_ISP_API_CmdLoadBinFile

2.2.112.1. 目的

载入 cmd load bin file, 于使用者在 IQ Tool 调校好的 IQ 参数后 tunnel 到客户应用程序中。

2.2.112.2. 语法

```
MI_S32 MI_ISP_API_CmdLoadBinFile(MI_U32 Channel, char* filepath, MI_U32 user_key);
```

2.2.112.3. 描述

调用此函式载入 IQ Tool 调校好之后的 api cmd bin 档案, 那么客户在 app 中直接 load 此 bin file, 就与 IQ Tool 调校效果一样。

2.2.112.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
filepath	ISP api bin 档案的路径位置。
user_key	user key 与 magic key 必须一致。magic key 可由 IQ Tool 内的 Setting 字段来修改值。

2.2.112.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.112.6. 需求

- 头文件: mi_isp.h
- 函式庫: libmi_isp.so

2.2.112.7. 注意

无。

2.2.112.8. 举例

Filepath 是 bin 文件的路径, 此 user_key 值会去比对 magic key, 而 IQ Tool 保存 magic key 至 ISP api bin 里面, 透过函数调用 ISP api bin 时会判断 ISP api bin 里面的 magic key 与函式传进去的 user_key 是否一致。magic key 值可以在工具的 Setting 里面设定, 如 Figure 2。

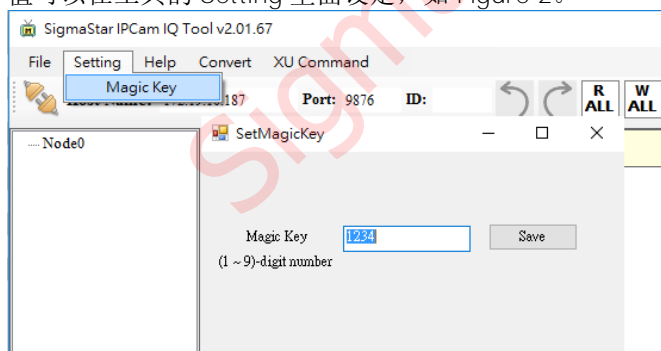


Figure 2: IQTool 接口设定 Magic key 示意图

2.2.113 MI_ISP_API_CmdLoadCaliData

2.2.113.1. 目的

载入 offline calibration data file, 于用户在脱机版校正程序的参数档案(*.data)后 tunnel 到客户应用程序中呼叫此 API function。

2.2.113.2. 语法

MI_S32 MI_ISP_API_CmdLoadCaliData(MI_U32 Channel, [MI_ISP_CALI_ITEM_e](#) eCaliItem, char* filepath);

2.2.113.3. 描述

调用此函数加载脱机版校正程序的参数档案(*.data)，那么客户在应用程序中直接 load 此 data file。

2.2.113.4. 参数

参数名称	描述
Channel	影像输入的信道号(一般为 0)。
eCaliItem	校正项目的列举结构体。
filepath	校正参数档案的路径位置。

2.2.113.5. 返回值

参数名称	描述
MI_ISP_OK	成功。
MI_ISP_FAILURE	失败。

2.2.113.6. 需求

- 头文件：mi_isp.h
- 函式庫：libmi_isp.so

2.2.113.7. 注意

无。

2.2.113.8. 举例

无。

3. MI ISP API 数据类型

3.1. MI ISP API 结构类型详细描述

3.1.1 MI_ISP_BOOL 结构体

3.1.1.1. 枚举结构体 (Enum Structure)

【说明】

布尔值的枚举结构体。

【定义】

```
typedef enum
{
    SS_FALSE = 0,
    SS_TRUE = ISS_FALSE,
    SS_BOOL_MAX
} MI_ISP_BOOL_e;
```

【名称】

变量名称	描述
SS_FALSE	布尔值等于 0。
SS_TRUE	布尔值等于 1。
SS_BOOL_MAX	判断布尔值枚举列表的最大值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.2 MI_ISP_OP_TYPE 结构体

3.1.2.1. 枚举结构体 (Enum Structure)

【说明】

工作模式的枚举结构体。

【定义】

```
typedef enum
{
    SS_OP_TYP_AUTO = 0,
    SS_OP_TYP_MANUAL = ISS_OP_TYP_AUTO,
    SS_OP_TYP_MODE_MAX
} MI_ISP_OP_TYPE_e;
```

【名称】

变量名称	描述
SS_OP_TYP_AUTO	自动工作模式。
SS_OP_TYP_MANUAL	手动工作模式。
SS_OP_TYP_MODE_MAX	判断工作模式列举列表的最大值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.3 MI_ISP_SM_STATE_TYPE 结构体

3.1.3.1. 列举结构体 (Enum Structure)

【说明】

状态机 (State Machine) 的列举结构体。

【定义】

```
typedef enum
{
    SS_ISP_STATE_NORMAL = 0,
    SS_ISP_STATE_PAUSE = 1,
    SS_ISP_STATE_MAX
} MI_ISP_SM_STATE_TYPE_e;
```

【名称】

变量名称	描述
SS_OP_TYP_AUTO	正常模式。
SS_OP_TYP_MANUAL	暂停模式。
SS_OP_TYP_MODE_MAX	判断状态机列举清单的最大值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.4 MI_ISP_IQ_PARAM_MODE 结构体

3.1.4.1. 枚举结构体 (Enum Structure)

【说明】

IQ 特殊模式的枚举结构体。

【定义】

```
typedef enum
{
    SS_IQ_PARAM_MODE_NORMAL = 0,
    SS_IQ_PARAM_MODE_NIGHT = 1,
    SS_IQ_PARAM_MODE_MAX
} MI_ISP_IQ_PARAM_MODE_e;
```

【名称】

变量名称	描述
SS_IQ_PARAM_MODE_NORMAL	IQ 白天工作模式。
SS_IQ_PARAM_MODE_NIGHT	IQ 夜间工作模式。
SS_IQ_PARAM_MODE_MAX	判断 IQ 工作模式枚举列表的最大值。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IQ_SetIQMode](#)(MI_U32 Channel, [MI_ISP_IQ_PARAM_MODE_e](#) *data);
MI_S32 [MI_ISP_IQ_GetIQMode](#)(MI_U32 Channel, [MI_ISP_IQ_PARAM_MODE_e](#) *data);。

3.1.5 MI_ISP_BYPASS_MODE 结构体

3.1.5.1. 枚举结构体 (Enum Structure)

【说明】

ByPass 工作模式的枚举结构体。

【定义】

```
typedef enum
{
    SS_BYPASS_OFF = 0,
    SS_BYPASS_ON,
    SS_BYPASS_MAX
} MI_ISP_BYPASS_MODE_e;
```

【名称】

变量名称	描述
SS_BYPASS_OFF	ByPass 工作模式等于 Disable。
SS_BYPASS_ON	ByPass 工作模式等于 Enable。
SS_BYPASS_MAX	判断 ByPass 工作模式列举列表的最大值。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IQ_SetAPIBypassMode](#)(MI_U32 Channel, [MI_ISP_API_BYPASS_TYPE_t](#) *data);。

MI_S32 [MI_ISP_IQ_GetAPIBypassMode](#)(MI_U32 Channel, [MI_ISP_API_BYPASS_TYPE_t](#) *data);。

3.1.6 MI_ISP_IQ_INDEX 结构体

3.1.6.1. 列举结构体 (Enum Structure)

【说明】

IQ 曝光指数 (ISO index) 的列举结构体。

【定义】

```
typedef enum
{
    SS_IQ_INDEX_0 = 0,
    SS_IQ_INDEX_1,
    SS_IQ_INDEX_2,
    SS_IQ_INDEX_3,
    SS_IQ_INDEX_4,
    SS_IQ_INDEX_5,
    SS_IQ_INDEX_6,
    SS_IQ_INDEX_7,
    SS_IQ_INDEX_8,
    SS_IQ_INDEX_9,
    SS_IQ_INDEX_10,
    SS_IQ_INDEX_11,
    SS_IQ_INDEX_12,
```

```

    SS_IQ_INDEX_13,
    SS_IQ_INDEX_14,
    SS_IQ_INDEX_15,
    SS_IQ_INDEX_MAX
} MI_ISP_IQ_INDEX_e;

```

【名称】

变量名称	描述
SS_IQ_INDEX_0	SS_IQ_INDEX_0 曝光指数等于 0、增益数值等于 1 倍。
SS_IQ_INDEX_1	SS_IQ_INDEX_1 曝光指数等于 1、增益数值等于 2 倍。
SS_IQ_INDEX_2	SS_IQ_INDEX_2 曝光指数等于 2、增益数值等于 4 倍。
SS_IQ_INDEX_3	SS_IQ_INDEX_3 曝光指数等于 3、增益数值等于 8 倍。
SS_IQ_INDEX_4	SS_IQ_INDEX_4 曝光指数等于 4、增益数值等于 16 倍。
SS_IQ_INDEX_5	SS_IQ_INDEX_5 曝光指数等于 5、增益数值等于 32 倍。
SS_IQ_INDEX_6	SS_IQ_INDEX_6 曝光指数等于 6、增益数值等于 64 倍。
SS_IQ_INDEX_7	SS_IQ_INDEX_7 曝光指数等于 7、增益数值等于 128 倍。
SS_IQ_INDEX_8	SS_IQ_INDEX_8 曝光指数等于 8、增益数值等于 256 倍。
SS_IQ_INDEX_9	SS_IQ_INDEX_9 曝光指数等于 9、增益数值等于 512 倍。
SS_IQ_INDEX_10	SS_IQ_INDEX_10 曝光指数等于 10、增益数值等于 1024 倍。
SS_IQ_INDEX_11	SS_IQ_INDEX_11 曝光指数等于 11、增益数值等于 2048 倍。
SS_IQ_INDEX_12	SS_IQ_INDEX_12 曝光指数等于 12、增益数值等于 4096 倍。
SS_IQ_INDEX_13	SS_IQ_INDEX_13 曝光指数等于 13、增益数值等于 8192 倍。
SS_IQ_INDEX_14	SS_IQ_INDEX_14 曝光指数等于 14、增益数值等于 16384 倍。
SS_IQ_INDEX_15	SS_IQ_INDEX_15 曝光指数等于 15、增益数值等于 32768 倍。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IQ_GetIQInd](#)(MI_U32 Channel, [MI_ISP_IQ_INDEX_e](#) *data);

3.1.7 MI_ISP_API_ID 结构体

3.1.7.1. 枚举结构体 (Enum Structure)

【说明】

每个 API ID 的枚举结构体。

【定义】

```
typedef enum
```

```
{  
    API20_AUTO = 0,  
    API20_FPN = 1,  
    API20_CCM = 2,  
    API20_SAT = 3,  
    API20_OBC = 4,  
    API20_OBC_P1 = 5,  
    API20_FCOLOR = 6,  
    API20_CR = 7,  
    API20_NR_DeSpike = 8,  
    API20_SHP2 = 9,  
    API20_NR_3D = 10,  
    API20_BRI = 11,  
    API20_LIG = 12,  
    API20_CST = 13,  
    API20_GMA = 14,  
    API20_CTG = 15,  
    API20_NR_NLM = 16,  
    API20_DEFOG = 17,  
    API20_DRC = 18,  
    API20_DPC = 19,  
    API20_HSV = 20,  
    API20_WDR_LOC = 21,  
    API20_RGBIR = 22,  
    API20_YUVGMA = 23,  
    API20_HDR16to12 = 24,  
    API20_COLORTRANS = 25,  
    API20_HDR = 26,  
    API20_EFFECT = 27,  
    API20_PFC = 28,  
    API20_NRChroma = 29,  
    API20_DM = 30,  
    API20_NR_3D_P1 = 31,  
    API20_WDRCurveAdv = 32,  
    API20_MXNR = 33,  
    API20_R2Y = 34,  
    API20_LSC = 35,  
    API20_LSC_CTRL = 36,  
    API20_ALSC = 37,  
    API20_ALSC_CTRL = 38,  
    API20_DPC_CLUSTER = 39,  
    API20_MAX = 40,  
} MI_ISP_API_ID_e;
```


【名称】

无。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IQ_SetAPIBypassMode](#)(MI_U32 Channel, [MI_ISP_API_BYPASS_TYPE_t](#) *data);

MI_S32 [MI_ISP_IQ_GetAPIBypassMode](#)(MI_U32 Channel, [MI_ISP_API_BYPASS_TYPE_t](#) *data);

3.1.8 MI_ISP_AWB_ALG_TYPE 结构体

3.1.8.1. 枚举结构体 (Enum Structure)

【说明】

自动白平衡算法的枚举结构体。

【定义】

```
typedef enum
{
    SS_AWB_ALG_DEFAULT = 0,
    SS_AWB_ALG_ADVANCE = 1,
    SS_AWB_ALG_MAX
} MI_ISP_AWB_ALG_TYPE_e;
```

【名称】

变量名称	描述
SS_AWB_ALG_DEFAULT	默认自动白平衡算法。
SS_AWB_ALG_ADVANCE	增强的自动白平衡算法，可以开启 MI_ISP_AWB_SetAttrEx 的设定功能。
SS_AWB_ALG_MAX	判断自动白平衡算法枚举列表的最大值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.9 MI_ISP_CALI_ITEM 结构体

3.1.9.1. 枚举结构体 (Enum Structure)

【说明】

校正选项的枚举结构体。

【定义】

```
typedef enum
{
    SS_CALI_ITEM_AWB = 0,
    SS_CALI_ITEM_OBC,
    SS_CALI_ITEM_SDC,
    SS_CALI_ITEM_ALSC,
    SS_CALI_ITEM_LSC,
    SS_CALI_ITEM_MAX,
} MI_ISP_CALI_ITEM_e;
```

【名称】

变量名称	描述
SS_CALI_ITEM_AWB	白平衡补偿校正。
SS_CALI_ITEM_OBC	黑电平补偿校正。
SS_CALI_ITEM_SDC	静态坏点补偿校正。
SS_CALI_ITEM_ALSC	镜头补偿校正。
SS_CALI_ITEM_LSC	镜头补偿校正。
SS_CALI_ITEM_MAX	判断校正选项列举列表的最大值。

【注意事项】

无。

【相关结构类型与函数】

MI_S32 [MI_ISP_API_CmdLoadCaliData](#)(MI_U32 Channel, [MI_ISP_CALI_ITEM_e](#) eCaliItem, char* filepath);

3.1.10 MI_ISP_IQ_VERSION_INFO 结构体

3.1.10.1 类型结构体 (Type Structure)

【说明】

取得 ISP IQ 版本号的结构体。

【定义】

```
typedef struct MI_ISP_IQ_VERSION_INFO_TYPE_s
{
    IQ\_VERSION\_INFO\_PARAM\_t stParaAPI;
} MI_ISP_IQ_VERSION_INFO_TYPE_t;
```

【名称】

变量名称	描述
stParaAPI	取得 ISP IQ 版本号的结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IQ_GetVersionInfo](#)(MI_U32 Channel, [MI_ISP_IQ_VERSION_INFO_TYPE_t](#) *data);

3.1.10.2. 参数结构体 (Parameter Structure)

【说明】

取得 IQ 版本号的参数结构体。

【定义】

```
typedef struct IQ_VERSION_INFO_PARAM_s
{
    MI_U32 u32Vendor;
    MI_U32 u32Major;
    MI_U32 u32Minor;
} IQ_VERSION_INFO_PARAM_t;
```

【名称】

变量名称	描述
u32Vendor	ISP IQ Vendor 版本号。
u32Major	ISP IQ Major 版本号。
u32Minor	ISP IQ Minor 版本号。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.11 MI_ISP_IQ_PARAM_INIT_INFO 结构体

3.1.11.1. 类型结构体 (Type Structure)

【说明】

取得 ISP 初始化 ready 状态的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_PARAM_INIT_INFO_TYPE_s
{
    IQ\_PARAM\_INIT\_INFO\_PARAM\_t stParaAPI;
} MI_ISP_IQ_PARAM_INIT_INFO_TYPE_t;
```

【名称】

变量名称	描述
stParaAPI	取得 ISP 初始化 ready 状态的类型结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IQ_GetParaInitStatus](#)(MI_U32 Channel, [MI_ISP_IQ_PARAM_INIT_INFO_TYPE_t](#) *data);

3.1.11.2. 参数结构体 (Parameter Structure)

【说明】

取得 ISP 初始化 ready 状态的参数结构体。

【定义】

```
typedef struct IQ_PARAM_INIT_INFO_PARAM_s
{
    MI\_ISP\_BOOL\_e bFlag;
} IQ_PARAM_INIT_INFO_PARAM_t;
```

【名称】

变量名称	描述
bFlag	取得 ISP 初始化完毕状态的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.12 MI_ISP_IQ_COLORTOGRAY 结构体

3.1.12.1. 类型结构体 (Type Structure)

【说明】

设定彩色转灰阶影像的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_COLORTOGRAY_TYPE_s
{
    MI\_ISP\_BOOL\_e bEnable;
} MI_ISP_IQ_COLORTOGRAY_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定彩色转灰阶影像功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IQ_SetColorToGray](#)(MI_U32 Channel, [MI_ISP_IQ_COLORTOGRAY_TYPE_t](#) *data);
MI_S32 [MI_ISP_IQ_GetColorToGray](#)(MI_U32 Channel, [MI_ISP_IQ_COLORTOGRAY_TYPE_t](#) *data);

3.1.13 MI_ISP_IQ_CONTRAST 结构体

3.1.13.1 类型结构体 (Type Structure)

【说明】

设定对比度的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_CONTRAST_TYPE_s
{
    MI\_ISP\_BOOL\_e          bEnable;
    MI\_ISP\_OP\_TYPE\_e       enOpType;
    CONTRAST\_AUTO\_ATTR\_t   stAuto;
    CONTRAST\_MANUAL\_ATTR\_t stManual;
} MI_ISP_IQ_CONTRAST_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定对比度功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定对比度的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。 手动模式：SS_OP_TYP_MANUAL = 1。
stAuto	设定对比度的自动模式属性结构体。
stManual	设定对比度的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IO_SetContrast](#)(MI_U32 Channel, [MI_ISP_IO_CONTRAST_TYPE_t](#) *data);
 MI_S32 [MI_ISP_IO_GetContrast](#)(MI_U32 Channel, [MI_ISP_IO_CONTRAST_TYPE_t](#) *data);

3.1.13.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定对比度的自动模式属性结构体。

【定义】

```
typedef struct CONTRAST_AUTO_ATTR_s
{
    LEVEL\_BASE\_PARAM\_t stParaAPI[MI_ISP_AUTO_NUM];
} CONTRAST_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.13.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定对比度的手动模式属性结构体。

【定义】

```
typedef struct CONTRAST_MANUAL_ATTR_s
{
    LEVEL\_BASE\_PARAM\_t stParaAPI;
} CONTRAST_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.13.4. 参数结构体 (Parameter Structure)**【说明】**

设定对比度、亮度、灰度的参数结构体。

【定义】

```
typedef struct LEVEL_BASE_PARAM_s
{
    MI_U32 u32Lev;
} LEVEL_BASE_PARAM_t;
```

【名称】

变量名称	描述
u32Lev	设定对比度、亮度、灰度的可变强度数值。值域范围：0 ~ 100。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.14 MI_ISP_IQ_BRIGHTNESS 结构体**3.1.14.1. 类型结构体 (Type Structure)****【说明】**

设定亮度的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_BRIGHTNESS_TYPE_s
{
    MI_ISP_BOOL_e          bEnable;
    MI_ISP_OP_TYPE_e       enOpType;
    BRIGHTNESS_AUTO_ATTR_t stAuto;
    BRIGHTNESS_MANUAL_ATTR_t stManual;
} MI_ISP_IQ_BRIGHTNESS_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定亮度功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定亮度的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。 手动模式：SS_OP_TYP_MANUAL = 1。
stAuto	设定亮度的自动模式属性结构体。
stManual	设定亮度的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IO_SetBrightness](#)(MI_U32 Channel, [MI_ISP_IO_BRIGHTNESS_TYPE_t](#) *data);
MI_S32 [MI_ISP_IO_GetBrightness](#)(MI_U32 Channel, [MI_ISP_IO_BRIGHTNESS_TYPE_t](#) *data);

3.1.14.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定亮度的自动模式属性结构体。

【定义】

```
typedef struct BRIGHTNESS_AUTO_ATTR_s
{
    LEVEL\_BASE\_PARAM\_t stParaAPI[MI_ISP_AUTO_NUM];
} BRIGHTNESS_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.14.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定亮度的手动模式属性结构体。

【定义】

```
typedef struct BRIGHTNESS_MANUAL_ATTR_s
{
    LEVEL\_BASE\_PARAM\_t stParaAPI;
} BRIGHTNESS_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.15 MI_ISP_IQ_LIGHTNESS 结构体

3.1.15.1. 类型结构体 (Type Structure)

【说明】

设定灰度的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_LIGHTNESS_TYPE_s
{
    MI\_ISP\_BOOL\_e bEnable;
    MI\_ISP\_OP\_TYPE\_e enOpType;
    LIGHTNESS\_AUTO\_ATTR\_t stAuto;
    LIGHTNESS\_MANUAL\_ATTR\_t stManual;
} MI_ISP_IQ_LIGHTNESS_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定灰度功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定灰度的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。

变量名称	描述
	手动模式：SS_OP_TYP_MANUAL = 1。
stAuto	设定灰度的自动模式属性结构体。
stManual	设定灰度的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IO_SetLightness](#)(MI_U32 Channel, [MI_ISP_IO_LIGHTNESS_TYPE_t](#) *data);
 MI_S32 [MI_ISP_IO_GetLightness](#)(MI_U32 Channel, [MI_ISP_IO_LIGHTNESS_TYPE_t](#) *data);

3.1.15.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定灰度的自动模式属性结构体。

【定义】

```
typedef struct LIGHTNESS_AUTO_ATTR_s
{
    LEVEL\_BASE\_PARAM\_t stParaAPI[MI_ISP_AUTO_NUM];
} LIGHTNESS_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.15.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定灰度的手动模式属性结构体。

【定义】

```
typedef struct LIGHTNESS_MANUAL_ATTR_s
{
    LEVEL\_BASE\_PARAM\_t stParaAPI;
} LIGHTNESS_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.16 MI_ISP_IQ_RGBGAMMA 结构体

3.1.16.1. 类型结构体 (Type Structure)

【说明】

设定 RGB Gamma curve 的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_RGBGAMMA_TYPE_s
{
    MI\_ISP\_BOOL\_e          bEnable;
    MI\_ISP\_OP\_TYPE\_e       enOpType;
    RGBGAMMA\_AUTO\_ATTR\_t   stAuto;
    RGBGAMMA\_MANUAL\_ATTR\_t stManual;
} MI_ISP_IQ_RGBGAMMA_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定 RGB Gamma curve 功能的布尔值。 关闭: SS_FALSE = 0。 开启: SS_TRUE = 1。
enOpType	设定 RGB Gamma curve 的工作模式。 自动模式: SS_OP_TYP_AUTO = 0。 手动模式: SS_OP_TYP_MANUAL = 1。
stAuto	设定 RGB Gamma curve 的自动模式属性结构体。
stManual	设定 RGB Gamma curve 的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IQ_SetRGBGamma](#)(MI_U32 Channel, [MI_ISP_IQ_RGBGAMMA_TYPE_t](#) *data);
MI_S32 [MI_ISP_IQ_GetRGBGamma](#)(MI_U32 Channel, [MI_ISP_IQ_RGBGAMMA_TYPE_t](#) *data);

3.1.16.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定 RGB Gamma curve 的自动模式属性结构体。

【定义】

```
typedef struct RGBGAMMA_AUTO_ATTR_s
{
    RGBGAMMA\_PARAM\_t stParaAPI[MI_ISP_AUTO_NUM];
} RGBGAMMA_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.16.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定 RGB Gamma curve 的手动模式属性结构体。

【定义】

```
typedef struct RGBGAMMA_MANUAL_ATTR_s
{
    RGBGAMMA\_PARAM\_t stParaAPI;
} RGBGAMMA_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.16.4. 参数结构体 (Parameter Structure)

【说明】

设定 RGB Gamma curve 的参数结构体。

【定义】

```
typedef struct RGBGAMMA_PARAM_s
{
    MI_U16 u16LutR[256];
    MI_U16 u16LutG[256];
    MI_U16 u16LutB[256];
} RGBGAMMA_PARAM_t;
```

【名称】

变量名称	描述
u16LutR[256]	R 通道分量。值域范围：0 ~ 1023。
u16LutG[256]	G 通道分量。值域范围：0 ~ 1023。
u16LutB[256]	B 通道分量。值域范围：0 ~ 1023。

【注意事项】

- 支援三個通道的 Gamma 設定，SigmaStar 建議三個通道設定一樣，如 Figure 3。



Figure 3: RGB Gamma curve

【相关结构类型与函式】

无。

3.1.17 MI_ISP_IQ_YUVGAMMA 结构体

3.1.17.1. 类型结构体 (Type Structure)

【说明】

设定 YUV Gamma curve 的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_YUVGAMMA_TYPE_s
{
    MI_ISP_BOOL_e          bEnable;
    MI_ISP_OP_TYPE_e       enOpType;
    YUVGAMMA_AUTO_ATTR_t   stAuto;
    YUVGAMMA_MANUAL_ATTR_t stManual;
} MI_ISP_IQ_YUVGAMMA_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定 YUV Gamma curve 功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定 YUV Gamma curve 的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。 手动模式：SS_OP_TYP_MANUAL = 1。
stAuto	设定 YUV Gamma curve 的自动模式属性结构体。
stManual	设定 YUV Gamma curve 的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 MI_ISP_IQ_SetYUVGamma(MI_U32 Channel, MI_ISP_IQ_YUVGAMMA_TYPE_t *data);
MI_S32 MI_ISP_IQ_GetYUVGamma(MI_U32 Channel, MI_ISP_IQ_YUVGAMMA_TYPE_t *data);

3.1.17.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定 YUV Gamma curve 的自动模式属性结构体。

【定义】

```
typedef struct YUVGAMMA_AUTO_ATTR_s
{
    YUVGAMMA\_PAMAR\_t stParaAPI[MI_ISP_AUTO_NUM];
} YUVGAMMA_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.17.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定 YUV Gamma curve 的手动模式属性结构体。

【定义】

```
typedef struct YUVGAMMA_MANUAL_ATTR_s
{
    YUVGAMMA\_PAMAR\_t stParaAPI;
} YUVGAMMA_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.17.4. 参数结构体 (Parameter Structure)

【说明】

设定 YUV Gamma curve 的参数结构体。

【定义】

```
typedef struct YUVGAMMA_PAMAR_s
{
    MI_U16 u16LutY[256];
    MI_U16 u16LutU[128];
    MI_U16 u16LutV[128];
} YUVGAMMA_PAMAR_t;
```

【名称】

变量名称	描述
u16LutY[256]	Y 通道分量。值域范围：0 ~ 1023。
u16LutU[128]	U 通道分量。值域范围：0 ~ 511。
u16LutV[128]	V 通道分量。值域范围：0 ~ 511。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.18 MI_ISP_IQ_SATURATION 结构体

3.1.18.1. 类型结构体 (Type Structure)

【说明】

设定饱和度的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_SATURATION_TYPE_s
{
    MI_ISP_BOOL_e          bEnable;
    MI_ISP_OP_TYPE_e       enOpType;
    SATURATION_AUTO_ATTR_t stAuto;
    SATURATION_MANUAL_ATTR_t stManual;
} MI_ISP_IQ_SATURATION_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定饱和度功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定饱和度的工作模式。

变量名称	描述
	自动模式: SS_OP_TYP_AUTO = 0。 手动模式: SS_OP_TYP_MANUAL = 1。
stAuto	设定饱和度的自动模式属性结构体。
stManual	设定饱和度的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IQ_SetSaturation](#)(MI_U32 Channel, [MI_ISP_IQ_SATURATION_TYPE_t](#) *data);
MI_S32 [MI_ISP_IQ_GetSaturation](#)(MI_U32 Channel, [MI_ISP_IQ_SATURATION_TYPE_t](#) *data);

3.1.18.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)**【说明】**

设定饱和度的自动模式属性结构体。

【定义】

```
typedef struct SATURATION_AUTO_ATTR_s
{
    SATURATION\_PARAM\_t stParaAPI[MI_ISP_AUTO_NUM];
} SATURATION_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体, MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.18.3. 手动模式属性结构体 (Manual Mode Attribute Structure)**【说明】**

设定饱和度的手动模式属性结构体。

【定义】

```
typedef struct SATURATION_MANUAL_ATTR_s
{
    SATURATION_PARAM_t stParaAPI;
} SATURATION_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.18.4. 参数结构体 (Parameter Structure)

【说明】

设定饱和度的参数结构体。

【定义】

```
typedef struct SATURATION_PARAM_s
{
    MI_U8 u8SatAllStr;
    MI_U8 u8SatByYSFTAdv[5];
    MI_U8 u8SatByYLUTAdv[6];
    MI_U8 u8SatBySSFTAdv[5];
    MI_U8 u8SatBySLUTAdv[6];
    MI_U8 u8SatCoring;
} SATURATION_PARAM_t;
```

【名称】

变量名称	描述
u8SatAllStr	整体饱和度的强度数值。值域范围：0 ~ 127 (32 等于 1 倍)。
u8SatByYSFTAdv[5]	调整亮度区间的强度数值。值域范围：0 ~ 8。
u8SatByYLUTAdv[6]	基于亮度区间调整饱和度的数值。值域范围：0 ~ 128。
u8SatBySSFTAdv[5]	调整彩度区间的强度数值。值域范围：0 ~ 8。
u8SatBySLUTAdv[6]	基于彩度区间调整饱和度的数值。值域范围：0 ~ 128。
u8SatCoring	彩度全部减一个值降低彩度。值域范围：0 ~ 255。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.19 MI_ISP_IQ_RGBMATRIX 结构体

3.1.19.1. 类型结构体 (Type Structure)

【说明】

设定色彩校正矩阵的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_RGBMATRIX_TYPE_s
{
    MI_ISP_BOOL_e          bEnable;
    MI_ISP_OP_TYPE_e       enOpType;
    RGBMATRIX_AUTO_ATTR_t  stAuto;
    RGBMATRIX_MANUAL_ATTR_t stManual;
} MI_ISP_IQ_RGBMATRIX_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定色彩校正矩阵功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定色彩校正矩阵的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。 手动模式：SS_OP_TYP_MANUAL = 1。
stAuto	设定色彩校正矩阵的自动模式属性结构体。
stManual	设定色彩校正矩阵的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 MI_ISP_IQ_SetRGBMatrix(MI_U32 Channel, MI_ISP_IQ_RGBMATRIX_TYPE_t *data);
MI_S32 MI_ISP_IQ_GetRGBMatrix(MI_U32 Channel, MI_ISP_IQ_RGBMATRIX_TYPE_t *data);

3.1.19.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定色彩校正矩阵的自动模式属性结构体。

【定义】

```
typedef struct RGBMATRIX_AUTO_ATTR_s
{
    MI_ISP_BOOL_e      bISOActEn;
    RGBMATRIX_PARAM_t  stParaAPI[MI_ISP_AUTO_NUM];
} RGBMATRIX_AUTO_ATTR_t;
```

【名称】

变量名称	描述
bISOActEn	设定 ISO 功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.19.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定色彩校正矩阵的手动模式属性结构体。

【定义】

```
typedef struct RGBMATRIX_MANUAL_ATTR_s
{
    MI_U16    u16CCM[9];
    MI_U8     u8CCMSat;
} RGBMATRIX_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
u16CCM[9]	色彩校正矩阵参数值。值域范围：0 ~ 8191。
u8CCMSat	透过色彩校正矩阵调整饱和度的数值强度。值域范围：0 ~ 100。

【注意事项】

- u8CCMSat = 0 表示為色彩校正的單位矩陣數值。
u8CCMSat = 100 表示为色彩校正的矩阵数值。

【相关结构类型与函式】

无。

3.1.19.4. 参数结构体 (Parameter Structure)

【说明】

设定色彩校正矩阵的参数结构体。

【定义】

```
typedef struct RGBMATRIX_PARAM_s
{
    MI_U16    u16CCTthr;
    MI_U16    u16CCM[9];
    MI_U8     u8CCMSat;
} RGBMATRIX_PARAM_t;
```

【名称】

变量名称	描述
u16CCTthr	该组 CCM 的代表色温。值域范围：0 ~ 10000。
u16CCM[9]	色彩校正矩阵参数值。值域范围：0 ~ 8191。
u8CCMSat	透过色彩校正矩阵调整饱和度的数值强度。值域范围：0 ~ 100。

【注意事项】

- u8CCMSat = 0 表示為色彩校正的單位矩陣數值。
u8CCMSat = 100 表示為色彩校正的矩陣數值。

【相关结构类型与函式】

无。

3.1.20 MI_ISP_IQ_FALSECOLOR 结构体

3.1.20.1. 类型结构体 (Type Structure)

【说明】

设定去伪彩的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_FALSECOLOR_TYPE_s
{
    MI_ISP_BOOL_e          bEnable;
    MI_ISP_OP_TYPE_e       enOpType;
    FALSECOLOR_AUTO_ATTR_t stAuto;
    FALSECOLOR_MANUAL_ATTR_t stManual;
} MI_ISP_IQ_FALSECOLOR_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定去伪彩功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定去伪彩的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。 手动模式：SS_OP_TYP_MANUAL = 1。
stAuto	设定去伪彩的自动模式属性结构体。
stManual	设定去伪彩的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

```
MI_S32 MI_ISP_IQ_SetFalseColor(MI_U32 Channel, MI_ISP_IQ_FALSECOLOR_TYPE_t *data);
MI_S32 MI_ISP_IQ_GetFalseColor(MI_U32 Channel, MI_ISP_IQ_FALSECOLOR_TYPE_t *data);
```

3.1.20.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定去伪彩的自动模式属性结构体。

【定义】

```
typedef struct FALSECOLOR_AUTO_ATTR_s
{
    FALSECOLOR_PARAM_t stParaAPI[MI_ISP_AUTO_NUM];
} FALSECOLOR_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.20.3. 手动模式属性结构体 (Manual Mode Attribute Structure)**【说明】**

设定去伪彩的手动模式属性结构体。

【定义】

```
typedef struct FALSECOLOR_MANUAL_ATTR_s
{
    FALSECOLOR_PARAM_t stParaAPI;
} FALSECOLOR_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.20.4. 参数结构体 (Parameter Structure)**【说明】**

设定去伪彩的参数结构体。

【定义】

```
typedef struct FALSECOLOR_PARAM_s
{
    MI_U8    u8FreqThrd;
    MI_U8    u8EdgeScoreThrd;
    MI_U8    u8ChromaThrdOfStrengthMax;
    MI_U8    u8ChromaThrdOfStrengthMid;
    MI_U8    u8ChromaThrdOfStrengthMin;
    MI_U8    u8StrengthMid;
    MI_U8    u8StrengthMin;
} FALSECOLOR_PARAM_t;
```

【名称】

变量名称	描述
u8FreqThrd	摩尔纹的判断频率阈值，超过此阈值判断为摩尔纹。值域范围：0 ~ 255，预设：140。
u8EdgeScoreThrd	摩尔纹的判断 Edge 阈值，小于此阈值判断为摩尔纹。值域范围：0 ~ 255，预设：31。
u8ChromaThrdOfStrengthMax	最大强度的阈值。值域范围：0 ~ 127，预设：10。
u8ChromaThrdOfStrengthMid	中间强度的阈值。值域范围：0 ~ 127，预设：40。
u8ChromaThrdOfStrengthMin	最小强度的阈值。值域范围：0 ~ 127，预设：80。
u8StrengthMid	中间强度。值域范围：0 ~ 7，预设：7。
u8StrengthMin	最小强度。值域范围：0 ~ 7，预设：7。

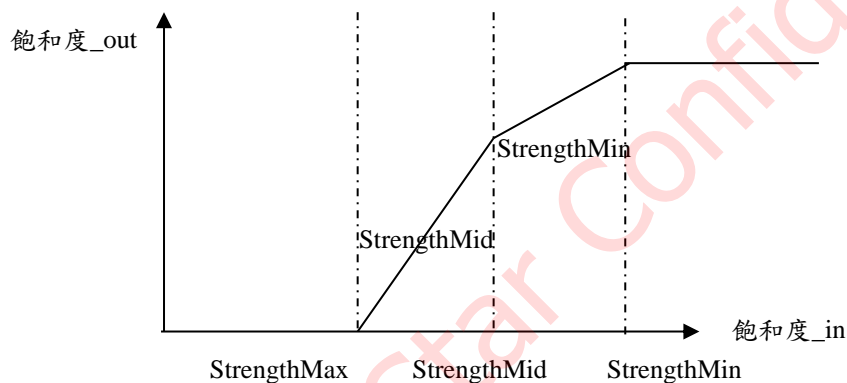


Figure 4: Chroma 强度的阈值限制条件示意图

【相关结构类型与函式】

无。

3.1.21 MI_ISP_IQ_NR3D 结构体

3.1.21.1. 类型结构体 (Type Structure)

【说明】

设定 3D 降噪的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_NR3D_TYPE_s
{
    MI_ISP_BOOL_e          bEnable;
    MI_ISP_OP_TYPE_e       enOpType;
    NR3D_AUTO_ATTR_t       stAuto;
    NR3D_MANUAL_ATTR_t      stManual;
} MI_ISP_IQ_NR3D_TYPE_t;
```


【名称】

变量名称	描述
bEnable	设定 3D 降噪功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定 3D 降噪的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。 手动模式：SS_OP_TYP_MANUAL = 1。
stAuto	设定 3D 降噪的自动模式属性结构体。
stManual	设定 3D 降噪的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IO_SetNR3D](#)(MI_U32 Channel, [MI_ISP_IO_NR3D_TYPE_t](#) *data);
MI_S32 [MI_ISP_IO_GetNR3D](#)(MI_U32 Channel, [MI_ISP_IO_NR3D_TYPE_t](#) *data);

3.1.21.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定 3D 降噪的自动模式属性结构体。

【定义】

```
typedef struct NR3D_AUTO_ATTR_s
{
    NR3D\_PARAM\_t stParaAPI[MI_ISP_AUTO_NUM];
} NR3D_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.21.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定 3D 降噪的手动模式属性结构体。

【定义】

```
typedef struct NR3D_MANUAL_ATTR_s
{
    NR3D\_PARAM\_t stParaAPI;
} NR3D_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.21.4. 参数结构体 (Parameter Structure)

【说明】

设定 3D 降噪的参数结构体。

【定义】

```
typedef struct NR3D_PARAM_s
{
    MI_U16    u16MdThd;
    MI_U16    u16MdGain;
    MI_U8     u8TfStr;
    MI_U8     u8TfStrEx;
    MI_U8     u8MdThdByY[16];
    MI_U8     u8MdGainByY[16];
    MI_U8     u8M2SLut[16];
    MI_U8     u8TfLut[16];
    MI_U8     u8YSfStr;
    MI_U8     u8YSfBlendLut[16];
} NR3D_PARAM_t;
```

【名称】

变量名称	描述
u16MdThd	MD 侦测，控制 motion 阈值，小于此阈值判断为静止物体，值越大 3D 降噪越强。值域范围：0~2048。
u16MdGain	MD 侦测，控制 motion 程度，值越大越容易判断为静止物体，3D 降噪越强。值域范围：0~2048。
u8TfStr	3D 降噪强度控制。值域范围：0~64
u8TfStrEx	3D 降噪强度控制 Ex，开此设定强度会更强，但拖影更明显值域范围：0~64
u8MdThdByY[16]	根据亮度控制 motion 阈值，值越大 3D 降噪越强。值域范围：0~255
u8MdGainByY[16]	根据亮度控制 motion 程度，值越大 3D 降噪越强。值域范围：0~255
u8M2SLut[16]	移动到静止过程中，3D 降噪的收敛速度，值越大，收敛越慢，可用来避免拖影。值域范围：0~31
u8TfLut[16]	根据移动程度，控制 3D 降噪的强度。值域范围：0~63
u8YSfStr	2D 亮度降噪强度控制。值域范围：0~31
u8YSfBlendLut[16]	根据移动程度，控制 2D 降噪的强度。值域范围：0~16

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.22 MI_ISP_IQ_NRDESPIKE 结构体

3.1.22.1. 类型结构体 (Type Structure)

【说明】

设定 DeSpike 降噪的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_NRDESPIKE_TYPE_s
{
    MI_ISP_BOOL_e          bEnable;
    MI_ISP_OP_TYPE_e       enOpType;
    NRDESPIKE_AUTO_ATTR_t  stAuto;
    NRDESPIKE_MANUAL_ATTR_t stManual;
} MI_ISP_IQ_NRDESPIKE_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定 DeSpike 降噪功能的布尔值。 关闭: SS_FALSE = 0。 开启: SS_TRUE = 1。
enOpType	设定 DeSpike 降噪的工作模式。 自动模式: SS_OP_TYP_AUTO = 0。 手动模式: SS_OP_TYP_MANUAL = 1。
stAuto	设定 DeSpike 降噪的自动模式属性结构体。
stManual	设定 DeSpike 降噪的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IO_SetNRDeSpike](#)(MI_U32 Channel, [MI_ISP_IO_NRDESPIKE_TYPE_t](#) *data);
MI_S32 [MI_ISP_IO_GetNRDeSpike](#)(MI_U32 Channel, [MI_ISP_IO_NRDESPIKE_TYPE_t](#) *data);

3.1.22.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定 DeSpike 降噪的自动模式属性结构体。

【定义】

```
typedef struct NRDESPIKE_AUTO_ATTR_s
{
    NRDESPIKE\_PARAM\_t stParaAPI[MI_ISP_AUTO_NUM];
} NRDESPIKE_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体, MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.22.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定 DeSpike 降噪的手动模式属性结构体。

【定义】

```
typedef struct NRDESPIKE_MANUAL_ATTR_s
{
    NRDESPIKE_PARAM_t stParaAPI;
} NRDESPIKE_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.22.4. 参数结构体 (Parameter Structure)

【说明】

设定 DeSpike 降噪的参数结构体。

【定义】

```
typedef struct NRDESPIKE_PARAM_s
{
    MI_U8    u8BlendRatio;
    MI_U8    u8StrengthCenterNeighbor;
    MI_U8    u8StrengthMeanStd;
    MI_U8    u8StrengthCornerCross;
    MI_U8    u8DiffGainMeanStd;
    MI_U16   u16DiffGainCenterNeighbor;
    MI_U16   u16DiffThdCornerCross;
} NRDESPIKE_PARAM_t;
```

【名称】

变量名称	描述
u8BlendRatio	调整 NRDeSpike 去噪整体强度，此数值越大去噪效果越强。 值域范围：0 ~ 15。
u8StrengthMeanStd	调整中心点与周围端点平均差值的强度。 值域范围：0 ~ 5。

变量名称	描述
u8StrengthCenterNeighbor	调整中心点与十字或对角十字端点平均差值的强度。 值域范围：0 ~ 5。
u8StrengthCornerCross	调整对角十字和十字端点平均值相减值的强度。 值域范围：0 ~ 5。
u8DiffGainMeanStd	调整中心点与周围端点平均差值乘上的 Gain 值。 值域范围：0 ~ 31。
u16DiffThdCenterNeighbor	调整中心点与十字或对角十字端点平均差值的 threshold 数值。 值域范围：0 ~ 255。
u16DiffThdCornerCross	调整对角十字和十字端点平均值相减值的 threshold 数值。 值域范围：0 ~ 255。

【注意事项】

- 根據中心點與周圍的端點差異，差異越大，越容易中心點抹除的動作。
當 DiffGain 設定越小、Strength 設定越大，越容易將中心點抹除。

【相关结构类型与函式】

无。

3.1.23 MI_ISP_IQ_NRLUMA 结构体

3.1.23.1. 类型结构体 (Type Structure)

【说明】

设定 Luma 降噪的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_NRLUMA_TYPE_s
{
    MI_ISP_BOOL_e          bEnable;
    MI_ISP_OP_TYPE_e       enOpType;
    NRLUMA_AUTO_ATTR_t     stAuto;
    NRLUMA_MANUAL_ATTR_t   stManual;
} MI_ISP_IQ_NRLUMA_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定 Luma 降噪功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定 Luma 降噪的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。 手动模式：SS_OP_TYP_MANUAL = 1。

变量名称	描述
stAuto	设定 Luma 降噪的自动模式属性结构体。
stManual	设定 Luma 降噪的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IQ_SetNRLuma](#)(MI_U32 Channel, [MI_ISP_IQ_NRLUMA_TYPE_t](#) *data);
MI_S32 [MI_ISP_IQ_GetNRLuma](#)(MI_U32 Channel, [MI_ISP_IQ_NRLUMA_TYPE_t](#) *data);

3.1.23.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定 Luma 降噪的自动模式属性结构体。

【定义】

```
typedef struct NRLUMA_AUTO_ATTR_s  
{  
    NRLUMA\_PARAM\_t stParaAPI[MI_ISP_AUTO_NUM];  
} NRLUMA_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.23.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定 Luma 降噪的手动模式属性结构体。

【定义】

```
typedef struct NRLUMA_MANUAL_ATTR_s
{
    NRLUMA\_PARAM\_t stParaAPI;
} NRLUMA_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.23.4. 参数结构体 (Parameter Structure)

【说明】

设定 Luma 降噪的参数结构体。

【定义】

```
typedef struct NRLUMA_PARAM_s
{
    MI\_ISP\_BOOL\_e bLumaAdjEn;
    MI_U8          u8LumaX[9];
    MI_U8          u16LumaStrByY[10];
    MI_U8          u8Wei[20];
} NRLUMA_PARAM_t;
```

【名称】

变量名称	描述
bLumaAdjEn	NRLuma adjust 开关。
u8LumaX[9]	NRLuma 节点。值域范围：0 ~ 7。限制是 $256 \leq \sum_{i=0}^8 2^{u8LumaX[i]} < 512$ 。
u16LumaStrByY[10]	根据不同亮度控制 NRLuma 的强度，值越大强度越强。值域范围：0 ~ 512。
u8Wei[20]	根据变异控制 NRLuma 的强度，值越大强度越强。值域范围：0 ~ 31。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.24 MI_ISP_IQ_NRChroma 结构体

3.1.24.1. 类型结构体 (Type Structure)

【说明】

设定颜色降噪的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_NRChroma_TYPE_s
{
    MI_ISP_BOOL_e          bEnable;
    MI_ISP_OP_TYPE_e       enOpType;
    NRChroma_AUTO_ATTR_t   stAuto;
    NRChroma_MANUAL_ATTR_t stManual;
} MI_ISP_IQ_NRChroma_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定颜色降噪功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定颜色降噪的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。 手动模式：SS_OP_TYP_MANUAL = 1。
stAuto	设定颜色降噪的自动模式属性结构体。
stManual	设定颜色降噪的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

```
MI_S32 MI_ISP_IQ_SetNRChroma(MI_U32 Channel, MI_ISP_IQ_NRChroma_TYPE_t *data);
MI_S32 MI_ISP_IQ_GetNRChroma(MI_U32 Channel, MI_ISP_IQ_NRChroma_TYPE_t *data);
```

3.1.24.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定颜色降噪的自动模式属性结构体。

【定义】

```
typedef struct NRChroma_AUTO_ATTR_s
{
    NRChroma\_PARAM\_t stParaAPI[MI_ISP_AUTO_NUM];
} NRChroma_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.24.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定颜色降噪的手动模式属性结构体。

【定义】

```
typedef struct NRChroma_MANUAL_ATTR_s
{
    NRChroma\_PARAM\_t stParaAPI;
} NRChroma_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.24.4. 参数结构体 (Parameter Structure)

【说明】

设定颜色降噪的参数结构体。

【定义】

```
typedef struct NRChroma_PARAM_s
{
    MI_U16 u16StrengthByY[8];
    MI_U16 u16StrengthByYEdge[8];
    MI_U16 u16StrengthByCEdge[8];
} NRChroma_PARAM_t;
```

【名称】

变量名称	描述
u8MatchRatio	符合比例阈值。值域范围：0 ~ 31。
u16UvTh	U/V 噪声阈值。值域范围：0 ~ 256。
u16StrengthByCEdge[6]	根据 Color Edge 强度控制颜色降噪的强度。值域范围：0 ~ 511。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.25 MI_ISP_IQ_SHARPNESS 结构体

3.1.25.1 类型结构体 (Type Structure)

【说明】

设定锐度的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_SHARPNESS_TYPE_s
{
    MI_ISP_BOOL_e bEnable;
    MI_ISP_OP_TYPE_e enOpType;
    SHARPNESS_AUTO_ATTR_t stAuto;
    SHARPNESS_MANUAL_ATTR_t stManual;
} MI_ISP_IQ_SHARPNESS_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定锐度功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。

变量名称	描述
enOpType	设定锐度的工作模式。 自动模式: SS_OP_TYP_AUTO = 0。 手动模式: SS_OP_TYP_MANUAL = 1。
stAuto	设定锐度的自动模式属性结构体。
stManual	设定锐度的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IO_SetSharpness](#)(MI_U32 Channel, [MI_ISP_IO_SHARPNESS_TYPE_t](#) *data);
MI_S32 [MI_ISP_IO_GetSharpness](#)(MI_U32 Channel, [MI_ISP_IO_SHARPNESS_TYPE_t](#) *data);

3.1.25.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定锐度的自动模式属性结构体。

【定义】

```
typedef struct SHARPNESS_AUTO_ATTR_s  
{  
    SHARPNESS\_PARAM\_t stParaAPI[MI_ISP_AUTO_NUM];  
} SHARPNESS_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体, MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.25.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定锐度的手动模式属性结构体。

【定义】

```
typedef struct SHARPNESS_MANUAL_ATTR_s  
{  
    SHARPNESS\_PARAM\_t stParaAPI;  
} SHARPNESS_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.25.4. 参数结构体 (Parameter Structure)

【说明】

设定锐度的参数结构体。

【定义】

```
typedef struct SHARPNESS_PARAM_s  
{  
    SHARPNESS\_LOCAL\_PARAM\_t stEdgeCtrl;  
    SHARPNESS\_LOCAL\_PARAM\_t stTextureCtrl;  
    MI_U8 u8EdgeKillLUT[6];  
    MI_U8 u8CornerReduce;  
    MI\_ISP\_BOOL\_e bDirEn;  
    MI_U16 u16SharpnessUD;  
    MI_U16 u16SharpnessD;  
    MI_U16 u16DetailTh;  
    MI_U8 u8DetailByY[6];  
    MI_U16 u16OverShootLimit;  
    MI_U16 u16UndreShootLimit;  
} SHARPNESS_PARAM_t;
```

【名称】

变量名称	描述
stEdgeCtrl	边缘(较细)的 edge 控制参数结构体。
stTextureCtrl	纹理(较粗)的 edge 控制参数结构体。
u8EdgeKillLUT[6]	设定 edge 的强度输出。值域范围：0 ~ 255，默认值：0。

变量名称	描述
u8CornerReduce	调整周围 edge 的锐化程度。值域范围：0 ~ 32，默认值：32。
bDirEn	设定强化边缘算法回去参考方向性功能的布尔值。 关闭：SS_FALSE = 0。表示不参考，只要是边缘都会增强。 开启：SS_TRUE = 1。
u16SharpnessUD	增强无方向性的细节纹理，可用于提高发丝、草地...等细小纹理清晰度。值域范围：0 ~ 1023。
u16SharpnessD	根据边缘方向强化锐化度，整体增强图像边缘，但调的太强会导致锯齿状。值域范围：0 ~ 1023。
u16DetailTh	SharpnessUD 阈值。值域范围：0 ~ 4095。
u8DetailByY	根据亮度调整 SharpnessUD。值域范围：0 ~ 255。
u16OverShootLimit	参考周围最亮点做调整，设 0 则 edge 上限为周围最亮点的 Y 值。值域范围：0 ~ 1023。
u16UndreShootLimit	参考周围最暗点做调整，设 0 则 edge 下限为周围最暗点的 Y 值，也就是不会 undershoot。值域范围：0 ~ 1023。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.25.5. 局部参数结构体 (Local Parameter Structure)**【说明】**

设定锐度的局部参数结构体。

【定义】

```
typedef struct SHARPNESS_LOCAL_PARAM_s
{
    MI_U8 u8OverShootGain;
    MI_U8 u8UnderShootGain;
    MI_U8 u8CorLUT[6];
    MI_U8 u8ScI_LUT[6];
} SHARPNESS_LOCAL_PARAM_t;
```

【名称】

变量名称	描述
u8OverShootGain	调整 edge overshoot 的强度程度，值越大锐化越强。 值域范围：0 ~ 255，默认值：128。

变量名称	描述
u8UnderShootGain	调整 edge undershoot 的强度程度，值越大锐化越强。 值域范围：0 ~ 255，默认值：128。
u8CorLUT[6]	抑制高亮度平坦区。值域范围：0 ~ 255，默认值：0。
u8ScI LUT[6]	根据亮度调整 edge 输出。值域范围：0 ~ 255，默认值：0。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.26 MI_ISP_IQ_CROSSTALK 结构体

3.1.26.1 类型结构体 (Type Structure)

【说明】

设定串扰的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_CROSSTALK_TYPE_s
{
    MI_ISP_BOOL_e          bEnable;
    MI_ISP_OP_TYPE_e       enOpType;
    CROSSTALK_AUTO_ATTR_t  stAuto;
    CROSSTALK_MANUAL_ATTR_t stManual;
} MI_ISP_IQ_CROSSTALK_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定串扰功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定串扰的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。 手动模式：SS_OP_TYP_MANUAL = 1。
stAuto	设定串扰的自动模式属性结构体。
stManual	设定串扰的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IO_SetCrossTalk](#)(MI_U32 Channel, [MI_ISP_IO_CROSSTALK_TYPE_t](#) *data);
MI_S32 [MI_ISP_IO_GetCrossTalk](#)(MI_U32 Channel, [MI_ISP_IO_CROSSTALK_TYPE_t](#) *data);

3.1.26.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定串扰的自动模式属性结构体。

【定义】

```
typedef struct CROSSTALK_AUTO_ATTR_s  
{  
    CROSSTALK\_PARAM\_t stParaAPI[MI_ISP_AUTO_NUM];  
} CROSSTALK_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.26.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定串扰的手动模式属性结构体。

【定义】

```
typedef struct CROSSTALK_MANUAL_ATTR_s  
{  
    CROSSTALK\_PARAM\_t stParaAPI;  
} CROSSTALK_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.26.4. 参数结构体 (Parameter Structure)

【说明】

设定串扰的参数结构体。

【定义】

```
typedef struct CROSSTALK_PARAM_s
{
    MI_U8      u8StrengthV2;
    MI_U16     u16ThresholdV2;
    MI_U16     u16ThresholdOffsetV2;
} CROSSTALK_PARAM_t;
```

【名称】

变量名称	描述
u8StrengthV2	串扰的可变强度数值。值域范围：0 ~ 31。
u16ThresholdV2	串扰的阈值比例数值。值域范围：0 ~ 255。
u16ThresholdOffsetV2	串扰的阈值平移数值。值域范围：0 ~ 4095。

【注意事项】

- 此功能開強會造成細節損失，因此如果沒有串擾問題則不需要開啟。

【相关结构类型与函式】

无。

3.1.27 MI_ISP_IQ_OBC 结构体

3.1.27.1. 类型结构体 (Type Structure)

【说明】

设定黑电平的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_OBC_TYPE_s
{
    MI_ISP_BOOL_e      bEnable;
    MI_ISP_OP_TYPE_e   enOpType;
    OBC_AUTO_ATTR_t    stAuto;
    OBC_MANUAL_ATTR_t  stManual;
} MI_ISP_IQ_OBC_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定黑电平功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定黑电平的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。 手动模式：SS_OP_TYP_MANUAL = 1。
stAuto	设定黑电平的自动模式属性结构体。
stManual	设定黑电平的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IO_SetOBC](#)(MI_U32 Channel, [MI_ISP_IO_OBC_TYPE_t](#) *data);
 MI_S32 [MI_ISP_IO_GetOBC](#)(MI_U32 Channel, [MI_ISP_IO_OBC_TYPE_t](#) *data);
 MI_S32 [MI_ISP_IO_SetOBC_P1](#)(MI_U32 Channel, [MI_ISP_IO_OBC_TYPE_t](#) *data);
 MI_S32 [MI_ISP_IO_GetOBC_P1](#)(MI_U32 Channel, [MI_ISP_IO_OBC_TYPE_t](#) *data);

3.1.27.2 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定黑电平的自动模式属性结构体。

【定义】

```
typedef struct OBC_AUTO_ATTR_s
{
    OBC\_PARAM\_t stParaAPI[MI_ISP_AUTO_NUM];
} OBC_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.27.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定黑电平的手动模式属性结构体。

【定义】

```
typedef struct OBC_MANUAL_ATTR_s  
{  
    OBC\_PARAM\_t stParaAPI;  
} OBC_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.27.4. 参数结构体 (Parameter Structure)

【说明】

设定黑电平的参数结构体。

【定义】

```
typedef struct OBC_PARAM_s  
{  
    MI_U16 u16ValR;  
    MI_U16 u16ValGr;  
    MI_U16 u16ValGb;  
    MI_U16 u16ValB;  
} OBC_PARAM_t;
```

【名称】

变量名称	描述
u16ValR	R 通道黑电平的可控制数值。 pipeline 0 值域范围：0 ~ 65535。 pipeline 1 值域范围：0 ~ 16383。
u16ValGr	Gr 通道黑电平的可控制数值。 pipeline 0 值域范围：0 ~ 65535。 pipeline 1 值域范围：0 ~ 16383。

变量名称	描述
u16ValGb	Gb 通道黑电平的可控制数值。 pipeline 0 值域范围：0 ~ 65535。 pipeline 1 值域范围：0 ~ 16383。
u16ValB	B 通道黑电平的可控制数值。 pipeline 0 值域范围：0 ~ 65535。 pipeline 1 值域范围：0 ~ 16383。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.28 MI_ISP_IQ_WDR 结构体

3.1.28.1. 类型结构体 (Type Structure)

【说明】

设定宽动态的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_WDR_TYPE_s
{
    MI_ISP_BOOL_e      bEnable;
    MI_ISP_OP_TYPE_e   enOpType;
    WDR_AUTO_ATTR_t    stAuto;
    WDR_MANUAL_ATTR_t  stManual;
} MI_ISP_IQ_WDR_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定宽动态功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定宽动态的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。 手动模式：SS_OP_TYP_MANUAL = 1。
stAuto	设定宽动态的自动模式属性结构体。
stManual	设定宽动态的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IQ_SetWDR](#)(MI_U32 Channel, [MI_ISP_IQ_WDR_TYPE_t](#) *data);

MI_S32 [MI_ISP_IQ_GetWDR](#)(MI_U32 Channel, [MI_ISP_IQ_WDR_TYPE_t](#) *data);

3.1.28.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定宽动态的自动模式属性结构体。

【定义】

```
typedef struct WDR_AUTO_ATTR_s
{
    WDR\_PARAM\_t stParaAPI[MI_ISP_AUTO_NUM];
} WDR_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.28.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定宽动态的手动模式属性结构体。

【定义】

```
typedef struct WDR_MANUAL_ATTR_s
{
    WDR\_PARAM\_t stParaAPI;
} WDR_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.28.4. 参数结构体 (Parameter Structure)

【说明】

设定宽动态的参数结构体。

【定义】

```
typedef struct WDR_PARAM_s
{
    MI_U8          u8BoxNum;
    MI_ISP_BOOL_e  bAutoDetailEnhance;
    MI_U8          u8ManualDetailEnhance;
    MI_U8          u8GlobalDarkToneEnhance;
    MI_U8          u8WDRStrByY[33];
    MI_U8          u8Strength;
    MI_U8          u8DarkLimit;
    MI_U8          u8BrightLimit;
    MI_ISP_BOOL_e  bGammaSyncEn;
} WDR_PARAM_t;
```

【名称】

变量名称	描述
u8BoxNum	区域宽动态作用范围的大小。值域范围：0 ~ 4。
bAutoDetailEnhance	额外的细节加强。值域范围：0 ~ 1。
u8ManualDetailEnhance	bAutoDetailEnhance = 1 时，可控制亮暗处细节的强度，值域范围：0 ~ 255。
u8GlobalDarkToneEnhance	全局宽动态的强度。值域范围：0 ~ 15。
u8WDRStrByY[33]	根据亮度控制宽动态强度。值域范围：0 ~ 255。
u8Strength	区域宽动态的强度。值域范围：0 ~ 255。
u8DarkLimit	限制暗区的强度。值域范围：0 ~ 255。
u8BrightLimit	限制亮区的强度。值域范围：0 ~ 255。
bGammaSyncEn	WDR 和 Gamma 连动开关。值域范围：0 ~ 1。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.29 MI_ISP_IQ_DYNAMIC_DP 结构体

3.1.29.1. 类型结构体 (Type Structure)

【说明】

设定动态补偿坏点的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_DYNAMIC_DP_TYPE_s
{
    MI_ISP_BOOL_e          bEnable;
    MI_ISP_OP_TYPE_e       enOpType;
    DYNAMIC_DP_AUTO_ATTR_t stAuto;
    DYNAMIC_DP_MANUAL_ATTR_t stManual;
} MI_ISP_IQ_DYNAMIC_DP_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定动态补偿坏点功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定动态补偿坏点的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。 手动模式：SS_OP_TYP_MANUAL = 1。
stAuto	设定动态补偿坏点的自动模式属性结构体。
stManual	设定动态补偿坏点的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

```
MI_S32 MI_ISP_IQ_SetDynamicDP(MI_U32 Channel, MI_ISP_IQ_DYNAMIC_DP_TYPE_t *data);
MI_S32 MI_ISP_IQ_GetDynamicDP(MI_U32 Channel, MI_ISP_IQ_DYNAMIC_DP_TYPE_t *data);
```

3.1.29.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定动态补偿坏点的自动模式属性结构体。

【定义】

```
typedef struct DYNAMIC_DP_AUTO_ATTR_s
{
    DYNAMIC\_DP\_PARAM\_t stParaAPI[MI_ISP_AUTO_NUM];
} DYNAMIC_DP_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.29.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定动态补偿坏点的手动模式属性结构体。

【定义】

```
typedef struct DYNAMIC_DP_MANUAL_ATTR_s
{
    DYNAMIC\_DP\_PARAM\_t stParaAPI;
} DYNAMIC_DP_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.29.4. 参数结构体 (Parameter Structure)

【说明】

设定动态补偿坏点的参数结构体。

【定义】

```
typedef struct DYNAMIC_DP_PARAM_s
{
    MI_ISP_BOOL_e bHotPixEn;
    MI_U16         u16HotPixCompSlpoe;
    MI_ISP_BOOL_e bDarkPixEn;
    MI_U16         u16DarkPixCompSlpoe;
    MI_U16         u16DPCTH;
    MI_ISP_BOOL_e bBlendEn;
    MI_U16         u16DiffLut[6];
    MI_U16         u16YLut[6];
} DYNAMIC_DP_PARAM_t;
```

【名称】

变量名称	描述
bHotPixEn	设定动态补偿 hot pixel 功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
u16HotPixCompSlpoe	调整动态补偿 hot pixel 的阈值。值域范围：0 ~ 255。
bDarkPixEn	设定动态补偿 dark pixel 功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
u16DarkPixCompSlpoe	调整动态补偿 dark pixel 的阈值。值域范围：0 ~ 255。
u16DPCTH	设定动态补偿坏点阈值。值域范围：0 ~ 255。
bBlendEn	设定 blending 功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
u16DiffLut[6]	依据 DPC 补偿结果和原来的点的差异来做混和。值域范围：0 ~ 1024。
u16YLut[6];	依据亮度来做混和。值域范围：0 ~ 1024。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.30 MI_ISP_IQ_DYNAMIC_DP_CLUSTER 结构体

3.1.30.1. 类型结构体 (Type Structure)

【说明】

设定动态补偿坏点的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_DYNAMIC_DP_CLUSTER_TYPE_s
{
    MI_ISP_BOOL_e                bEnable;
    MI_ISP_OP_TYPE_e            enOpType;
    DYNAMIC_DP_CLUSTER_AUTO_ATTR_t stAuto;
    DYNAMIC_DP_CLUSTER_MANUAL_ATTR_t stManual;
} MI_ISP_IQ_DYNAMIC_DP_CLUSTER_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定动态补偿坏点功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定动态补偿坏点的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。 手动模式：SS_OP_TYP_MANUAL = 1。
stAuto	设定动态补偿坏点的自动模式属性结构体。
stManual	设定动态补偿坏点的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

```
MI_S32 MI_ISP_IQ_SetDynamicDP_CLUSTER(MI_U32 Channel, MI_ISP_IQ_DYNAMIC_DP_CLUSTER_TYPE_t
*data);
MI_S32 MI_ISP_IQ_GetDynamicDP_CLUSTER (MI_U32 Channel, MI_ISP_IQ_DYNAMIC_DP_CLUSTER_TYPE_t
*data);
```

3.1.30.2 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定动态补偿坏点的自动模式属性结构体。

【定义】

```
typedef struct DYNAMIC_DP_CLUSTER_AUTO_ATTR_s
{
    DYNAMIC_DP_PARAM_t stParaAPI[MI_ISP_AUTO_NUM];
} DYNAMIC_DP_CLUSTER_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.30.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定动态补偿坏点的手动模式属性结构体。

【定义】

```
typedef struct DYNAMIC_DP_CLUSTER_MANUAL_ATTR_s
{
    DYNAMIC\_DP\_PARAM\_t stParaAPI;
} DYNAMIC_DP_CLUSTER_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.30.4. 参数结构体 (Parameter Structure)

【说明】

设定动态补偿坏点的参数结构体。

【定义】

```
typedef struct DYNAMIC_DP_CLUSTER_PARAM_s
{
    MI_ISP_BOOL_e bEdgeMode;
    MI_U16         u16NeiDeltaTh;
    MI_U8          u8NeiSmooth;
    MI_ISP_BOOL_e bSortEn;
    MI_ISP_BOOL_e bSortRGGEn[4];
    MI_ISP_BOOL_e bSort1x3ModeEn;
    MI_U16         u16SortLumTbIL[6];
    MI_U16         u16SortLumTbID[6];
    MI_U16         u16SortLumTbIS[6];
} DYNAMIC_DP_CLUSTER_PARAM_t;
```

【名称】

变量名称	描述
bEdgeMode	设定 Edge Mode 功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
u16NeiDeltaTh	中心点与周围 8 点的差异阈值。值域范围：0 ~ 4095。
u8NeiSmooth	累计阈值。值域范围：0 ~ 8。
bSortEn	Sort mode 总开关
bSortRGGEn[4]	Sort mode 各自 channel 开关。
bSort1x3ModeEn	1x3 mode 开关。
u16SortLumTbIL[6]	最亮点和次亮点阈值。值域范围：0 ~ 4095。
u16SortLumTbID[6]	最暗点和次暗点阈值。值域范围：0 ~ 4095。
u16SortLumTbIS[6]	次亮(暗)点和第三亮(暗)点阈值。值域范围：0 ~ 4095。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.31 MI_ISP_IQ_HSV 结构体

3.1.31.1. 类型结构体 (Type Structure)

【说明】

设定 HSV 的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_HSV_TYPE_s
{
    MI_ISP_BOOL_e      bEnable;
    MI_ISP_OP_TYPE_e   enOpType;
    HSV_AUTO_ATTR_t    stAuto;
    HSV_MANUAL_ATTR_t  stManual;
} MI_ISP_IQ_HSV_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定 HSV 功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定 HSV 的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。 手动模式：SS_OP_TYP_MANUAL = 1。
stAuto	设定 HSV 的自动模式属性结构体。
stManual	设定 HSV 的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 MI_ISP_IQ_SetHSV(MI_U32 Channel, MI_ISP_IQ_HSV_TYPE_t *data);
MI_S32 MI_ISP_IQ_GetHSV(MI_U32 Channel, MI_ISP_IQ_HSV_TYPE_t *data);

3.1.31.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定 HSV 的自动模式属性结构体。

【定义】

```
typedef struct HSV_AUTO_ATTR_s
{
    HSV_PARAM_t stParaAPI[MI_ISP_AUTO_NUM];
} HSV_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.31.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定 HSV 的手动模式属性结构体。

【定义】

```
typedef struct HSV_MANUAL_ATTR_s
{
    HSV_PARAM t stParaAPI;
} HSV_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.31.4. 参数结构体 (Parameter Structure)

【说明】

设定 HSV 的参数结构体。

【定义】

```
typedef struct HSV_PARAM_s
{
    MI_S16    s16HueLut[24];
    MI_U16    u16SatLut[24];
    MI_S16    s16HueLut_ForUnitCCM[24];
    MI_U16    u16SatLut_ForUnitCCM[24];
    MI_U8     u8GlobalSat;
} HSV_PARAM_t;
```

【名称】

变量名称	描述
s16HueLut[24]	将 360 度色相等分为 24 份，分别控制每个等分的色相旋转角度，自动模式下根据色温作切换。 值域范围：-15 ~ 15
u16SatLut[24]	将 360 度色相等分为 24 份，分别控制每个等分的饱和度，自动模式下根据色温作切换。 值域范围：0 ~ 255 (64 = 1x)
s16HueLut_ForUnitCCM[24]	将 360 度色相等分为 24 份，分别控制每个等分的色相旋转角度与 Unit CCM 搭配参数，自动模式下根据色温作切换。 值域范围：-15~15
u16SatLut_ForUnitCCM[24]	将 360 度色相等分为 24 份，分别控制每个等分的饱和度与 Unit CCM 搭配参数，自动模式下根据色温作切换。 值域范围：0~255 (64 = 1x)
u8GlobalSat	整体饱和度控制，自动模式下根据增益作切换。 值域范围：0 ~ 255 (64 = 1x)

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.32 MI_ISP_IQ_RGBIR 结构体

3.1.32.1 类型结构体 (Type Structure)

【说明】

设定 RGBIR 的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_RGBIR_TYPE_s
{
    MI_ISP_BOOL_e          bEnable;
    MI_ISP_OP_TYPE_e       enOpType;
    RGBIR_AUTO_ATTR_t      stAuto;
    RGBIR_MANUAL_ATTR_t    stManual;
} MI_ISP_IQ_RGBIR_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定 RGBIR 功能的布尔值。 关闭: SS_FALSE = 0。 开启: SS_TRUE = 1。
enOpType	设定 RGBIR 的工作模式。 自动模式: SS_OP_TYP_AUTO = 0。 手动模式: SS_OP_TYP_MANUAL = 1。
stAuto	设定 RGBIR 的自动模式属性结构体。
stManual	设定 RGBIR 的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IO_SetRGBIR](#)(MI_U32 Channel, [MI_ISP_IO_RGBIR_TYPE_t](#) *data);

MI_S32 [MI_ISP_IO_GetRGBIR](#)(MI_U32 Channel, [MI_ISP_IO_RGBIR_TYPE_t](#) *data);

3.1.32.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定 RGBIR 的自动模式属性结构体。

【定义】

```
typedef struct RGBIR_AUTO_ATTR_s
{
    RGBIR\_PARAM\_t stParaAPI[MI_ISP_AUTO_NUM];
} RGBIR_AUTO_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体, MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.32.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定 RGBIR 的手动模式属性结构体。

【定义】

```
typedef struct RGBIR_MANUAL_ATTR_s
{
    RGBIR\_PARAM\_t stParaAPI;
} RGBIR_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.32.4. 参数结构体 (Parameter Structure)

【说明】

设定 RGBIR 的参数结构体。

【定义】

```
typedef struct RGBIR_PARAM_s
{
    MI_U8          u8IrPosType;
    MI\_ISP\_BOOL\_e bRemoveEn;
    MI_U16         u16Ratio_R[6];
    MI_U16         u16Ratio_G[6];
    MI_U16         u16Ratio_B[6];
} RGBIR_PARAM_t;
```

【名称】

变量名称	描述
u8IrPosType	RGBIR 分布型态。值域范围：0 ~ 7，默认值：0。
bRemoveEn	设定减 IR 功能的布尔值，默认值：1。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
u16Ratio_R[6]	红色减 IR 比例，由暗至亮区@12bits

变量名称	描述
u16Ratio_G[6]	绿色减 IR 比例，由暗至亮区@12bits
u16Ratio_B[6]	蓝色减 IR 比例，由暗至亮区@12bits

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.33 MI_ISP_IQ_DEMOSAIC 结构体

3.1.33.1. 类型结构体 (Type Structure)

【说明】

设定去马赛克的结构体。

【定义】

```
typedef struct MI_ISP_IQ_DEMOSAIC_TYPE_s
{
    MI_ISP_BOOL_e          bEnable;
    DEMOSAIC_MANUAL_ATTR_t stManual;
} MI_ISP_IQ_DEMOSAIC_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定去马赛克功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
stManual	设定去马赛克的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 MI_ISP_IQ_SetDEMOSAIC(MI_U32 Channel, MI_ISP_IQ_DEMOSAIC_TYPE_t *data);
MI_S32 MI_ISP_IQ_GetDEMOSAIC(MI_U32 Channel, MI_ISP_IQ_DEMOSAIC_TYPE_t *data);

3.1.33.2. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定去马赛克的手动模式属性结构体。

【定义】

```
typedef struct DEMOSAIC_MANUAL_ATTR_s  
{  
    DEMOSAIC\_PARAM\_t paraAPI;  
} DEMOSAIC_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.33.3. 参数结构体 (Parameter Structure)

【说明】

设定去马赛克的参数结构体。

【定义】

```
typedef struct DEMOSAIC_PARAM_s  
{  
    MI_U8 u8DirThrd;  
    MI_U8 u8EdgeSmoothThrdY;  
    MI_U8 u8EdgeSmoothThrdC;  
} DEMOSAIC_PARAM_t;
```

【名称】

变量名称	描述
u8DirThrd	方向性的阈值，超过此阈值，去马赛克会取方向性的结果来替代。值域范围：0 ~ 63，默认值：35。
u8EdgeSmoothThrdY	Smooth edge 的亮度阈值，值越小边缘越不清晰，但假边可减少。值域范围：0 ~ 255，默认值：230。
u8EdgeSmoothThrdC	Smooth edge 的颜色阈值，值越小边缘越不清晰，但假边可减少。值域范围：0 ~ 127，默认值：20。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.34 MI_ISP_IQ_R2Y 结构体

3.1.34.1 类型结构体 (Type Structure)

【说明】

设定色彩转换的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_R2Y_TYPE_s
{
    MI_ISP_BOOL_e      bEnable;
    R2Y_MANUAL_ATTR_t  stManual;
} MI_ISP_IQ_R2Y_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定色彩转换功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
stManual	设定色彩转换的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

```
MI_S32 MI_ISP_IQ_SetR2Y(MI_U32 Channel, MI_ISP_IQ_R2Y_TYPE_t *data);
MI_S32 MI_ISP_IQ_GetR2Y(MI_U32 Channel, MI_ISP_IQ_R2Y_TYPE_t *data);
```

3.1.34.2 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定色彩转换的手动模式属性结构体。

【定义】

```
typedef struct R2Y_MANUAL_ATTR_s
{
    R2Y_PARAM_t stParaAPI;
} R2Y_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.34.3. 参数结构体 (Parameter Structure)

【说明】

设定色彩转换的参数结构体。

【定义】

```
typedef struct R2Y_PARAM_s
{
    MI_U16 u16Matrix[9];
    MI_U8 u8AddY16;
} R2Y_PARAM_t;
```

【名称】

变量名称	描述
u16Matrix[9]	调整 RGB to YUV 矩阵，值域范围：0 ~ 8191，1 倍为 1024，负数以 2 的补码表示，等同于±3.996。
u8AddY16	调整 Y Offset，经过矩阵转换后，用来控制是否要对 Y 加 16。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.35 MI_ISP_IQ_COLORTTRANS 结构体

3.1.35.1. 类型结构体 (Type Structure)

【说明】

设定色彩转换的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_COLORTRANS_TYPE_s
{
    MI_ISP_BOOL_e          bEnable;
    COLORTRANS_MANUAL_ATTR_t stManual;
} MI_ISP_IQ_COLORTRANS_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定色彩转换功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
stManual	设定色彩转换的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 MI_ISP_IQ_SetCOLORTRANS(MI_U32 Channel, MI_ISP_IQ_COLORTRANS_TYPE_t *data);
MI_S32 MI_ISP_IQ_GetCOLORTRANS(MI_U32 Channel, MI_ISP_IQ_COLORTRANS_TYPE_t *data);

3.1.35.2. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定色彩转换的手动模式属性结构体。

【定义】

```
typedef struct COLORTRANS_MANUAL_ATTR_s
{
    COLORTRANS_PARAM_t stParaAPI;
} COLORTRANS_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.35.3. 参数结构体 (Parameter Structure)

【说明】

设定色彩转换的参数结构体。

【定义】

```
typedef struct COLORTRANS_PARAM_s
{
    MI_U16 u16Y_OFST;
    MI_U16 u16U_OFST;
    MI_U16 u16V_OFST;
    MI_U16 u16Matrix[9];
} COLORTRANS_PARAM_t;
```

【名称】

变量名称	描述
u16Y_OFST	调整 Y 偏差值。10bit domain，值域范围：0 ~ 2047，负数以 2 的补码表示，等同于 ±1023。
u16U_OFST	调整 U 偏差值。10bit domain，值域范围：0 ~ 2047，负数以 2 的补码表示，等同于 ±1023。
u16V_OFST	调整 V 偏差值。10bit domain，值域范围：0 ~ 2047，负数以 2 的补码表示，等同于 ±1023。
u16Matrix[9]	调整 RGBYUV 矩阵，值域范围：0 ~ 1023，1 倍为 256，负数以 2 的补码表示，等同于 ±1.996。

【注意事项】

对于 YUV_OFST，Matrix[9]，负数以 2 的补码表示。
UV 偏差值，在矩阵相乘后，已经预设加上 128。

下面例子为，将 YUV 转为 16~235 的值域范围

$$Y = (0.257 * R) + (0.504 * G) + (0.098 * B) + 16$$

$$Cb = -(0.148 * R) - (0.291 * G) + (0.439 * B) + 128$$

$$Cr = (0.439 * R) - (0.368 * G) - (0.071 * B) + 128$$

Matrix[9] = {66, 129, 25, 986, 950, 112, 112, 930, 1006}

Y_OFST = 64

U_OFST = 0 (预设以加上 128*4，所以填 0)

V_OFST = 0 (预设以加上 128*4，所以填 0)

【相关结构类型与函式】

无。

3.1.36 MI_ISP_IQ_HDR 结构体

3.1.36.1. 类型结构体 (Type Structure)

【说明】

设定高动态范围的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_HDR_TYPE_s
{
    MI_ISP_BOOL_e          bEnable;
    MI_ISP_OP_TYPE_e       enOpType;
    HDR_AUTO_ATTR_t        stAuto;
    HDR_MANUAL_ATTR_t      stManual;
} MI_ISP_IQ_HDR_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定高动态范围功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
enOpType	设定高动态范围的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。 手动模式：SS_OP_TYP_MANUAL = 1。
stAuto	设定高动态范围的自动模式属性结构体。
stManual	设定高动态范围的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

```
MI_S32 MI_ISP_IQ_SetHDR(MI_U32 Channel, MI_ISP_IQ_HDR_TYPE_t *data);
MI_S32 MI_ISP_IQ_GetHDR(MI_U32 Channel, MI_ISP_IQ_HDR_TYPE_t *data);
```

3.1.36.2. 自动模式属性结构体 (Automatic Mode Attribute Structure)

【说明】

设定高动态范围的自动模式属性结构体。

【定义】

```
typedef struct HDR_AUTO_ATTR_s
{
    HDR_PARAM_t stParaAPI[MI_ISP_AUTO_NUM];
} HDR_AUTO_ATTR_t;
```


【名称】

变量名称	描述
stParaAPI[MI_ISP_AUTO_NUM]	自动模式属性结构体，MI_ISP_AUTO_NUM = 16。 该数列组的 16 个值分别对应在不同增益情况下的设定值。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.36.3. 手动模式属性结构体 (Manual Mode Attribute Structure)

【说明】

设定高动态范围的手动模式属性结构体。

【定义】

```
typedef struct HDR_MANUAL_ATTR_s
{
    HDR\_PARAM\_t stParaAPI;
} HDR_MANUAL_ATTR_t;
```

【名称】

变量名称	描述
stParaAPI	手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.36.4. 参数结构体 (Parameter Structure)

【说明】

设定高动态范围的参数结构体。

【定义】

```
typedef struct HDR_PARAM_s
{
    MI_U16    u16SensorExpRatio;
    MI_U16    bNrEn;
    MI_U8     bDynRatioEn;
    MI_U8     u8CRefSel;
    MI_U16    u16YwtTh1;
    MI_U16    u16YwtTh2;
} HDR_PARAM_t;
```

【名称】

变量名称	描述
u16SensorExpRatio	设定 HDR 长短曝光比例。值域范围：2048 ~ 65535。
bNrEn	控制短曝 NR 开关。值域范围：0 ~ 1。
bDynRatioEn	控制动态补偿线性度开关。值域范围：0 ~ 1。
u8CRefSel	选择长曝或短曝，来控制动态补偿， 值域范围：0 ~ 1 (0：长曝、1：短曝)。
u16YwtTh1	长曝曝光的最低亮度门阀。值域范围：0 ~ 1022。
u16YwtTh2	长曝曝光的最高亮度门阀。值域范围：1 ~ 1023。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.37 MI_ISP_IQ_LSC 结构体

3.1.37.1 类型结构体 (Type Structure)

【说明】

设定 LSC 的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_LSC_TYPE_s
{
    MI_ISP_BOOL_e bEnable;
    MI_U8          u8RGainTable[32*3];
    MI_U8          u8GGainTable[32*3];
    MI_U8          u8BGainTable[32*3];
} MI_ISP_IQ_LSC_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定 LSC 功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
u8RGainTable[32*3]	红色通道的 LSC 补正 table。值域范围：0 ~ 255。
u8GGainTable[32*3]	绿色通道的 LSC 补正 table。值域范围：0 ~ 255。
u8BGainTable[32*3]	蓝色通道的 LSC 补正 table。值域范围：0 ~ 255。

【相关结构类型与函式】

MI_S32 [MI_ISP_IQ_SetLSC](#)(MI_U32 Channel, [MI_ISP_IQ_LSC_TYPE_t](#) *data);

MI_S32 [MI_ISP_IQ_GetLSC](#)(MI_U32 Channel, [MI_ISP_IQ_LSC_TYPE_t](#) *data);

3.1.38 MI_ISP_IQ_LSC_CTRL 结构体

3.1.38.1 类型结构体 (Type Structure)

【说明】

设定 LSC CTRL 的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_LSC_CTRL_TYPE_s
{
    MI\_ISP\_BOOL\_e bEnable;
    MI_U8      u8RRatioByCct[3];
    MI_U8      u8GRatioByCct[3];
    MI_U8      u8BRatioByCct[3];
    INTP_LUT_IQ RatioByBv;
} MI_ISP_IQ_LSC_TYPE_t;
typedef struct
{
    MI_U16      u16NumOfPoints;
    MI_S32      s32Y[16];
    MI_S32      s32X[16];
} INTP_LUT_IQ;
```

【名称】

变量名称	描述
bEnable	设定 LSC CTRL 功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
u8RRatioByCct [3]	红色通道的 LSC 额外 ratio。值域范围：0 ~ 128 (128 = 1x)。

变量名称	描述
u8GRatioByCct [3]	绿色通道 LSC 额外 ratio。值域范围：0 ~ 128 (128 = 1x)。
u8BRatioByCct [3]	蓝色通道 LSC 额外 ratio。值域范围：0 ~ 128 (128 = 1x)。
RatioByBv.u16NumOfPoints	RatioByBv 节点个数。值域范围：0 ~ 16。
RatioByBv.s32Y	各节点 Ratio。值域范围：0 ~ 128 (128 = 1x)。
RatioByBv.s32X	各节点 BV。

【相关结构类型与函式】

MI_S32 [MI_ISP_IQ_SetLSC_CTRL](#)(MI_U32 Channel, [MI_ISP_IQ_LSC_CTRL_TYPE_t](#) *data);

MI_S32 [MI_ISP_IQ_GetLSC_CTRL](#) (MI_U32 Channel, [MI_ISP_IQ_LSC_CTRL_TYPE_t](#) *data);

3.1.39 MI_ISP_IQ_ALSC 结构体

3.1.39.1 类型结构体 (Type Structure)

【说明】

设定 ALSC 的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_ALSC_TYPE_s
{
    MI\_ISP\_BOOL\_e bEnable;
    MI_U16          u16TableR[255*3];
    MI_U16          u16TableG[255*3];
    MI_U16          u16TableB[255*3];
} MI_ISP_IQ_ALSC_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定 ALSC 功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
u16TableR[255*3]	红色通道的 ALSC 补正 table。值域范围：0 ~ 4095 (1024 = 1x)。
u16TableG[255*3]	绿色通道 LSC 补正 table。值域范围：0 ~ 4095 (1024 = 1x)。
u16TableB[255*3]	蓝色通道的 ALSC 补正 table。值域范围：0 ~ 4095 (1024 = 1x)。

【注意事项】

- R、G、B Table 数值不建议设低於 1024，否则高亮部分会因为无法饱和而造成色偏问题。

【相关结构类型与函式】

MI_S32 [MI_ISP_IQ_SetALSC](#)(MI_U32 Channel, [MI_ISP_IQ_ALSC_TYPE_t](#) *data);

MI_S32 [MI_ISP_IQ_GetALSC](#)(MI_U32 Channel, [MI_ISP_IQ_ALSC_TYPE_t](#) *data);

3.1.40 MI_ISP_IQ_ALSC_CTRL 结构体

3.1.40.1 类型结构体 (Type Structure)

【说明】

设定 ALSC CTRL 的类型结构体。

【定义】

```
typedef struct MI_ISP_IQ_ALSC_CTRL_TYPE_s
{
    MI\_ISP\_BOOL\_e bEnable;
    MI_U8      u8RRatioByCct[3];
    MI_U8      u8GRatioByCct[3];
    MI_U8      u8BRatioByCct[3];
    INTP_LUT_IQ RatioByBv;
} MI_ISP_IQ_ALSC_CTRL_TYPE_t;
typedef struct
{
    MI_U16      u16NumOfPoints;
    MI_S32      s32Y[16];
    MI_S32      s32X[16];
} INTP_LUT_IQ;
```

【名称】

变量名称	描述
bEnable	设定 LSC CTRL 功能的布尔值。 关闭: SS_FALSE = 0。 开启: SS_TRUE = 1。
u8RRatioByCct [3]	红色通道的 ALSC 额外 ratio。值域范围: 0 ~ 128 (128 = 1x)。
u8GRatioByCct [3]	绿色通道的 ALSC 额外 ratio。值域范围: 0 ~ 128 (128 = 1x)。
u8BRatioByCct [3]	蓝色通道的 ALSC 额外 ratio。值域范围: 0 ~ 128 (128 = 1x)。
RatioByBv.u16NumOfPoints	RatioByBv 节点个数。值域范围: 0 ~ 16。
RatioByBv.s32Y	各节点 Ratio。值域范围: 0 ~ 128 (128 = 1x)。
RatioByBv.s32X	各节点 BV。

【相关结构类型与函式】

MI_S32 [MI_ISP_IQ_SetALSC_CTRL](#)(MI_U32 Channel, [MI_ISP_IQ_ALSC_CTRL_TYPE_t](#) *data);

MI_S32 [MI_ISP_IQ_GetALSC_CTRL](#) (MI_U32 Channel, [MI_ISP_IQ_ALSC_CTRL_TYPE_t](#) *data);

3.1.41 MI_ISP_API_BYPASS 结构体

3.1.41.1 类型结构体 (Type Structure)

【说明】

设定 ISP 对应 IP 模块的略过模式的类型结构体。

【定义】

```
typedef struct MI_ISP_API_BYPASS_TYPE_s
{
    MI\_ISP\_BYPASS\_MODE\_e bEnable;
    MI\_ISP\_API\_ID\_e eAPIIndex;
} MI_ISP_API_BYPASS_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定 ISP 对应 IP 模块的略过模式功能的布尔值。 关闭: SS_FALSE = 0。 开启: SS_TRUE = 1。
eAPIIndex	设定 ISP API ID 列举数值。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_IO_SetAPIBypassMode](#)(MI_U32 Channel, [MI_ISP_API_BYPASS_TYPE_t](#) *data);

MI_S32 [MI_ISP_IO_GetAPIBypassMode](#)(MI_U32 Channel, [MI_ISP_API_BYPASS_TYPE_t](#) *data);

3.1.42 MI_ISP_AE_EXPO_VALUE 结构体

3.1.42.1 类型结构体 (Type Structure)

【说明】

纪录曝光参数的类型结构体。

【定义】

```
typedef struct MI_ISP_AE_EXPO_VALUE_TYPE_s
{
    MI_U32 u32FNx10;
    MI_U32 u32SensorGain;
    MI_U32 u32ISPGain;
    MI_U32 u32US;
} MI_ISP_AE_EXPO_VALUE_TYPE_t;
```

【名称】

变量名称	描述
u32FNx10	光圈值×10 (ex: 光圈 F1.8 则此项等于 18)。
u32SensorGain	Sensor 增益值 (含 sensor 模拟增益与 sensor 数字增益, 1024 等于 1 倍)。
u32ISPGain	ISP 数字增益(1024 等于 1 倍)。
u32US	曝光时间(μsec)。

【注意事项】

无。

【相关结构类型与函式】

[MI_ISP_AE_EXPO_INFO_TYPE_t](#)

MI_S32 [MI_ISP_AE_SetManualExpo](#)(MI_U32 Channel, [MI_ISP_AE_EXPO_VALUE_TYPE_t](#) *data);

MI_S32 [MI_ISP_AE_GetManualExpo](#)(MI_U32 Channel, [MI_ISP_AE_EXPO_VALUE_TYPE_t](#) *data);

3.1.43 MI_ISP_AE_HIST_WEIGHT_Y 结构体

3.1.43.1. 类型结构体 (Type Structure)

【说明】

纪录当前画面亮度及直方统计数据之结构体。

【定义】

```
typedef struct MI_ISP_AE_HIST_WEIGHT_Y_TYPE_s
{
    MI_U32 u32LumY;
    MI_U32 u32AvgY;
    MI_U32 u32Hits[128];
} MI_ISP_AE_HIST_WEIGHT_Y_TYPE_t;
```

【名称】

变量名称	描述
u32LumY	经区块权重表加权过后的当前画面亮度。
u32AvgY	当前画面的平均亮度(无加权)。
u32Hits[128]	画面值方图统计资料(共 128 柱对应到 0 ~ 255)。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_AE_GetHistoWghtY](#)(MI_U32 Channel, [MI_ISP_AE_HIST_WEIGHT_Y_TYPE_t](#) *data);

3.1.44 MI_ISP_AE_EXPO_INFO 结构体

3.1.44.1 类型结构体 (Type Structure)

【说明】

获取当前整体曝光信息的结构体。

【定义】

```
typedef struct MI_ISP_AE_EXPO_INFO_TYPE_s
{
    MI\_ISP\_BOOL\_e                blsStable;
    MI\_ISP\_BOOL\_e                blsReachBoundary;
    MI\_ISP\_AE\_EXPO\_VALUE\_TYPE\_t  stExpoValueLong;
    MI\_ISP\_AE\_EXPO\_VALUE\_TYPE\_t  stExpoValueShort;
    MI\_ISP\_AE\_HIST\_WEIGHT\_Y\_TYPE\_t stHistWeightY;
    MI_U32                      u32LVx10;
    MI_S32                      s32BV;
    MI_U32                      u32SceneTarget;
} MI_ISP_AE_EXPO_INFO_TYPE_t;
```

【名称】

变量名称	描述
blsStable	AE 是否收敛(1: 收敛, 0: 尚未收敛)。
blsReachBoundary	当前曝光参数是否已顶到变动范围上下界(1: 顶到, 0: 未顶到)。
stExpoValueLong	当前曝光参数(如为 HDR mode 则为长曝曝光参数)。
stExpoValueShort	HDR mode 下的短曝曝光参数。
stHistWeightY	当前画面亮度及直方图统计资料。
u32LVx10	当前 LV 数值× 10。
s32BV	当前 BV 数值× 16384。
u32SceneTarget	当前 AE 目标亮度。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_AE_QueryExposureInfo](#)(MI_U32 Channel, [MI_ISP_AE_EXPO_INFO_TYPE_t](#) *data);

3.1.45 MI_ISP_AE_EV_COMP 结构体

3.1.45.1. 类型结构体 (Type Structure)

【说明】

设定额外增减曝光补偿强度的结构体。

【定义】

```
typedef struct MI_ISP_AE_EV_COMP_TYPE_s
{
    MI_S32    s32EV;
    MI_U32    u32Grad;
} MI_ISP_AE_EV_COMP_TYPE_t;
```

【名称】

变量名称	描述
s32EV	额外的增减曝光能亮之分子。值域范围：-10 ~ 10。
u32Grad	额外的增减曝光能亮之分母。值域范围：-10 ~ 10。

【注意事项】

- Ex：s32EV = 1、u32Grad = 2，则曝光能亮會較原訂自動模式再額外增加 1/2 EV。

【相关结构类型与函式】

MI_S32 [MI_ISP_AE_SetEVComp](#)(MI_U32 Channel, [MI_ISP_AE_EV_COMP_TYPE_t](#) *data);
 MI_S32 [MI_ISP_AE_GetEVComp](#)(MI_U32 Channel, [MI_ISP_AE_EV_COMP_TYPE_t](#) *data);

3.1.46 MI_ISP_AE_MODE 结构体

3.1.46.1. 类型结构体 (Type Structure)

【说明】

设定全自动、半自动或手动曝光模式。

【定义】

```
typedef enum
{
    SS_AE_MODE_A,
    SS_AE_MODE_AV,
    SS_AE_MODE_SV,
    SS_AE_MODE_TV,
    SS_AE_MODE_M,
    SS_AE_MODE_MAX
} MI_ISP_AE_MODE_TYPE_e;
```

【名称】

变量名称	描述
SS_AE_MODE_A	全自动曝光模式。
SS_AE_MODE_AV	光圈先决模式。
SS_AE_MODE_SV	增益先决模式。
SS_AE_MODE_TV	快门先决模式。
SS_AE_MODE_M	手动模式。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 MI_ISP_AE_SetExpoMode(MI_U32 Channel, MI_ISP_AE_MODE_TYPE_e *data);

MI_S32 MI_ISP_AE_GetExpoMode(MI_U32 Channel, MI_ISP_AE_MODE_TYPE_e *data);

3.1.47 MI_ISP_AE_INTP_LUT 结构体

3.1.47.1. 类型结构体 (Type Structure)

【说明】

具结点内插功能之参数数据设定表。

可用于设定随环境亮度变化的画面目标亮度(ex: X轴: BV、Y轴: 画面目标亮度)。

或随环境亮度变化的画面目标亮度浮动范围(ex: X轴: BV、Y轴: 画面目标亮度的浮动范围)。

【定义】

```
typedef struct MI_ISP_AE_INTP_LUT_TYPE_s
{
    MI_U16 u16NumOfPoints;
    MI_U32 u32Y[16];
    MI_U32 u32X[16];
} MI_ISP_AE_INTP_LUT_TYPE_t;
```

【名称】

变量名称	描述
u16NumOfPoints	实际表中数据的有效节点数，最大支持 16 个节点，也就是可填入 16 笔 X、Y 数据。
u32Y[16]	对应每一节点的 Y 轴参数，如不使用的节点可将参数设 0。
u32X[16];	对应每一节点的 X 轴参数，如不使用的节点可将参数设 0。

【注意事项】

无。

【相关结构类型与函式】

[MI_ISP_AE_STRATEGY_TYPE_t](#)MI_S32 [MI_ISP_AE_SetTarget](#)(MI_U32 Channel, [MI_ISP_AE_INTP_LUT_TYPE_t](#) *data);MI_S32 [MI_ISP_AE_GetTarget](#)(MI_U32 Channel, [MI_ISP_AE_INTP_LUT_TYPE_t](#) *data);

3.1.48 MI_ISP_AE_CONV_CONDITON 结构体

3.1.48.1. 类型结构体 (Type Structure)

【说明】

设定自动曝光的收敛条件。

【定义】

```
typedef struct MI_ISP_AE_CONV_CONDITON_TYPE_s
{
    CONV\_THD\_PARAM\_t      stConvThrd;
    CONV\_SPEED\_PARAM\_t    stConvSpeed;
} MI_ISP_AE_CONV_CONDITON_TYPE_t;
```

【名称】

变量名称	描述
stConvThrd	设定自动曝光启动收敛或判断收敛完成的区间。
stConvSpeed	设定自动曝光之收敛速度。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_AE_SetConverge](#)(MI_U32 Channel, [MI_ISP_AE_CONV_CONDITON_TYPE_t](#) *data);MI_S32 [MI_ISP_AE_GetConverge](#)(MI_U32 Channel, [MI_ISP_AE_CONV_CONDITON_TYPE_t](#) *data);

3.1.48.2. 收敛区间结构体 (CONV_THD_PARAM_t)

【说明】

设定自动曝光启动收敛或判断收敛完成的区间。

【定义】

```
typedef struct CONV_THD_PARAM_s
{
    MI_U32 u32InThd;
    MI_U32 u32OutThd;
} CONV_THD_PARAM_t;
```

【名称】

变量名称	描述
u32InThd	若画面加权亮度与画面目标亮度的差小于此数值，则判断为自动曝光已收敛。值域范围：1 ~ 100。
u32OutThd	若画面加权亮度与画面目标亮度的差大于此数值，则判断为自动曝光未收敛，自动曝光将启动收敛程序。值域范围：2 ~ 100。

【注意事项】

无。

【相关结构类型与函式】

[MI_ISP_AE_CONV_CONDITON_TYPE_t](#)。

3.1.48.3. 收敛速度结构体 (CONV_SPEED_PARAM_t)

【说明】

设定自动曝光的收敛速度。

共提供四组节点，X 轴为对应的画面亮度，Y 轴为对应画面亮度的收敛速度。

【定义】

```
typedef struct CONV_SPEED_PARAM_s
{
    MI_U32 u32SpeedX[4];
    MI_U32 u32SpeedY[4];
} CONV_SPEED_PARAM_t;
```

【名称】

变量名称	描述
u32SpeedX[4]	选择要对到的当前画面加权亮度。值域范围：1 ~ 2550。
u32SpeedY[4]	收敛速度，基础为 1024 (ex: 若 Y 设为 500，则当画面亮度刚好落到对应 Y 的 X 节点时，自动曝光往曝光目标逼近一次的幅度为 500 / 1024)。 值域范围：1 ~ 1024。

【注意事项】

- u32SpeedX[1] & u32SpeedX[2] 中間兩節點固定內定為當前的畫面目標亮度，以便在目標亮度不固定的狀況下能自動對應到相應的數值，故手動修改這兩項將無效果。

【相关结构类型与函式】

无。

3.1.49 MI_ISP_AE_EXPO_LIMIT 结构体

3.1.49.1. 类型结构体 (Type Structure)

【说明】

设定自动曝光运作的极值。

【定义】

```
typedef struct MI_ISP_AE_EXPO_LIMIT_TYPE_s  
{  
    MI_U32 u32MinShutterUS;  
    MI_U32 u32MaxShutterUS;  
    MI_U32 u32MinFNx10;  
    MI_U32 u32MaxFNx10;  
    MI_U32 u32MinSensorGain;  
    MI_U32 u32MinISPGain;  
    MI_U32 u32MaxSensorGain;  
    MI_U32 u32MaxISPGain;  
} MI_ISP_AE_EXPO_LIMIT_TYPE_t;
```

【名称】

变量名称	描述
u32MinShutterUS	最短 Shutter 限制(μ sec)。值域范围：1 ~ 1000000。
u32MaxShutterUS	最长 Shutter 限制(μ sec)。值域范围：1 ~ 1000000。
u32MinFNx10	最小光圈值(ex: F1.8 等于 18)。值域范围：10 ~ 220。
u32MaxFNx10	最大光圈值(ex: F1.8 等于 18)。值域范围：10 ~ 220。
u32MinSensorGain	最小 Sensor gain(1024 等于 1 倍)。值域范围：1024 ~ 5242880。
u32MinISPGain	最小 ISP gain(1024 等于 1 倍)。值域范围：1024 ~ 262144。
u32MaxSensorGain	最大 Sensor gain(1024 等于 1 倍)。值域范围：1024 ~ 5242880。
u32MaxISPGain	最大 ISP gain(1024 等于 1 倍)。值域范围：1024 ~ 262144。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_AE_SetExposureLimit](#)(MI_U32 Channel, [MI_ISP_AE_EXPO_LIMIT_TYPE_t](#) *data);
MI_S32 [MI_ISP_AE_GetExposureLimit](#)(MI_U32 Channel, [MI_ISP_AE_EXPO_LIMIT_TYPE_t](#) *data);

3.1.50 MI_ISP_AE_EXPO_TABLE 结构体

3.1.50.1. 类型结构体 (Type Structure)

【说明】

曝光行程表。

【定义】

```
typedef struct MI_ISP_AE_EXPO_TABLE_TYPE_s
{
    MI_U32 u32NumPoints;
    EXPO_POINT_PARAM_t stExpoTbl[16];
} MI_ISP_AE_EXPO_TABLE_TYPE_t;
```

【名称】

变量名称	描述
u32NumPoints	曝光行程表的节点数。值域范围：1 ~ 16。
stExpoTbl[16]	曝光行程表的节点参数数据。

【注意事项】

无。

【相关结构类型与函式】

```
MI_S32 MI_ISP_AE_SetPlainLongExpoTable(MI_U32 Channel, MI_ISP_AE_EXPO_TABLE_TYPE_t *data);
MI_S32 MI_ISP_AE_GetPlainLongExpoTable(MI_U32 Channel, MI_ISP_AE_EXPO_TABLE_TYPE_t *data);
MI_S32 MI_ISP_AE_SetPlainShortExpoTable(MI_U32 Channel, MI_ISP_AE_EXPO_TABLE_TYPE_t *data);
MI_S32 MI_ISP_AE_GetPlainShortExpoTable(MI_U32 Channel, MI_ISP_AE_EXPO_TABLE_TYPE_t *data);
```

3.1.50.2. 曝光行程节点参数结构体 (EXPO_POINT_PARAM_t)

【说明】

设定曝光行程中各节点的曝光时间、光圈及增益值。

【定义】

```
typedef struct EXPO_POINT_PARAM_s
{
    MI_U32 u32FNumx10;
    MI_U32 u32Shutter;
    MI_U32 u32TotalGain;
    MI_U32 u32SensorGain;
} EXPO_POINT_PARAM_t;
```

【名称】

变量名称	描述
u32FNumx10	光圈值(ex: F1.8 等于 18)。值域范围: 10 ~ 220。
u32Shutter	快门时间(μ sec)。值域范围: 1 ~ 1000000。
u32TotalGain	总增益 = sensor 增益 × ISP 数字增益(1024 等于 1 倍)。值域范围: 1024 ~ 5242880。
u32SensorGain	Sensor 增益(1024 等于 1 倍)。值域范围: 1024 ~ 5242880。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.51 MI_ISP_AE_WIN_WEIGHT_MODE 结构体

3.1.51.1 类型结构体 (Type Structure)

【说明】

单独设定或切换画面亮度计算的权重表种类。

【定义】

```
typedef enum
```

```
{
    SS_AE_WEIGHT_AVERAGE = 0,
    SS_AE_WEIGHT_CENTER,
    SS_AE_WEIGHT_SPOT,
    SS_AE_WEIGHT_MAX
} MI_ISP_AE_WIN_WEIGHT_MODE_TYPE_e;
```

【名称】

变量名称	描述
SS_AE_WEIGHT_AVERAGE	切换至平均测光的权重表
SS_AE_WEIGHT_CENTER	切换至中央测光的权重表
SS_AE_WEIGHT_SPOT	切换至点状测光的权重表

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_AE_SetWinWgtType](#)(MI_U32 Channel, [MI_ISP_AE_WIN_WEIGHT_MODE_TYPE_e](#) *data);

MI_S32 [MI_ISP_AE_GetWinWgtType](#)(MI_U32 Channel, [MI_ISP_AE_WIN_WEIGHT_MODE_TYPE_e](#) *data);

3.1.52 MI_ISP_AE_WIN_WEIGHT 结构体

3.1.52.1. 类型结构体 (Type Structure)

【说明】

设定或切换画面亮度计算的权重表种类并设定权重表。

【定义】

```
typedef struct MI_ISP_AE_WIN_WEIGHT_TYPE_s
{
    MI\_ISP\_AE\_WIN\_WEIGHT\_MODE\_TYPE\_e eTypeID;
    WIN\_WEIGHT\_PARAM\_t stParaAPI;
} MI_ISP_AE_WIN_WEIGHT_TYPE_t;
```

【名称】

变量名称	描述
eTypeID	选择要切换过去的测光权重表种类。 平均测光: SS_AE_WEIGHT_AVERAGE 中央测光: SS_AE_WEIGHT_CENTER 点状测光: SS_AE_WEIGHT_SPOT
PARAM_t	设定曝光权重表。

【注意事项】

无。

【相关结构类型与函式】

```
MI_S32 MI\_ISP\_AE\_SetWinWgt(MI_U32 Channel, MI\_ISP\_AE\_WIN\_WEIGHT\_TYPE\_t *data);
MI_S32 MI\_ISP\_AE\_GetWinWgt(MI_U32 Channel, MI\_ISP\_AE\_WIN\_WEIGHT\_TYPE\_t *data);
```

3.1.52.2. 测光权重表参数结构体 (WIN_WEIGHT_PARAM_t)

【说明】

设定 32× 32 的测光权重表。

【定义】

```
typedef struct WIN_WEIGHT_PARAM_s
{
    MI_U8 u8AverageTbl[32× 32];
    MI_U8 u8CenterTbl[32× 32];
    MI_U8 u8SpotTbl[32× 32];
} WIN_WEIGHT_PARAM_t;
```


【名称】

变量名称	描述
u8AverageTbl[32× 32]	平均测光分块权重表。值域范围：0 ~ 255。
u8CenterTbl[32× 32]	中央测光分块权重表。值域范围：0 ~ 255。
u8SpotTbl[32× 32]	点状测光分块权重表。值域范围：0 ~ 255。

【注意事项】

- 最大支援尺寸为 32× 32，目前默认尺寸为 16× 16。

【相关结构类型与函式】

无。

3.1.53 MI_ISP_AE_FLICKER 结构体

3.1.53.1. 类型结构体 (Type Structure)

【说明】

设定抗频闪频率。

【定义】

```
typedef enum
```

```
{
    SS_AE_FLICKER_TYPE_DISABLE = 0,
    SS_AE_FLICKER_TYPE_60HZ = 1,
    SS_AE_FLICKER_TYPE_50HZ = 2,
    SS_AE_FLICKER_TYPE_AUTO = 3,
    SS_AE_FLICKER_TYPE_MAX
} MI_ISP_AE_FLICKER_TYPE_e;
```

【名称】

变量名称	描述
SS_AE_FLICKER_TYPE_DISABLE	关闭抗频闪功能。
SS_AE_FLICKER_TYPE_60HZ	设定频闪环境为 60Hz。
SS_AE_FLICKER_TYPE_50HZ	设定频闪环境为 50Hz。
SS_AE_FLICKER_TYPE_AUTO	自动设定频闪环境。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_AE_SetFlicker](#)(MI_U32 Channel, [MI_ISP_AE_FLICKER_TYPE_e](#) *data);

MI_S32 [MI_ISP_AE_GetFlicker](#)(MI_U32 Channel, [MI_ISP_AE_FLICKER_TYPE_e](#) *data);

3.1.54 MI_ISP_AE_STRATEGY 结构体

3.1.54.1. 类型结构体 (Type Structure)

【说明】

设定自动曝光之曝光策略。

【定义】

```
typedef struct MI_ISP_AE_STRATEGY_TYPE_s
```

```
{
    MI_ISP_AE_STRATEGY_TYPE_e eAESTrategyMode;
    MI_U32 u32Weighting;
    MI_ISP_AE_INTP_LUT_TYPE_t stUpperOffset;
    MI_ISP_AE_INTP_LUT_TYPE_t stLowerOffset;
    MI_U32 u32BrightToneStrength;
    MI_U32 u32BrightToneSensitivity;
    MI_U32 u32DarkToneStrength;
    MI_U32 u32DarkToneSensitivity;
    MI_U32 u32AutoStrength;
    MI_U32 u32AutoSensitivity;
} MI_ISP_AE_STRATEGY_TYPE_t;
```

【名称】

变量名称	描述
eAESTrategyMode	自动曝光策略之模式选择。
u32Weighting	自动曝光策略下计算出的画面目标亮度与 MI_ISP_AE_SetTarget 中所设定的画面目标亮度间的权重。 FinalAETarget = defaultAETarget × (1 - weighting) + StragetyTarget × weighting。
stUpperOffset	自动曝光策略向上浮动调整画面亮度目标的空间。 X 轴为环境亮度 BV、Y 轴为向上浮动空间(0 ~ 470)。
stLowerOffset	自动曝光策略向下浮动调整画面亮度目标的空间。 X 轴为环境亮度 BV、Y 轴为向下浮动空间(0 ~ 470)。
u32BrightToneStrength	eAESTrategyMode 设定为 SS_AE_STRATEGY_BRIGHTTONE 时, 可用来调整亮阶区域的目标亮度(数值愈大画面将愈暗)。 值域范围: 0 ~ 1024。
BrightToneSensitivity	eAESTrategyMode 设定为 SS_AE_STRATEGY_BRIGHTTONE 时, 可用来调整对亮阶区域的敏感度(数值愈大愈敏感, 即使影像中仅少部分区域过曝也会选择将画面压暗)。 值域范围: 0 ~ 1024。
DarkToneStrength	eAESTrategyMode 设定为 SS_AE_STRATEGY_DARKTONE 时, 可用来调整暗阶区域的目标亮度(数值愈大画面将愈亮)。 值域范围: 0 ~ 1024。

变量名称	描述
DarkToneSensitivity	eAESTrategyMode 设定为 SS_AE_STRATEGY_DARKTONE 时，可用来调整对暗阶区域的敏感度(数值愈大愈敏感，即使影像中仅少部分区域很暗也会选择将画面拉亮)。 值域范围：0 ~ 1024。
AutoStrength	eAESTrategyMode 设定为 SS_AE_STRATEGY_AUTO 时，可用来调整亮阶区域的目标亮度(数值愈大画面将愈暗)。 值域范围：0 ~ 1024。
AutoSensitivity	eAESTrategyMode 设定为 SS_AE_STRATEGY_AUTO 时，可用来调整对亮阶区域的敏感度(数值愈大愈敏感，即使影像中仅少部分区域过曝也会选择将画面压暗)。 值域范围：0 ~ 1024。

【注意事项】

如 stUpperOffset / stLowerOffset 之浮动区间设定为 0，则因无上下浮动的区间，AE 曝光策略的效果会失效，而画面目标亮度会等于 MI_ISP_AE_SetTarget 所设定的数值(default target)。

曝光策略浮动 target 的变化范围等于 default target - lower limit ~ default target + upperlimit。

【相关结构类型与函式】

MI_S32 [MI_ISP_AE_SetStrategy](#)(MI_U32 Channel, [MI_ISP_AE_STRATEGY_TYPE_t](#) *data);

MI_S32 [MI_ISP_AE_GetStrategy](#)(MI_U32 Channel, [MI_ISP_AE_STRATEGY_TYPE_t](#) *data);

3.1.54.2. 曝光策略模式结构体 ([MI_ISP_AE_STRATEGY_TYPE_e](#))

【说明】

选择曝光策略之模式。

【定义】

```
typedef enum
{
    SS_AE_STRATEGY_BRIGHTTONE,
    SS_AE_STRATEGY_DARKTONE,
    SS_AE_STRATEGY_AUTO,
    SS_AE_STRATEGY_MAX
} MI_ISP_AE_STRATEGY_TYPE_e;
```

【名称】

变量名称	描述
SS_AE_STRATEGY_BRIGHTTONE	强光抑制模式(亮阶优先)。
SS_AE_STRATEGY_DARKTONE	背光补偿模式(暗阶优先)。
SS_AE_STRATEGY_AUTO	自动补偿模式。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.55 MI_ISP_AWB_QUERY_INFO 结构体

3.1.55.1. 类型结构体 (Type Structure)

【说明】

设定当前整体白平衡信息的类型结构体。

【定义】

```
typedef struct MI_ISP_AWB_QUERY_INFO_TYPE_s
{
    MI_ISP_BOOL_e bIsStable;
    MI_U16         u16Rgain;
    MI_U16         u16Grgain;
    MI_U16         u16Gbgain;
    MI_U16         u16Bgain;
    MI_U16         u16ColorTemp;
    MI_U8          u8WPInd;
    MI_ISP_BOOL_e bMultiLSDetected;
    MI_U8          u8FirstLSInd;
    MI_U8          u8SecondLSInd;
} MI_ISP_AWB_QUERY_INFO_TYPE_t;
```

【名称】

变量名称	描述
bIsStable	获取目前 AWB 是否在收敛状态。
u16Rgain	获取目前设定的白平衡 R 通道增益。
u16Grgain	获取目前设定的白平衡 Gr 通道增益。
u16Gbgain	获取目前设定的白平衡 Gb 通道增益。
u16Bgain	获取目前设定的白平衡 B 通道增益。
u16ColorTemp	获取目前计算的色温值。
u8WPInd	当开启 WpWeight 功能，获取目前参考白点色温框 index。
bMultiLSDetected	当开启混光色偏校正功能，获取目前是否侦测出混光。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
u8FirstLSInd	当开启混光色偏校正功能，获取混光中数量第一大光源 index× 2。

变量名称	描述
u8SecondLSInd	当开启混光色偏校正功能，获取混光中数量第二大光源 index×2。

【注意事项】

- u8FirstLSInd 與 u8SecondLSInd 顯示的 index 值是實際色溫 Index 值乘以 2。也就是當 u8FirstLSInd 為 8 時，實際代表的色溫 index 是 $8 / 2 = 4$ (5000K)，當 u8FirstLSInd 為 9 時，實際代表的色溫 index 是 $9 / 2 = 4.5$ 其中 .5 的意思代表此光源色溫落在 index 4(5000K)與 5(4000K)之間。

【相关结构类型与函式】

MI_S32 [MI_ISP_AWB_QueryInfo](#)(MI_U32 Channel, [MI_ISP_AWB_QUERY_INFO_TYPE_t](#) *data);

3.1.56 MI_ISP_AWB_CT_STATISTICS 结构体

3.1.56.1. 类型结构体 (Type Structure)

【说明】

设定自动白平衡校正统计值的类型结构体。

【定义】

```
typedef struct MI_ISP_AWB_CT_STATISTICS_TYPE_s
{
    MI_U16 u16Width;
    MI_U16 u16Height;
    MI_U16 u16StatisX[5760];
    MI_U16 u16StatisY[5760];
} MI_ISP_AWB_CT_STATISTICS_TYPE_t;
```

【名称】

变量名称	描述
u16Width	有效区宽度。
u16Height	有效区高度。
u16StatisX[5760]	色温域的统计 X。
u16StatisY[5760]	色温域的统计 Y。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_AWB_GetCTStats](#)(MI_U32 Channel, [MI_ISP_AWB_CT_STATISTICS_TYPE_t](#) *data);

3.1.57 MI_ISP_AWB_HW_STATISTICS 结构体

3.1.57.1. 类型结构体 (Type Structure)

【说明】

设定 AWB 硬件统计值的类型结构体。

【定义】

```
typedef struct MI_ISP_AWB_HW_STATISTICS_TYPE_s
{
    MI_U8 u8AwbBuffer[34560];
} MI_ISP_AWB_HW_STATISTICS_TYPE_t;
```

【名称】

变量名称	描述
u8AwbBuffer	AWB 统计值资料，n = 34560。 其资料排列依序为{R ₀ , G ₀ , B ₀ , R ₁ , G ₁ , B ₁ , ..., R _{n-1} , G _{n-1} , B _{n-1} }。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_AWB_GetHWStats](#)(MI_U32 Channel, [MI_ISP_AWB_HW_STATISTICS_TYPE_t](#) *data);

3.1.58 MI_ISP_AWB_ATTR 结构体

3.1.58.1. 类型结构体 (Type Structure)

【说明】

设定 AWB 属性的类型结构体。

【定义】

```
typedef struct MI_ISP_AWB_ATTR_TYPE_s
{
    MI\_ISP\_SM\_STATE\_TYPE\_e    eState;
    MI\_ISP\_OP\_TYPE\_e          eOpType;
    MWB\_ATTR\_PARAM\_t         stManualParaAPI;
    AWB\_ATTR\_PARAM\_t         stAutoParaAPI;
} MI_ISP_AWB_ATTR_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定 AWB 属性功能的工作模式。 正常：SS_ISP_STATE_NORMAL = 0。 暂停：SS_ISP_STATE_PAUSE = 1。

变量名称	描述
enOpType	设定 AWB 属性的工作模式。 自动模式：SS_OP_TYP_AUTO = 0。 手动模式：SS_OP_TYP_MANUAL = 1。
stAuto	设定 AWB 属性的自动模式属性结构体。
stManual	设定 AWB 属性的手动模式属性结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_AWB_SetAttr](#)(MI_U32 Channel, [MI_ISP_AWB_ATTR_TYPE_t](#) *data);

MI_S32 [MI_ISP_AWB_GetAttr](#)(MI_U32 Channel, [MI_ISP_AWB_ATTR_TYPE_t](#) *data);

3.1.58.2 手动模式参数结构体 (Manual Parameter Structure)

【说明】

设定的手动模式参数结构体。

【定义】

```
typedef struct MWB_ATTR_PARAM_s
{
    MI_U16 u16Rgain;
    MI_U16 u16Grgain;
    MI_U16 u16Gbgain;
    MI_U16 u16Bgain;
} MWB_ATTR_PARAM_t;
```

【名称】

变量名称	描述
u16Rgain	白平衡 R 通道增益。值域范围：0 ~ 8192。
u16Grgain	白平衡 Gr 通道增益。值域范围：0 ~ 8192。
u16Gbgain	白平衡 Gb 通道增益。值域范围：0 ~ 8192。
u16Bgain	白平衡 B 通道增益。值域范围：0 ~ 8192。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.58.3. 自动模式参数结构体 (Automatic Parameter Structure)

【说明】

设定的自动模式参数结构体。

【定义】

```
typedef struct AWB_ATTR_PARAM_s
{
    MI_U8                u8Speed;
    MI_U8                u8ConvInThd;
    MI_U8                u8ConvOutThd;
    MI_U16               u16ZoneSel;
    MI_ISP_AWB_ALG_TYPE_e eAlgType;
    MI_U8                u8RGStrength;
    MI_U8                u8BGStrength;
    CT_LIMIT_PARAM_t     stCTLimit;
    CT_WEIGHT_PARAM_t     stLvWeight[MI_ISP_AWB_LV_CT_TBL_NUM];
    CT_RATIO_PARAM_t     stPreferRRatio[MI_ISP_AWB_LV_CT_TBL_NUM];
    CT_RATIO_PARAM_t     stPreferBRatio[MI_ISP_AWB_LV_CT_TBL_NUM];
    MI_ISP_BOOL_e        bWpWeightEnable;
    MI_U16               u16WpWeight[MI_ISP_AWB_CT_TBL_NUM];
    MS_U32               u4WeightWin[MI_ISP_AWB_WEIGHT_WIN_NUM];
} AWB_ATTR_PARAM_t;
```

【名称】

变量名称	描述
u8Speed	自动白平衡算法收敛速度，值越大收敛速度越快。 值域范围：0 ~ 100。
u8ConvInThd	设定白平衡的收敛区间，值越小则 AWB 收敛的与目标值越接近。 值域范围：0 ~ 255，不建议将值设太小，容易导致 AWB 一直处在不稳定状态，建议值 32。
u8ConvOutThd	设定要让白平衡离开收敛状态重新计算白平衡的阈值，值越大则 AWB 收敛稳定后，越不容易重新计算。 值域范围：0 ~ 255，不建议将值设太大，建议值 64。
u16ZoneSel	设定 AWB 属性功能的工作模式。 预设版：SS_AWB_ALG_DEFAULT = 0。 增强版：SS_AWB_ALG_ADVANCE = 1。 值 0：全局自动白平衡算法，整张影像的统计值都参与白平衡计算。主要提供给产线 Calibration 使用，例如：在产在线拍摄校正的光源时，取得 Rgain 和 Bgain 使用。 值 1：分色温区域算法，自动白平衡算法对所有统计值按色温区域分类，再计算 Gain 值。

变量名称	描述
eAlgType	自动白平衡算法的类型, 设定为 SS_AWB_ALG_ADVANCE 时, 可以开启 MI_ISP_AWB_SetAttrEx API 设定功能。
u8RGStrength	自动白平衡 R Gain 的强度。 等于 0x80 表示 R Gain 不变 大于 0x80 表示 R Gain 变大偏红 小于 0x80 表示 R Gain 变小
u8BGStrength	自动白平衡 B Gain 的强度。 等于 0x80 表示 B Gain 不变 大于 0x80 表示 B Gain 变大偏蓝 小于 0x80 表示 B Gain 变小
stCTLimit	白平衡算法环境色温, 不在色温曲线范围内时, 设定增益限制的 计算模式及手动增益值, 此参数不建议调整。此参数是限制 整个色温曲线坐标的范围, 超出范围的统计值则不纳入白平 衡算法计算。
stLvWeight[MI_ISP_AWB_LV_CT_TBL_NUM]	给定指定环境亮度下各色温的权重, 权重越高, 最后的目标 R、B gain 会越倾向该色温的 R、B gain, 共 18 组。 值域范围: 1 ~ 255。
stPreferRRatio[MI_ISP_AWB_LV_CT_TBL_NUM]	给定指定环境亮度下各色温的 R ratio, 用来微调目标 R gain, 共 18 组。值域范围: 1 ~ 255。
stPreferBRatio[MI_ISP_AWB_LV_CT_TBL_NUM]	给定指定环境亮度下各色温的 B ratio, 用来微调目标 B gain, 共 18 组。值域范围: 1 ~ 255。
bWpWeightEnable	设定 bWpWeightEnable 功能的布尔值。 关闭: SS_FALSE = 0。 开启: SS_TRUE = 1。 当开启时, u16WpWeight 才有作用, 且白平衡会找出一个参 考白点色温, 将其作灰; 当关闭时, 白平衡则是依画面中各色 温落点比例计算 R、B gain。
u16WpWeight[MI_ISP_AWB_CT_TBL_NUM]	参考白点权重, 权重越高的色温越容易被选为参考白点, 共 10 组。值域范围: 1 ~ 400, 默认值 100。
u4WeightWin[MI_ISP_AWB_WEIGHT_WIN_NUM]	将画面区分成 9x9 个区域, 可以分别给定各区域的权重, 让 白平衡计算更偏向将该区域做灰。值域范围: 0 ~ 16。

【注意事项】

- u16RefColorTemp、u16RefRgain、u16RefBgain 3 个参数是用来微调所有的色温曲线框位置。当相同的 Sensor 不同模组之间白平衡色温曲线变大时, 可以用此 3 个参数微调色温曲线, 调整的方式如下:
 1. 设置 u16ZoneSel = 0, 切换成全局自动白平衡算法。
 2. 拍摄指定色温的灰卡或白卡, 需涵盖整个画面, 建议使用 D50 环境。
 3. 呼叫 MI_ISP_AWB_QueryInfo 取得拍摄时的 R Gain 和 B Gain。
 4. 将指定的色温值和取得的 R Gain、B Gain 填入 u16RefColorTemp、u16RefRgain、u16RefBgain。
- u8RGStrength, u8BGStrength 调整建议: 依客户喜好调整, 例如: 某些场景下, 客户希望偏绿, 可以同时微调 u8RGStrength 和 u8BGStrength, 让这两个值同时小于 128。

【相关结构类型与函式】

无。

3.1.58.4. 参数结构体 (Parameter Structure)

【说明】

设定定义白平衡的增益范围限制的参数结构体。

【定义】

```
typedef struct CT_LIMIT_PARAM_s
{
    MI_U16 u16MaxRgain;
    MI_U16 u16MinRgain;
    MI_U16 u16MaxBgain;
    MI_U16 u16MinBgain;
} CT_LIMIT_PARAM_t;
```

【名称】

变量名称	描述
u16MaxRgain	设定高色温时的最大 R Gain。值域范围：256 ~ 4095。
u16MinRgain	设定低色温时的最小 R Gain。值域范围：256 ~ 4095。
u16MaxBgain	设定高色温时的最大 B Gain。值域范围：256 ~ 4095。
u16MinBgain	设定低色温时的最小 B Gain。值域范围：256 ~ 4095。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.58.5. 参数结构体 (Parameter Structure)

【说明】

设定计算目标 R、B gain 时各色温的权重的参数结构体。

【定义】

```
typedef struct CT_WEIGHT_PARAM_s
{
    MI_U16 u16Weight[MI_ISP_AWB_CT_TBL_NUM];
} CT_WEIGHT_PARAM_t;
```

【名称】

变量名称	描述
u16Weight[MI_ISP_AWB_CT_TBL_NUM]	计算目标 R、B gain 时各色温的权重，权重越大，目标 R、B gain 会越倾向该色温 R、B gain，MI_ISP_AWB_CT_TBL_NUM = 10。 值域范围：1 ~ 255，默认值 100。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.58.6. 参数结构体 (Parameter Structure)

【说明】

设定微调目标 R、B gain 所乘上的一个 ratio 的参数结构体。

【定义】

```
typedef struct CT_RATIO_PARAM_s
{
    MI_U16 u16Ratio[MI_ISP_AWB_CT_TBL_NUM];
} CT_RATIO_PARAM_t;
```

【名称】

变量名称	描述
u16Ratio [MI_ISP_AWB_CT_TBL_NUM]	用来微调各环境色温目标 R、B gain 所乘上的一个 ratio， MI_ISP_AWB_CT_TBL_NUM = 10。 值域范围：1 ~ 255，默认值 100。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.59 MI_ISP_AWB_ATTR_EX 结构体

3.1.59.1. 类型结构体 (Type Structure)

【说明】

设定 AWB 增强属性的类型结构体。

【定义】

```
typedef struct MI_ISP_AWB_ATTR_EX_TYPE_s
{
    MI_ISP_BOOL_e          bExtraLightEn;
    AWB_ATTR_EXTRA_LIGHTSOURCE_PARAM_t  stLightInfo[4];
} MI_ISP_AWB_ATTR_EX_TYPE_t;
```

【名称】

变量名称	描述
bExtraLightEn	设定自动白平衡计算时是否考虑使用者设的独立色温范围功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
stLightInfo[4]	用户设定的独立光源点信息。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_AWB_SetAttrEx](#)(MI_U32 Channel, [MI_ISP_AWB_ATTR_EX_TYPE_t](#) *data);
MI_S32 [MI_ISP_AWB_GetAttrEx](#)(MI_U32 Channel, [MI_ISP_AWB_ATTR_EX_TYPE_t](#) *data);

3.1.59.2. 白平衡独立光源点参数结构体(AWB Extar LightSource Parameter Structure)

【说明】

设定白平衡独立光源点的参数结构体。

【定义】

```
typedef struct AWB_ATTR_EXTRA_LIGHTSOURCE_PARAM_s
{
    MI_U16          u16WhiteRgain;
    MI_U16          u16WhiteBgain;
    MI_U8           u8AreaSize;
    MI_ISP_BOOL_e   bExclude;
} AWB_ATTR_EXTRA_LIGHTSOURCE_PARAM_t;
```

【名称】

变量名称	描述
u16WhiteRgain	此设定值会映像到色温坐标的某一个独立点，不需要参考色温，使用者设定的独立光源点的 R Gain 值($G \times 1024 / R$)。 值域范围：256 ~ 4095。

变量名称	描述
u16WhiteBgain	此设定值会映像到色温坐标的某一个独立点，不需要参考色温，使用者设定的独立光源点的 B Gain 值($G \times 1024 / B$)。 值域范围：256 ~ 4095。
u8AreaSize	使用者设定的独立光源点，在色温坐标的范围，值越大表示范围越广。值域范围：1 ~ 32。
bExclude	设定白平衡演算时排除或包含使用者设定的独立光源点功能的布尔值。在白平衡算法计算时，判定白点是否纳入或排除使用者设定的光源点范围。 包含：SS_FALSE = 0。 排除：SS_TRUE = 1。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.60 MI_ISP_AWB_MULTILS_LS 结构体

3.1.60.1. 类型结构体 (Type Structure)

【说明】

设定自动白平衡混光偏校正的类型结构体。

【定义】

```
typedef struct MI_ISP_AWB_MULTILS_LS_TYPE_s
{
    MI_ISP_BOOL_t bEnable;
    MI_U8          u8Sensitive;
    MI_U8          u8CaliStrength;
    MI_U16         u16CcmForLow[9];
    MI_U16         u16CcmForHigh[9];
} MI_ISP_AWB_MULTILS_LS_TYPE_t;
```

【名称】

变量名称	描述
bEnable	设定混光色偏校正功能的布尔值。 关闭：SS_FALSE = 0。 开启：SS_TRUE = 1。
u8Sensitive	判断混光的敏感度，值越大越容易判断为混光场景。 值域范围：1 ~ 10，默认值为 5。

变量名称	描述
u8CaliStrength	最大混光色温差距的色偏校正强度，最大混光差距表示两个光源的落点刚好分别落在 StartIdx 与 EndIdx。实际作用的强度会根据两光源色温差距与最大色温差距的比例做调整。 值域范围：0 ~ 100，默认值为 100。
u16CcmForLow[9]	当判断出混光，且将高色温光源做灰时，用来补正低色温光源处色偏的 CCM。注意 u16CcmForLow[3] ~ [5](G 列)是无作用的，修改请以 u16CcmForLow[0] ~ [2](R 列)及 u16CcmForLow[6] ~ [8](B 列)为主。值域范围：0 ~ 8191。
u16CcmForHigh[9]	当判断出混光，且将低色温光源做灰时，用来补正高色温光源处色偏的 CCM。注意 u16CcmForLow[3] ~ [5](G 列)是无作用的，修改请以 u16CcmForLow[0] ~ [2](R 列)及 u16CcmForLow[6] ~ [8](B 列)为主。值域范围：0 ~ 8191。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_AWB_SetMultiLSAttr](#)(MI_U32 Channel, [MI_ISP_AWB_MULTILS_LS_TYPE_t](#) *data);

MI_S32 [MI_ISP_AWB_GetMultiLSAttr](#)(MI_U32 Channel, [MI_ISP_AWB_MULTILS_LS_TYPE_t](#) *data);

3.1.61 MI_ISP_AWB_CT_CALI 结构体

3.1.61.1 类型结构体 (Type Structure)

【说明】

设定自动白平衡校正的类型结构体。

【定义】

```
typedef struct MI_ISP_AWB_CT_CALI_TYPE_s
{
    MI_U16 u16StartIdx;
    MI_U16 u16EndIdx;
    MI_U16 u16CtParams[40];
} MI_ISP_AWB_CT_CALI_TYPE_t;
```

【名称】

变量名称	描述
u16StartIdx	亮区开始索引。
u16EndIdx	亮区结束索引。
u16CtParams[40]	色温 cali 参数。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_AWB_SetCTCaliAttr](#)(MI_U32 Channel, [MI_ISP_AWB_CT_CALI_TYPE_t](#) *data);

MI_S32 [MI_ISP_AWB_GetCTCaliAttr](#)(MI_U32 Channel, [MI_ISP_AWB_CT_CALI_TYPE_t](#) *data);

3.1.62 MI_ISP_AF_QUERY_INFO 结构体

3.1.62.1 类型结构体 (Type Structure)

【说明】

设定当前整体自动对焦信息的类型结构体。

【定义】

```
typedef struct MI_ISP_AF_QUERY_INFO_TYPE_s  
{  
    AF\_STATS\_PARAM\_t stParaAPI;  
} MI_ISP_AF_QUERY_INFO_TYPE_t;
```

【名称】

变量名称	描述
stParaAPI	AF 统计值。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_AF_QueryInfo](#)(MI_U32 Channel, [MI_ISP_AF_QUERY_INFO_TYPE_t](#) *data);

3.1.62.2 参数结构体 (Parameter Structure)

【说明】

设定 AF 统计状态的参数结构体。

【定义】

```
typedef struct AF_STATS_PARAM_s
{
    MI_U32 u32IIR_1[10];
    MI_U32 :32;
    MI_U32 :32;
    MI_U32 u32IIR_2[10];
    MI_U32 :32;
    MI_U32 :32;
    MI_U32 u32Luma[10];
    MI_U32 :32;
    MI_U32 :32;
    MI_U32 u32FIR_V[10];
    MI_U32 :32;
    MI_U32 :32;
    MI_U32 u32FIR_H[10];
    MI_U32 :32;
    MI_U32 :32;
} AF_STATS_PARAM_t;
```

【名称】

变量名称	描述
iir_1[10]	分区间统计的 AF 水平方向 IIR 滤波器统计值。
iir_2[10]	分区间统计的 AF 水平方向 IIR 滤波器统计值。
luma[10]	分区间统计的 AF 亮度统计值。
FIR_v[10]	分区间统计的 AF 水平方向 FIR 滤波器统计值。
FIR_h[10]	分区间统计的 AF 垂直方向 FIR 滤波器统计值。

【注意事项】

无。

【相关结构类型与函式】

- iir_1[10]、iir_2[10]、FIR_v[10]、FIR_h[10] 每個 filter 都是 10 組，前面九個預設對應畫面的九宮格分割位置，第 10 個是自訂使用的，通常是拿來做 touch AF。IIR FIR 數值越高代表畫面就越清晰。Luma 是表示亮度。

3.1.63 MI_ISP_AF_WINDOW 结构体

3.1.63.1. 类型结构体 (Type Structure)

【说明】

设定自动对焦窗口的类型结构体。

【定义】

```
typedef struct MI_ISP_AF_WINDOW_TYPE_s
{
    MI_U8                u8WindowIndex;
    AF\_WINDOW\_PARAM\_t    stParaAPI;
} MI_ISP_AF_WINDOW_TYPE_t;
```

【名称】

变量名称	描述
u8WindowIndex	窗口 ID，一共支持 10 个窗口设定。值域范围：0 ~ 9。
stParaAPI	窗口坐标的参数结构体。

【注意事项】

无。

【相关结构类型与函式】

MI_S32 [MI_ISP_AF_SetWindow](#)(MI_U32 Channel, [MI_ISP_AF_WINDOW_TYPE_t](#) *data);
 MI_S32 [MI_ISP_AF_GetWindow](#)(MI_U32 Channel, [MI_ISP_AF_WINDOW_TYPE_t](#) *data);

3.1.63.2. 参数结构体 (Parameter Structure)

【说明】

设定自动对焦窗口的参数结构体。

【定义】

```
typedef struct AF_WINDOW_PARAM_s
{
    MI_U32 u32StartX;
    MI_U32 u32StartY;
    MI_U32 u32EndX;
    MI_U32 u32EndY;
} AF_WINDOW_PARAM_t;
```

【名称】

变量名称	描述
u32StartX	窗口开始点的 x 坐标。值域范围：0 ~ 1023。
u32StartY	窗口开始点的 y 坐标。值域范围：0 ~ 1023。
u32EndX	窗口结束点的 x 坐标。值域范围：0 ~ 1023。
u32EndY	窗口结束点的 y 坐标。值域范围：0 ~ 1023。

【注意事项】

无。

【相关结构类型与函式】

无。

3.1.64 MI_ISP_AF_FILTER 结构体

3.1.64.1. 类型结构体 (Type Structure)

【说明】

设定自动对焦硬件统计值 IIR/FIR 系数的类型结构体。

【定义】

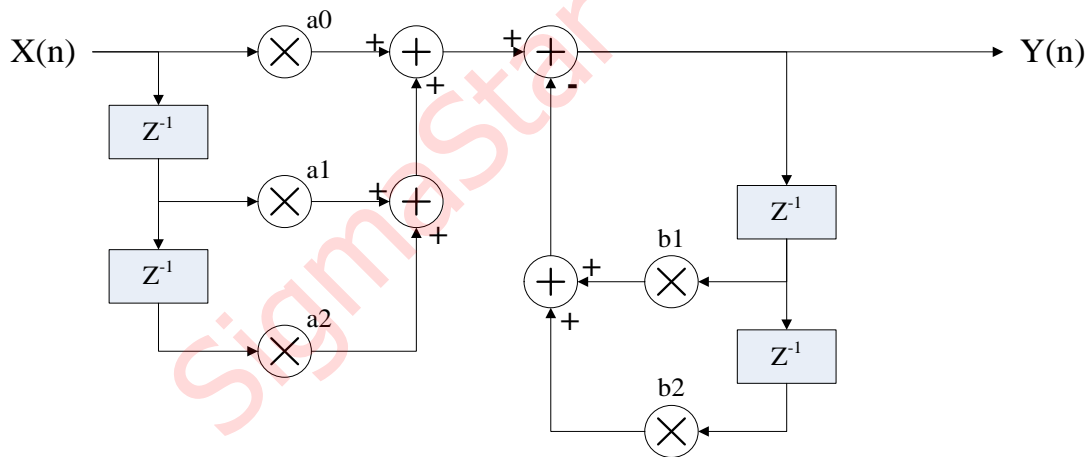
```
typedef struct MI_ISP_AF_FILTER_TYPE_s
{
    MI_U16 u16IIR1_a0;
    MI_U16 u16IIR1_a1;
    MI_U16 u16IIR1_a2;
    MI_U16 u16IIR1_b1;
    MI_U16 u16IIR1_b2;
    MI_U16 u16IIR1_1st_low_clip;
    MI_U16 u16IIR1_1st_high_clip;
    MI_U16 u16IIR1_2nd_low_clip;
    MI_U16 u16IIR1_2nd_high_clip;
    MI_U16 u16IIR2_a0;
    MI_U16 u16IIR2_a1;
    MI_U16 u16IIR2_a2;
    MI_U16 u16IIR2_b1;
    MI_U16 u16IIR2_b2;
    MI_U16 u16IIR2_1st_low_clip;
    MI_U16 u16IIR2_1st_high_clip;
    MI_U16 u16IIR2_2nd_low_clip;
    MI_U16 u16IIR2_2nd_high_clip;
} MI_ISP_AF_FILTER_TYPE_t;
```

【名称】

变量名称	描述
u16IIR1_a0	iir1 a0 乘法器，默认 63。
u16IIR1_a1	iir1 a1 乘法器，默认-126。
u16IIR1_a2	iir1 a2 乘法器，默认 63。
u16IIR1_b1	iir1 b1 乘法器，默认-109。
u16IIR1_b2	iir1 b2 乘法器，默认 45。

变量名称	描述
u16IIR1_1st_low_clip	iir1 X(n) input low clip, 默认 0。
u16IIR1_1st_high_clip	iir1 X(n) input high clip, 默认 320。
u16IIR1_2nd_low_clip	iir1 Y(n) output low clip, 默认 0。
u16IIR1_2nd_high_clip	iir1 Y(n) output high clip, 默认 1023。
u16IIR2_a0	iir2 a0 乘法器, 默认 63。
u16IIR2_a1	iir2 a1 乘法器, 默认 -126。
u16IIR2_a2	iir2 a2 乘法器, 默认 63。
u16IIR2_b1	iir2 b1 乘法器, 默认 -109。
u16IIR2_b2	iir2 b2 乘法器, 默认 45。
u16IIR2_1st_low_clip	iir2 X(n) input low clip, 默认 0。
u16IIR2_1st_high_clip	iir2 X(n) input high clip, 默认 320。
u16IIR2_2nd_low_clip	iir2 Y(n) output low clip, 默认 0。
u16IIR2_2nd_high_clip	iir2 Y(n) output high clip, 默认 1023。

【注意事项】



回应公式: $y(n) = a_0 x(n) + a_1 x(n-1) + a_2 x(n-2) - b_1 y(n-1) - b_2 y(n-2)$

Figure 5: IIR/FIR 系数

【相关结构类型与函式】

MI_S32 [MI_ISP_AF_SetFilter](#)(MI_U32 Channel, [MI_ISP_AF_FILTER_TYPE_t](#) *data);

MI_S32 [MI_ISP_AF_GetFilter](#)(MI_U32 Channel, [MI_ISP_AF_FILTER_TYPE_t](#) *data);

3.2. 错误码

参数名称	数值	描述
MI_ISP_OK	0	成功。
MI_ISP_FAILURE	-1	失败。

SigmaStar Confidential