SDK6 API Image Processing

Draft Version 0.8

April 21, 2016

Ambarella

I Contents

II Preface

This document provides technical details using a set of consistent typographical conventions to help the user differentiate key concepts at a glance.

Conventions include:

Example	Description
AmbaGuiGen, **DirectUSB** **Save**, **File > Save** **Power**, **Reset**, **Home**	Software names GUI commands and command sequences Computer / Hardware buttons
Flash_IO_control **da**, **status**, **enable**	Register names and register fields. For example, **Flash_IO_control** is the register for global control of Flash I/O, and bit 17 (**da)** is used for DMA acknowledgement.
GPIO81, **CLK_AU**	Hardware external pins
VIL, VIH, VOL, VOH	Hardware pin parameters
INT_O, RXDATA_I	Hardware pin signals
amb_performance_t ***amb_operating_mode_t*** **amb_set_operating_mode()**	API details (e.g., functions, structures, and type definitions)

`/usr/local/bin`
`success = amb_set_operating_mode (amb_XXX_base_address, & operating_mode)`
	User entries into software dialogues and GUI windows
	File names and paths
	Command line scripting and Code

Table II-1. Typographical Conventions for Technical Documents.

Additional Ambarella typographical conventions include:

- Acronyms are given in UPPER CASE using the default font (e.g., AHB, ARM11 and DDRIO).
- Names of Ambarella documents and publicly available standards, specifications, and databooks appear in *italic* type.

1 Overview

1.1 Overview: Introduction

This document defines the commands supported by image algorithm middleware (ImageProc). The chapters included are as follows:

- Chapter 2 "Data Initialization"
- Chapter 3 "Algorithm Control"
- Chapter 4 "Image Quality Control"
- Chapter 5 "AAA Statistics"

2 Data Initialization

2.1 Data Initialization: Overview

This chapter provides the Data Initialization APIs, image parameters, adjustment parameters, and AAA parameters.

Relevant information can be found in the following sections:

- Section 2.2 "Data Initialization: List of APIs"
- Section 2.3 "Data Initialization: Image Parameters"
- Section 2.4 "Data Initialization: Adjustment Parameters"
- Section 2.5 "Data Initialization: AAA Parameters"

2.2 Data Initialization: List of APIs

2.3 Data Initialization: Image Parameters

The SDK6 Image DSP pipeline could be configured by the application to achieve the desired Image Quality for specific image sensor and lens modules.
This section lists the APIs that initialize all necessary parameters for both video and still modes.

2.3.1 MW_IP_SET_IMG_PARAM_ADD

API Syntax:

MW_IP_SET_IMG_PARAM_ADD (UINT32 channelNo, UINT32 imgParamsAdd)

Function Description:

- This API provides the application with a mechanism to initialize the image DSP parameters which will be used inside the Image Proc. module.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For a single sensor application, the **channelNo** is 0.
UINT32	**imgParamsAdd**	The parameter is used to indicate the address of the image parameter table.

*Table 2-1. Parameters for Data Initialization API **MW_IP_SET_IMG_PARAM_ADD()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 2-2. Returns for Data Initialization API **MW_IP_SET_IMG_PARAM_ADD()**.*

Example:

None

See Also:

None

2.3.2 MW_IP_GET_IMG_PARAM_ADD

API Syntax:

MW_IP_GET_IMG_PARAM_ADD (UINT32 channelNo, UINT32 *imgParamsAdd)

Function Description:

- This API provides the application with get the image DSP parameters which will be used inside the Image Proc. module.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the Vin channel number. For the single sensor applications, the **channelNo** is 0.
UINT32	***imgParamsAdd**	The pointer is used to get the address of the image parameter table.

*Table 2-3. Parameters for Data Initialization API **MW_IP_GET_IMG_PARAM_ADD()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 2-4. Returns for Data Initialization API **MW_IP_GET_IMG_PARAM_ADD()**.*

Example:

None

See Also:

None

2.4 Data Initialization: Adjustment Parameters

To achieve the best image quality for various environments, the Image Proc Module auto-adjusts both the SDK6 Image DSP pipeline and the AAA algorithm. This section provides the API to initialize the tables used by the Auto-Adjust module. For a detailed description of each field, please refer to “Ambarella AN: Camera” for coverage on “Image Quality Tuning”.

2.4.1 MW_IP_SET_ADJ_PARAMS_ADD

API Syntax:

MW_IP_SET_ADJ_PARAMS_ADD (UINT32 channelNo, UINT8 adjType, UINT32 adjParamsAdd)

Function Description:

- This API is used to initialize the parameters for auto adjustment.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor applications, the **channelNo** is 0.
UINT8	**adjType**	**IQ_PARAMS_VIDEO_ADJ**: Video ADJ table **IQ_PARAMS_PHOTO_ADJ**: Photo preview ADJ table **IQ_PARAMS_STILL_LISO_ADJ**: Still LISO ADJ table **IQ_PARAMS_STILL_HISO_ADJ**: Still HISO ADJ table **IQ_PARAMS_STILL_IDX_INFO_ADJ**: Still parameter table
UINT32	**adjParamsAdd**	The parameter is used to indicate the address of the adj parameter table.

*Table 2-5. Parameters for Data Initialization API **MW_IP_SET_ADJ_PARAMS_ADD()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 2-6. Returns for Data Initialization API **MW_IP_SET_ADJ_PARAMS_ADD()**.*

Example:

None

See Also:

None

2.4.2 MW_IP_GET_ADJ_PARAMS_ADD

API Syntax:

MW_IP_GET_ADJ_PARAMS_ADD (UINT32 channelNo, UINT8 adjType, UINT32 *adjParamsAdd)

Function Description:

- This API is used to get the specified adjustment table address.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor applications, the **channelNo** is 0.
UINT8	**adjType**	**IQ_PARAMS_VIDEO_ADJ**: Video ADJ table **IQ_PARAMS_PHOTO_ADJ**: Photo preview ADJ table **IQ_PARAMS_STILL_LISO_ADJ**: Still LISO ADJ table **IQ_PARAMS_STILL_HISO_ADJ**: Still HISO ADJ table **IQ_PARAMS_STILL_IDX_INFO_ADJ**: Still parameter table
UINT32	***adjParamsAdd**	The pointer is used to get the address of the adj parameter table.

*Table 2-7. Parameters for Data Initialization API **MW_IP_GET_ADJ_PARAMS_ADD()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 2-8. Returns for Data Initialization API **MW_IP_GET_ADJ_PARAMS_ADD()**.*

Example:

None

See Also:

None

2.4.3 MW_IP_ADJ_AWBAE_CONTROL

API Syntax:

MW_IP_ADJ_AWBAE_CONTROL (UINT32 channelNo, ADJ_IQ_INFO_s *pAdjVideoIqInfo)

Function Description:

- This API is used to get the adjustment table number per different inputs, **ev_index** and input **wb_gain**.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For single sensor applications, the **channelNo** is 0.
ADJ_IQ_INFO_s	***pAdjVideoIqInfo**	Pointer to the Video IQ Information Table. Please refer to Section 2.4.3.1 below for the definition.

*Table 2-9. Parameters for Data Initialization API **MW_IP_ADJ_AWBAE_CONTROL()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 2-10. Returns for Data Initialization API **MW_IP_ADJ_AWBAE_CONTROL()**.*

Example:

None

See Also:

None

2.4.3.1 MW_IP_ ADJ_AWBAE_CONTROL > ADJ_IQ_INFO_s

Type	Field	Description
UINT8	**Mode**	Input mode, **IP_PREVIEW_MODE**: Preiview **IP_CAPTURE_MODE**: Capture **IP_PREFLASH_MODE**: Preflash
AMBA_AE_INFO_s	**Ae**	Input **AE** information. Please refer to Section 2.4.3.2 below for the definition.

Type	Field	Description
AMBA_DSP_ IMG_WB_ GAIN_s	**Wb**	Input **Wb** gain.
UINT16	**DZoomStep**	Input Dzoom step.
UINT32	**AwbAeParamAdd**	Input the address of **ADJ_AWB_AE_s**.
UINT32	**ColorParamAdd**	Input the address of **COLOR_3D_s**.
UINT32	**FilterParamAdd**	Input the address of **ADJ_VIDEO_PARAM_s**.
UINT16	**AdjTableNo**	Input the ADJ table number.

*Table 2-11. Definition of **ADJ_IQ_INFO_s** for API **MW_IP_ADJ_AWBAE_CONTROL()**.*

2.4.3.2 MW_IP_ ADJ_AWBAE_CONTROL > AMBA_AE_INFO_s

Type	Field	Description
UINT16	**EvIndex**	Input **EvIndex**.
UINT16	**NfIndex**	Input **NfIndex**.
INT16	**ShutterIndex**	Input **ShutterIndex**.
INT16	**AgcIndex**	Input **AgcIndex**.
INT16	**IrisIndex**	Input **IrisIndex**.
INT16	**Dgain**	Input **Dgain**.
UINT16	**IsoValue**	Input **IsoValue**.
UINT16	**Flash**	Input **Flash** mode.
UINT16	**Mode**	Input still mode, such as LISO or HISO.
float	**ShutterTime**	Exposure time in seconds
float	**AgcGain**	Gain in factor

*Table 2-12. Definition of **AMBA_AE_INFO_s** for API **MW_IP_ADJ_AWBAE_CONTROL()**.*

2.4.4 MW_IP_ADJ_VIDEO_CONTROL

API Syntax:

MW_IP_ADJ_VIDEO_CONTROL (UINT32 channelNo, , ADJ_IQ_INFO_s *pAdjVideoIqInfo)

Function Description:

- This API is used to calculate the interpolated ADJ parameters from the input Adj table per different inputs such as **EvIndex**, **NfIndex**, **WbGain**, and D.zoom step for the video mode.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
ADJ_IQ_INFO_s	***pAdjVideoIqInfo**	Pointer to the video IQ information table.

Table 2-13. Parameters for Data Initialization API ***MW_IP_ADJ_VIDEO_CONTROL()***.

Returns:

Return	Description
0	Success
-1	Failure

Table 2-14. Returns for Data Initialization API ***MW_IP_ADJ_VIDEO_CONTROL()***.

Example:

None

See Also:

None

2.4.5 MW_IP_ADJ_STILL_CONTROL

API Syntax:

MW_IP_ADJ_STILL_CONTROL (UINT32 channelNo, ADJ_STILL_CONTROL_s *pAdjStillControl)

Function Description:

- This API is used to calculate the interpolated ADJ parameters per different input such as **EvIndex**, **NfIndex**,**WbGain**, D.zoom step, and flash mode for the still mode.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
ADJ_STILL_CONTROL_s	*** pAdjStillControl**	Pointer to the still control adjustment detail. Please refer to Section 2.4.5.1 below for the definition.

*Table 2-15. Parameters for Data Initialization API **MW_IP_ADJ_STILL_CONTROL()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 2-16. Returns for Data Initialization API **MW_IP_ADJ_STILL_CONTROL()**.*

Example

None

See Also:

None

2.4.5.1 MW_IP_ADJ_STILL_CONTROL > ADJ_STILL_CONTROL_s

Type	Field	Description
UINT8	**StillMode**	Input Still mode, such as LISO, HISO. Currently 0 is the LISO mode.
UINT16	**ShIndex**	Input **sht_index**
UINT16	**EvIndex**	Input **ev_index**.
UINT16	**NfIndex**	Input **nf_index**.
AMBA_DSP_ IMG_WB_ GAIN_s	**WbGain**	Input **wb_gain**.
UINT16	**DZoomStep**	Input D.zoom step.
UINT8	**FlashMode**	Input flash mode.
UINT8	**LutNo**	Input table number.

*Table 2-17. Definition of **ADJ_STILL_CONTROL_s** for Data Initialization API **MW_IP_ADJ_STILL_CONTROL()**.*

2.4.6 MW_IP_CHK_IQ_PARAM_VER

API Syntax:

MW_IP_CHK_IQ_PARAM_VER (UINT32 channelNo, int type, UINT32 tableAddr)

Function Description:

- This API is used to check the structure version of a specific type of the IQ parameter table. If the structure version of a certain IQ paramter table is not the same as the current SDK, then the system will be asserted.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor applications, the **channelNo** is 0.
int	**Type**	**IQ_PARAMS_IMG_DEF**: Image parameter table **IQ_PARAMS_VIDEO_ADJ**: Video ADJ table **IQ_PARAMS_PHOTO_ADJ**: Photo preview ADJ table **IQ_PARAMS_STILL_LISO_ADJ**: Still LISO ADJ table **IQ_PARAMS_STILL_HISO_ADJ**: Still HISO ADJ table **IQ_PARAMS_AAA**: AAA parameter table **IQ_PARAMS_STILL_IDX_INFO_ADJ**: Still parameter table
UINT32	**tableAddr**	The address of the IQ parameter table

*Table 2-18. Parameters for Data Initialization API **MW_IP_CHK_IQ_PARAM_VER()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 2-19. Returns for Data Initialization API **MW_IP_CHK_IQ_PARAM_VER()**.*

Example:

None

See Also:

None

2.4.7 MW_IP_ADJ_VIDEO_HDR_INIT

API Syntax:

MW_IP_ADJ_VIDEO_HDR_INIT (UINT32 chNo, HDR_INFO_s *hdrInfo)

Function Description:

- This API is used to initialize the HDR auto-adjust module according to AE/AWB settings and IQ parameters.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor applications, the **channelNo** is 0.
HDR_INFO_s	***hdrInfo**	Pointer to the hdr adjustment detail. Please refer to Section 2.4.7.1 below for the definition.

*Table 2-20.   Parameters for Data Initialization API **MW_IP_ADJ_VIDEO_HDR_INIT()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 2-21.   Returns for Data Initialization API **MW_IP_ADJ_VIDEO_HDR_INIT()**.*

Example:

```
{
     HDR_INFO_s                HdrInfo;

     memset(&HdrInfo, 0, sizeof(HdrInfo));

     AmbaImg_Proc_Cmd(MW_IP_GET_AE_INFO, 0, IP_MODE_VIDEO, (UINT32)&HdrInfo.
AeInfo[0]); //long
     AmbaImg_Proc_Cmd(MW_IP_GET_AE_INFO, 1, IP_MODE_VIDEO, (UINT32)&HdrInfo.
AeInfo[1]); //short

     AmbaImg_Proc_Cmd(MW_IP_GET_PIPE_WB_GAIN, 0, IP_MODE_VIDEO,
(UINT32)&HdrInfo.WbGain[0]); //long
     AmbaImg_Proc_Cmd(MW_IP_GET_PIPE_WB_GAIN, 0, IP_MODE_VIDEO,
(UINT32)&HdrInfo.WbGain[1]); //short

     AmbaImg_Proc_Cmd(MW_IP_ADJ_VIDEO_HDR_INIT, 0, (UINT32)&HdrInfo, 0);
}
```

See Also:

MW_IP_GET_AE_INFO()
MW_IP_GET_PIPE_WB_GAIN()

2.4.7.1 MW_IP_ADJ_VIDEO_HDR_INIT > HDR_INFO_s

Type	Field	Description
AMBA_AE_ INFO_s	**AEINFO[3]**	Input Ae settings 0: Long, 1: Short 2: Very short
AMBA_DSP_ IMG_WB_ GAIN_s	**WBGAIN[3]**	Input Awb settings 0: Long 1: Short 2: Very short

*Table 2-22. Definition of **HDR_INFO_s** for Data Initialization API **MW_IP_ADJ_VIDEO_HDR_INIT()**.*

2.4.8 MW_IP_ADJ_VIDEO_HDR_CONTROL

API Syntax:

MW_IP_ADJ_VIDEO_HDR_CONTROL (UINT32 chNo, HDR_INFO_s *hdrInfo)

Function Description:

- This API is is the entry point of HDR auto-adjust module.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor applications, the **channelNo** is 0.
HDR_INFO_s	***hdrInfo**	Pointer to the hdr adjustment detail. Please refer to Section 2.4.7.1 for the definition.

*Table 2-23. Parameters for Data Initialization API **MW_IP_ADJ_VIDEO_HDR_CONTROL()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 2-24. Returns for Data Initialization API **MW_IP_ADJ_VIDEO_HDR_CONTROL()**.*

Example:

```
{
     HDR_INFO_s               HdrInfo;

     memset(&HdrInfo, 0, sizeof(HdrInfo));

     AmbaImg_Proc_Cmd(MW_IP_GET_AE_INFO, 0, IP_MODE_VIDEO, (UINT32)&HdrInfo.
AeInfo[0]); //long
     AmbaImg_Proc_Cmd(MW_IP_GET_AE_INFO, 1, IP_MODE_VIDEO, (UINT32)&HdrInfo.
AeInfo[1]); //short

     AmbaImg_Proc_Cmd(MW_IP_GET_PIPE_WB_GAIN, 0, IP_MODE_VIDEO,
(UINT32)&HdrInfo.WbGain[0]); //long
     AmbaImg_Proc_Cmd(MW_IP_GET_PIPE_WB_GAIN, 0, IP_MODE_VIDEO,
(UINT32)&HdrInfo.WbGain[1]); //short

     AmbaImg_Proc_Cmd(MW_IP_ADJ_VIDEO_HDR_CONTROL, 0, (UINT32)&HdrInfo, 0);
}
```

See Also:

MW_IP_GET_AE_INFO()
MW_IP_GET_PIPE_WB_GAIN()

2.5 Data Initialization: AAA Parameters

Image Proc Module exposes all parameters for the control of the AAA Algorithms.

2.5.1 MW_IP_SET_AAA_PARAM

API Syntax:

MW_IP_SET_AAA_PARAM (UINT32 channelNo, AAA_PARAM_s * aaaDefParams)

Function Description:

- This API provides the application with a mechanism to initialize the AAA algorithm parameters which will be called inside the Image Proc. module.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AAA_PARAM_s *	**aaaDefParams**	Pointer to structure with AAA parameters.

*Table 2-25. Parameters for Data Initialization API **MW_IP_SET_AAA_PARAM()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 2-26. Returns for Data Initialization API **MW_IP_SET_AAA_PARAM()**.*

Example:

None

See Also:

For the detailed definition of **AAA_PARAM_s**, please refer to Anxxxxx.

2.5.2 MW_IP_GET_AAA_PARAM

API Syntax:

MW_IP_GET_AAA_PARAM (UINT32 channelNo, AAA_PARAM_s * aaaDefParams)

Function Description:

- This API is used to get the AAA parameters.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AAA_PARAM_s *	**aaaDefParams**	Pointer to the structure with AAA parameters.

Table 2-27. Parameters for Data Initialization API ***MW_IP_GET_AAA_PARAM ()***.

Returns:

Return	Description
0	Success
-1	Failure

Table 2-28. Returns for Data Initialization API ***MW_IP_GET_AAA_PARAM()***.

Example:

None

See Also:

MW_IP_SET_AAA_PARAM()

3 Algorithm Control

3.1 Algorithm Control: Introduction

This chapter provides the functions for the AAA algorithm, AE algorithm, and AWB algorithm.

3.2 Algorithm Control: List of APIs

3.3 Algorithm Control: AAA Algorithm

This section describes all commands to control the behavior of the AAA in the Image Proc Module (IP).

3.3.1 MW_IP_REGISTER_FUNC

API Syntax:

MW_IP_REGISTER_FUNC (UINT32 channelNo, IMG_PROC_FUNC_s *pIpFunc)

Function Description:

- This command provides the application with a mechanism to register call back functions for its own algorithms.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
IMG_PROC_FUNC_s *	**pIpFunc**	Pointer to the structure of IP callback functions. Please refer to Section 3.3.1.1 for more details.

Table 3-1. Parameters for Algorithm Control API ***MW_IP_REGISTER_FUNC()***.

Returns:

Return	Description
0	Success
-1	Failure

Table 3-2. Returns for Algorithm Control API ***MW_IP_REGISTER_FUNC()***.

Example:

```
{
    IMG_PROC_FUNC_s IpFuncTmp = {NULL, NULL, NULL, NULL, NULL, NULL, NULL};
    IpFuncTmp.AeAwbAdj_Init    = Amba_AeAwbAdj_Init;
    IpFuncTmp.AeAwbAdj_Control = Amba_AeAwbAdj_Control;
    IpFuncTmp.Ae_Ctrl  =  Amba_Ae_Ctrl;
    IpFuncTmp.Awb_Ctrl =  Amba_Awb_Ctrl;
    IpFuncTmp.Adj_Ctrl =  Amba_Adj_Ctrl;
    IpFuncTmp.QueryActualShutterTime =  _QueryActualShutterTime;
    IpFuncTmp.QueryActualGainFactor =  _QueryActualGainFactor;
    AmbaImg_Proc_Cmd(MW_IP_REGISTER_FUNC, 0, (UINT32)&IpFuncTmp, 0);
}
```

See Also:

None

3.3.1.1 MW_IP_REGISTER_FUNC > IMG_PROC_FUNC_s

Type	Field	Description
void	**(* AeAwbAdj_Init)(void *hdlr,UINT8 initFlg,AMBA_KAL_BYTE_POOL_T *PMMPL)**	0: AAA 1: AE only 2: AWB only 3: ADJ only
void	**(*AeAwbAdj_Control)(void *hdlr)*)**	Callback of the **AeAwbAdj_Control** function for the customer to register.
void	**(* Ae_Ctrl)(void *hdlr)**	Callback of the **Ae_Ctrl** function for the customer to register.
void	**(* Awb_Ctrl)(void *hdlr)**	Callback of the **Awb_Ctrl** function for the customer to register.
void	**(* Adj_Ctrl)(void *hdlr)**	Callback of the **Adj_Ctrl** function for the customer to register.
int	**(* QUERYACTUALSHUTTERTIME)(* QUERYACTUALSHUTTERTIME)(UINT32 MAINVIEWID, UINT32 EXPOSUREFRAME, FLOAT *DESIREDSHUTTER, AMBA_IMG_SENSOR_SHUTTER_INFO_S *ACTUALSHUTTER);**	Callback of the QueryActualShutterTime function for customer to register.
int	**(* QUERYACTUALGAINFACTOR)(UINT32 MAINVIEWID, UINT32 EXPOSUREFRAME, AMBA_IMG_SENSOR_GAIN_INFO_S *DESIREDGAIN, AMBA_IMG_SENSOR_GAIN_INFO_S *ACTUALGAIN)**	Callback of the QueryActualGainFactor function for customer to register.

*Table 3-3. Definition of **IMG_PROC_FUNC_s** for Algorithm Control API **MW_IP_REGISTER_FUNC()**.*

3.3.2 MW_IP_UNREGISTER_FUNC

API Syntax:

MW_IP_UNREGISTER_FUNC (UINT32 channelNo)

Function Description:

- This command provides the application to unregister call back functions for its own algorithms.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.

*Table 3-4. Parameters for Algorithm Control API **MW_IP_UNREGISTER_FUNC()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-5. Returns for Algorithm Control API **MW_IP_UNREGISTER_FUNC()**.*

Example:

None

See Also:

None

3.3.3 MW_IP_GET_REG_FUNC

API Syntax:

MW_IP_GET_REG_FUNC (UINT32 channelNo, IMG_PROC_FUNC_s *pIpFunc)

Function Description:

- This command provides the application to get the registered call back functions for its own algorithms.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
IMG_PROC_FUNC_s *	**pIpFunc**	Pointer to the structure of IP callback functions.

*Table 3-6. Parameters for Algorithm Control API **MW_IP_GET_REG_FUNC()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-7. Returns for Algorithm Control API **MW_IP_GET_REG_FUNC()**.*

Example:

None

See Also:

None

3.3.4 MW_IP_GET_3A_STATUS

API Syntax:

MW_IP_GET_3A_STATUS (UINT32 channelNo, AMBA_3A_STATUS_s *aaaVideoStatus, AMBA_3A_STATUS_s *aaaStillStatus)

Function Description:

- This API is used to get AAA status for both the video mode and the still mode.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AMBA_3A_STATUS_s *	**aaaVideoStatus**	Please refer to Section 3.3.4.1 for more details.
AMBA_3A_STATUS_s *	**aaaStillStatus**	Please refer to Section 3.3.4.1 for more details.

*Table 3-8. Parameters for Algorithm Control API **MW_IP_GET_3A_STATUS()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-9. Returns for Algorithm Control API **MW_IP_GET_3A_STATUS()**.*

Example:

None

See Also:

None

3.3.4.1 MW_IP_GET_3A_STATUS > AMBA_3A_STATUS_s

Type	Field	Description
UINT8	**Ae**	**AMBA_LOCK**: AE lock **AMBA_PROCESSING**: AE running **AMBA_IDLE**: AE IDLE
UINT8	**Awb**	**AMBA_LOCK**: AWB lock **AMBA_PROCESSING**: AWB running **AMBA_IDLE**: AWB IDLE
UINT8	**Af**	**AMBA_LOCK**: AF lock **AMBA_PROCESSING**: AF running **AMBA_IDLE**: AF IDLE

*Table 3-10. Definition of **AMBA_3A_STATUS_s** for Algorithm Control API **MW_IP_GET_3A_STATUS()**.*

3.3.5 MW_IP_SET_3A_STATUS

API Syntax:

MW_IP_SET_3A_STATUS (UINT32 channelNo, AMBA_3A_STATUS_s *aaaVideoStatus, AMBA_3A_STATUS_s *aaaStillStatus)

Function Description:

- This API set the current AAA status for both video mode and still modes.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AMBA_3A_STATUS_s *	**aaaVideoStatus**	Please refer to Section 3.3.4.1 for more details.
AMBA_3A_STATUS_s *	**aaaStillStatus**	Please refer to Section 3.3.4.1 for more details.

*Table 3-11. Parameters for Algorithm Control API **MW_IP_SET_3A_STATUS()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-12. Returns for Algorithm Control API **MW_IP_SET_3A_STATUS()**.*

Example:

None

See Also:

None

3.3.6 MW_IP_SET_AAA_OP_INFO

API Syntax:

MW_IP_SET_AAA_OP_INFO (UINT32 channelNo, AMBA_3A_OP_INFO_s* pAaaOpInfo)

Function Description:

- This API sets the AAA operation mode as well as the Ambarella auto adjust noise control operation mode.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AMBA_3A_OP_INFO_s *	**pAaaOpInfo**	Pointer to the AAA operation information. Please refer to Section 3.3.6.1 for more details.

Table 3-13. Parameters for Algorithm Control API ***MW_IP_SET_AAA_OP_INFO()****.*

Returns:

Return	Description
0	Success
-1	Failure

Table 3-14. Returns for Algorithm Control API ***MW_IP_SET_AAA_OP_INFO()****.*

Example:

None

See Also:

None

3.3.6.1 MW_IP_SET_AAA_OP_INFO > AMBA_3A_OP_INFO_s

Type	Field	Description
UINT8	**AeOp**	Auto exposure mode: 0: Off 1: On
UINT8	**AwbOp**	Auto white balancing mode: 0: Off 1: On
UINT8	**AfOp**	Auto focus mode: 0: Off 1: On
UINT8	**AdjOp**	Auto adjust mode: 0: Off 1: On
UINT8	**RESERVED[4]**	Reserved for future use.

*Table 3-15. Definition of **AMBA_3A_OP_INFO_s** for Algorithm Control API **MW_IP_SET_AAA_OP_INFO()**.*

3.3.7 MW_IP_GET_AAA_OP_INFO

API Syntax:

MW_IP_GET_AAA_OP_INFO (UINT32 channelNo, AMBA_3A_OP_INFO_s* pAaaOpInfo)

Function Description:

- This API returns the AAA operation mode as well as the Ambarella auto adjust noise control operation mode.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For single sensor application, the channelNo is 0.
AMBA_3A_OP_INFO_s *	**pAaaOpInfo**	Pointer to the AAA operation information. Please refer to Section 3.3.6.1 for more details.

Table 3-16. Parameters for Algorithm Control API ***MW_IP_GET_AAA_OP_INFO()****.*

Returns:

Return	Description
0	Success
-1	Failure

Table 3-17. Returns for Algorithm Control API ***MW_IP_GET_AAA_OP_INFO()****.*

Example:

None

See Also:

MW_IP_SET_AAA_OP_INFO()

3.3.8 MW_IP_AMBA_AEAWBADJ_INIT

API Syntax:

MW_IP_AMBA_AEAWBADJ_INIT (UINT32 channelNo, UINT8 aaaFlg)

Function Description:

- This API is used to init AE or AWB or ADJ algorithms.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the channelNo is 0.
UINT8	**aaaFlg**	0: Init all algo 1: Init ae algo 2: Init awb algo 3: Init adj algo

*Table 3-18. Parameters for Algorithm Control API **MW_IP_AMBA_AEAWBADJ_INIT()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-19. Returns for Algorithm Control API **MW_IP_AMBA_AEAWBADJ_INIT()**.*

Example

None

See Also:

None

3.3.9 MW_IP_GET_FRAME_RATE

API Syntax:

MW_IP_GET_FRAME_RATE (UINT32 channelNo, UINT32 *frameRate, UINT32 *frameRatex1000)

Function Description:

- This API is used to get the main frame rate for certain channels.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the channelNo is 0.
UINT32	*** frameRate**	Pointer to the frame rate
UINT32	***frameRatex1000**	Pointer to the frame rate x 1000

*Table 3-20. Parameters for Algorithm Control API **MW_IP_GET_FRAME_RATE()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-21. Returns for Algorithm Control API **MW_IP_GET_FRAME_RATE()**.*

Example

None

See Also:

None

3.3.10 MW_IP_SET_FRAME_RATE

API Syntax:

MW_IP_SET_FRAME_RATE (UINT32 channelNo, UINT32 frameRate, UINT32 frameRatex1000)

Function Description:

- This API is used to set the main frame rate for certain channels.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT32	**frameRate**	The frame rate
UINT32	**frameRatex1000**	The frame rate x 1000

Table 3-22. Parameters for Algorithm Control API ***MW_IP_SET_FRAME_RATE()***.

Returns:

Return	Description
0	Success
-1	Failure

Table 3-23. Returns for Algorithm Control API ***MW_IP_SET_FRAME_RATE()***.

Example

None

See Also:

None

3.3.11 MW_IP_GET_CURR_FRAME_RATE

API Syntax:

MW_IP_GET_CURR_FRAME_RATE (UINT32 channelNo, UINT32 *frameRate, UINT32 *frameRatex1000)

Function Description:

- This API is used to get the current frame rate for certain channels.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT32	*** frameRate**	Pointer to the frame rate
UINT32	*** frameRatex1000**	Pointer to the frame rate x 1000

*Table 3-24. Parameters for Algorithm Control API **MW_IP_GET_CURR_FRAME_RATE()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-25. Returns for Algorithm Control API **MW_IP_GET_CURR_FRAME_RATE()**.*

Example

None

See Also:

None

3.3.12 MW_IP_SET_CURR_FRAME_RATE

API Syntax:

MW_IP_SET_CURR_FRAME_RATE (UINT32 channelNo, UINT32 frameRate, UINT32 frameRatex1000)

Function Description:

- This API is used to set the current frame rate for certain channels.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT32	**frameRate**	The frame rate
UINT32	**frameRatex1000**	The frame rate x 1000

*Table 3-26. Parameters for Algorithm Control API **MW_IP_SET_CURR_FRAME_RATE()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-27. Returns for Algorithm Control API **MW_IP_SET_CURR_FRAME_RATE()**.*

Example

None

See Also:

None

3.3.13 MW_IP_CHK_PHOTO_PREVIEW

API Syntax:

MW_IP_CHK_PHOTO_PREVIEW (UINT8 *photoPreview)

Function Description:

- This API is used to check if it is the photo preview or not.

Parameters:

Type	Parameter	Description
UINT8	***photoPreview**	Pointer to photoPreview. 0: Video preview 1: Photo preview

*Table 3-28. Parameters for Algorithm Control API **MW_IP_CHK_PHOTO_PREVIEW()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-29. Returns for Algorithm Control API **MW_IP_CHK_PHOTO_PREVIEW()**.*

Example

None

See Also:

None

3.3.14 MW_IP_SET_PHOTO_PREVIEW

API Syntax:

MW_IP_SET_PHOTO_PREVIEW (UINT8 photoPreview)

Function Description:

- This API is used to set the photo preview or the video preview.

Parameters:

Type	Parameter	Description
UINT8	**photoPreview**	photoPreview. 0: Video preview 1: Photo preview

*Table 3-30. Parameters for Algorithm Control API **MW_IP_SET_PHOTO_PREVIEW()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-31. Returns for Algorithm Control API **MW_IP_SET_PHOTO_PREVIEW()**.*

Example

None

See Also:

None

3.3.15 MW_IP_GET_VIDEO_HDR_ENABLE

API Syntax:

MW_IP_GET_VIDEO_HDR_ENABLE (UINT32 channelNo, UINT8 *enable)

Function Description:

- This API is used to get the current video hdr enable status.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT8	*** enable**	Pointer to HDR enable. IMG_NORMAL_MODE 0 //Without HDR IMG_DSP_HDR_MODE_0 1 //DSP HDR, 2 exposures IMG_DSP_HDR_MODE_1 2 //DSP HDR, 3 exposures

*Table 3-32. Parameters for Algorithm Control API **MW_IP_GET_VIDEO_HDR_ENABLE()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-33. Returns for Algorithm Control API **MW_IP_GET_VIDEO_HDR_ENABLE()**.*

Example

None

See Also:

None

3.3.16 MW_IP_SET_VIDEO_HDR_ENABLE

API Syntax:

MW_IP_SET_VIDEO_HDR_ENABLE (UINT32 channelNo, UINT8 enable)

Function Description:

- This API is used to set the current video hdr enable status.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT8	**enable**	HDR enable. IMG_NORMAL_MODE 0 //Without HDR IMG_DSP_HDR_MODE_0 1 //DSP HDR, 2 exposures IMG_DSP_HDR_MODE_1 2 //DSP HDR, 3 exposures

*Table 3-34. Parameters for Algorithm Control API **MW_IP_SET_VIDEO_HDR_ENABLE()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-35. Returns for Algorithm Control API **MW_IP_SET_VIDEO_HDR_ENABLE()**.*

Example

None

See Also:

None

3.3.17 MW_IP_GET_ADJ_AQP_INFO

API Syntax:

MW_IP_GET_ADJ_AQP_INFO (UINT32 channelNo, ADJ_AQP_INFO_s *pAdjAQPInfo, UINT8 StrNum)

Function Description:

- This API is used to get the ADJ AQP information for certain channel and stream.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
ADJ_AQP_INFO_s	*** pAdjAQPInfo**	Pointer to AQP information. Please refer to Section 3.3.17.1 for more details.
UINT8	**StrNum**	Stream number 0: Main stream 1: Second stream

*Table 3-36. Parameters for Algorithm Control API **MW_IP_GET_ADJ_AQP_INFO()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-37. Returns for Algorithm Control API **MW_IP_GET_ADJ_AQP_INFO()**.*

Example

None

See Also:

None

3.3.17.1 MW_IP_GET_ADJ_AQP_INFO > ADJ_AQP_INFO_s

Type	Field	Description
UINT8	**UPDATEFLG**	Update flag. 0: No update 1: Need to update
ADJ_LUT_s	**AQPPARAMS**	AQP parameters

*Table 3-38. Definition of **ADJ_AQP_INFO_s** for Algorithm Control API **MW_IP_GET_ADJ_AQP_INFO()**.*

3.3.18 MW_IP_SET_ADJ_AQP_INFO

API Syntax:

MW_IP_SET_ADJ_AQP_INFO (UINT32 channelNo, ADJ_AQP_INFO_s *pAdjAQPInfo, UINT8 StrNum)

Function Description:

- This API is used to set the ADJ AQP information for certain channel and stream.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
ADJ_AQP_INFO_s	*** pAdjAQPInfo**	Pointer to AQP information. Please refer to Section 3.3.17.1 for more details.
UINT8	**StrNum**	Stream number 0: Main stream 1: Second stream

*Table 3-39. Parameters for Algorithm Control API **MW_IP_SET_ADJ_AQP_INFO()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-40. Returns for Algorithm Control API **MW_IP_SET_ADJ_AQP_INFO()**.*

Example

None

See Also:

None

3.3.19 MW_IP_SET_AE_STATUS

API Syntax:

MW_IP_SET_AE_STATUS (UINT32 channelNo, UINT8 videoAeStatus, UINT8 stillAeStatus)

Function Description:

- This API is used to get the AE status for both the video mode and the still mode.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT8	**videoAeStatus**	**AMBA_LOCK**: AE lock **AMBA_PROCESSING**: AE running **AMBA_IDLE**: AE IDLE
UINT8	**stillAeStatus**	**AMBA_LOCK**: AE lock **AMBA_PROCESSING**: AE running **AMBA_IDLE**: AE IDLE

Table 3-41. Parameters for Algorithm Control API ***MW_IP_SET_AE_STATUS()***.

Returns:

Return	Description
0	Success
-1	Failure

Table 3-42. Returns for Algorithm Control API ***MW_IP_SET_AE_STATUS()***.

Example

None

See Also:

None

3.3.20 MW_IP_SET_AWB_STATUS

API Syntax:

MW_IP_SET_AWB_STATUS (UINT32 channelNo, UINT8 videoAwbStatus, UINT8 stillAwbStatus)

Function Description:

- This API is used to get the AWB status for both the video mode and the still mode.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT8	**videoAwbStatus**	**AMBA_LOCK**: AWB lock **AMBA_PROCESSING**: AWB running **AMBA_IDLE**: AWB IDLE
UINT8	**stillAwbStatus**	**AMBA_LOCK**: AWB lock **AMBA_PROCESSING**: AWB running **AMBA_IDLE**: AWB IDLE

*Table 3-43. Parameters for Algorithm Control API **MW_IP_SET_AWB_STATUS()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-44. Returns for Algorithm Control API **MW_IP_SET_AWB_STATUS()**.*

Example

None

See Also:

None

3.3.21 MW_IP_SET_AF_STATUS

API Syntax:

MW_IP_SET_AF_STATUS (UINT32 channelNo, UINT8 videoAfStatus, UINT8 stillAfStatus)

Function Description:

- This API is used to get the AF status for both the video mode and the still mode.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT8	**videoAfStatus**	**AMBA_LOCK**: AF lock **AMBA_PROCESSING**: AF running **AMBA_IDLE**: AF IDLE
UINT8	**stillAfStatus**	**AMBA_LOCK**: AF lock **AMBA_PROCESSING**: AF running **AMBA_IDLE**: AF IDLE

*Table 3-45. Parameters for Algorithm Control API **MW_IP_SET_AF_STATUS()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-46. Returns for Algorithm Control API **MW_IP_SET_AF_STATUS()**.*

Example

None

See Also:

None

3.3.22 MW_IP_GET_CURR_LV_NO

API Syntax:

MW_IP_GET_CURR_LV_NO (UINT32 channelNo, UINT16 * lvNo)

Function Description:

- This API is used to get the current LV number.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT8	***lvNo**	LV = 1, 2,3,4,5,…

*Table 3-47. Parameters for Algorithm Control API **MW_IP_GET_CURR_LV_NO()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-48. Returns for Algorithm Control API **MW_IP_GET_CURR_LV_NO()**.*

Example

None

See Also:

None

3.3.23 MW_IP_GET_CURR_LV

API Syntax:

MW_IP_GET_CURR_LV (UINT32 channelNo, UINT16 * lvNo)

Function Description:

- This API is used to get the current LV number (x100).

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT8	***lvNo**	LV = 100,200,300,400….

*Table 3-49. Parameters for Algorithm Control API **MW_IP_GET_CURR_LV()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-50. Returns for Algorithm Control API **MW_IP_GET_CURR_LV()**.*

Example

None

See Also:

None

3.3.24 MW_IP_GET_WB_LUT_NO

API Syntax:

MW_IP_GET_WB_LUT_NO (UINT32 channelNo, INT16 * lutNo)

Function Description:

- This API is used to get the current WB lookup table number.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
INT16	***lutNo**	Current WB lookup table number

*Table 3-51. Parameters for Algorithm Control API **MW_IP_GET_WB_LUT_NO()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-52. Returns for Algorithm Control API **MW_IP_GET_WB_LUT_NO()**.*

Example

None

See Also:

None

3.4 Algorithm Control: AE Algorithm

This section describes all commands to control the behavior of the AE algorithm in the Image Proc Module.

3.4.1 MW_IP_GET_MULTI_AE_CONTROL_CAPABILITY

API Syntax:

MW_IP_GET_MULTI_AE_CONTROL_CAPABILITY (UINT32 channelNo, AE_CONTROL_s *aeControlMode)

Function Description:

- This API is used for retrieving the AGC, shutter, and aperture control in AE.
- In the automatic AGC mode, AE will control AGC, shutter, and aperture to optimize the image exposure in the AE program mode.
- In the aperture-priority AE mode, AE will control AGC and shutter to optimize the image exposure.
- In the shutter-priority AE mode, AE will control AGC and aperture to optimize the image exposure.
- However, in the manual AGC mode, AE will only control shutter time and aperture to optimize the image exposure in different AE modes.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For single sensor application, the **channelNo** is 0.
AE_ CONTROL_s*	**aeControlMode**	Pointer to AE Control Capability

Table 3-53. Parameters for Algorithm Control API ***MW_IP_GET_MULTI_AE_CONTROL_CAPABILITY()***.

Returns:

Return	Description
0	Success
-1	Failure

Table 3-54. Returns for Algorithm Control API ***MW_IP_GET_MULTI_AE_CONTROL_CAPABILITY()***.

Example:

None

See Also:

MW_IP_SET_MULTI_AE_CONTROL_CAPABILITY()

3.4.2 MW_IP_SET_MULTI_AE_CONTROL_CAPABILITY

API Syntax:

MW_IP_SET_MULTI_AE_CONTROL_CAPABILITY (UINT32 channelNo, AE_CONTROL_S *aeControlMode)

Function Description:

- This API is used for setting the AGC, shutter, and aperture control in AE.
- In the automatic AGC mode, AE will control AGC, shutter, and aperture to optimize the image exposure in the AE program mode.
- In the aperture-priority AE mode, AE will control AGC and shutter to optimize the image exposure.
- In the shutter-priority AE mode, AE will control AGC and aperture to optimize the image exposure.
- However, in manual AGC mode, AE will only control shutter time and aperture to optimize the image exposure in different AE modes.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AE_ CONTROL_s*	**aeControlMode**	Pointer to AE Control Capability.

Table 3-55. Parameters for Algorithm Control API ***MW_IP_SET_MULTI_AE_CONTROL_CAPABILITY()***.

Returns:

Return	Description
0	Success
-1	Failure

Table 3-56. Returns for Algorithm Control API ***MW_IP_SET_MULTI_AE_CONTROL_CAPABILITY()***.

Example:

None

See Also:

MW_IP_GET_MULTI_AE_CONTROL_CAPABILITY()
For more details on **AE_CONTROL_s**, please refer to *AMBARELLA_SDK6_AN_IQ_Tuning*.

3.4.3 MW_IP_GET_MULTI_AE_ALGO_PARAMS

API Syntax:

MW_IP_GET_MULTI_AE_ALGO_PARAMS (UINT32 channelNo, AE_ALGO_INFO_s *aeAlgoInfo)

Function Description:

- This API is used to get the parameters for the AE algorithm.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AE_ALGO_INFO_s*	**aeAlgoInfo**	Pointer to AE Algo Information.

Table 3-57. Parameters for Algorithm Control API ***MW_IP_GET_MULTI_AE_ALGO_PARAMS()***.

Returns:

Return	Description
0	Success
- 1	Failure

Table 3-58. Returns for Algorithm Control API ***MW_IP_GET_MULTI_AE_ALGO_PARAMS()***.

Example:

None

See Also:

For more details on **AE_ALGO_INFO_s**, please refer to *AMBARELLA_SDK6_AN_IQ_Tuning*.

3.4.4 MW_IP_SET_MULTI_AE_ALGO_PARAMS

API Syntax:

MW_IP_SET_MULTI_AE_ALGO_PARAMS (UINT32 channelNo, AE_ALGO_INFO_s *aeAlgoInfo)

Function Description:

- This API is used to configure the parameters for the AE algorithm. It affects the behavior of the AE algorithm for determining the EvIndex value.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AE_ALGO_INFO_s*	**aeAlgoInfo**	Pointer to the AE Algo Information.

*Table 3-59. Parameters for Algorithm Control API **MW_IP_SET_MULTI_AE_ALGO_PARAMS()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-60. Returns for Algorithm Control API **MW_IP_SET_MULTI_AE_ALGO_PARAMS()**.*

Example:

None

See Also:

For more details on **AE_ALGO_INFO_s**, please refer to *AMBARELLA_SDK6_AN_IQ_Tuning*.

3.4.5 MW_IP_GET_MULTI_CURR_SCENE_MODE

API Syntax:

MW_IP_GET_MULTI_CURR_SCENE_MODE (MULTI_SCENE_MODE_s *pMultiSceneMode)

Function Description:

- This API is used to get the current scene mode setting for AE or AWB or ADJ of Ambarella.

Parameters:

Type	Parameter	Description
MULTI_SCENE_MODE_s	**pMultiSceneMode**	Pointer to the scene mode information. Please refer to Section 3.4.5.1 for more details.

*Table 3-61. Parameters for Algorithm Control API **MW_IP_GET_MULTI_CURR_SCENE_MODE()**.*

Returns:

Return	Description
0	Success
- 1	Failure

*Table 3-62. Returns for Algorithm Control API **MW_IP_GET_MULTI_CURR_SCENE_MODE()**.*

Example:

None

See Also:

MW_IP_SET_MULTI_CURR_SCENE_MODE()

3.4.5.1 MW_IP_GET_MULTI_CURR_SCENE_MODE-> MULTI_SCENE_MODE_s

Type	Field	Description
UINT32	**VinNum**	The parameter is used to indicate the current channel number, such as the VIN channel number. For single sensor application, the channelNo is 0.
int	**mode**	0: **IP_MODE_VIDEO** 1: **IP_MODE_STILL**
Int *	**sceneMode**	Pointer to current scene mode

*Table 3-63. Definition of **MULTI_SCENE_MODE_s** for Algorithm Control API **MW_IP_GET_MULTI_CURR_SCENE_MODE()**.*

3.4.6 MW_IP_SET_MULTI_CURR_SCENE_MODE

API Syntax:

MW_IP_SET_MULTI_CURR_SCENE_MODE (MULTI_SCENE_MODE_s *pMultiSceneMode)

Function Description:

- This API is used for setting different scene modes for AE, AWB or ADJ of Ambarella.

Parameters:

Type	Parameter	Description
MULTI_SCENE_MODE_s	**pMultiSceneMode**	Pointer to the scene mode information. Please refer to Section 3.4.5.1 for more details.

*Table 3-64. Parameters for Algorithm Control API **MW_IP_SET_MULTI_CURR_SCENE_MODE()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-65. Returns for Algorithm Control API **MW_IP_SET_MULTI_CURR_SCENE_MODE()**.*

Example:

None

See Also:

MW_IP_GET_MULTI_CURR_SCENE_MODE()

Mode	Description
254	SCENE_AUTO
0	SCENE_OFF
1	SCENE_FLASH
2	SCENE_TV_OFF
3	SCENE_AV_OFF
4	SCENE_SV_OFF
5	SCENE_TV_ONLY
6	SCENE_AV_ONLY
7	SCENE_SV_ONLY
8	SCENE_NIGHT
9	SCENE_NIGHT_PORTRAIT
10	SCENE_SPORTS
11	SCENE_LANDSCAPE

Mode	Description
12	SCENE_PORTRAIT
13	SCENE_SUNSET
14	SCENE_SAND_SNOW
15	SCENE_FLOWER
16	SCENE_FIRE_WORK
17	SCENE_WATER
18	SCENE_BACK_LIGHT
19	SCENE_BACK_LIGHT_PORTRAIT
20	SCENE_TRIPOD
21	SCENE_BLUE_SKY
22	SCENE_MACRO
23	SCENE_MACRO_TEXT
24	SCENE_ARENA
25	SCENE_D_LIGHTING
26	SCENE_MUSEUM
27	SCENE_BEACH
28	SCENE_CHILDREN
29	SCENE_PARTY
30	SCENE_FISHEYE
31	SCENE_INDOOR
32	SCENE_THROUGH_GLASS
33	SCENE_PANNING
34	SCENE_PHOTO_FRAME
35	SCENE_LOMO
36	SCENE_SELF_PORTRAIT
37	SCENE_CAR_DV
38	SCENE_LAST

Table 3-66. The Scene Modes.

3.4.7 MW_IP_GET_SCENE_MODE_INFO

API Syntax:

MW_IP_GET_SCENE_MODE_INFO (int sceneMode,SCENE_DATA_s *info)

Function Description:

- This API is used to get the scene mode information.

Parameters:

Type	Parameter	Description
int	**sceneMode**	Input scene mode
SCENE_DATA_s *	**info**	Pointer to the scene mode information.

Table 3-67. Parameters for Algorithm Control API ***MW_IP_GET_SCENE_MODE_INFO()****.*

Returns:

Return	Description
0	Success
- 1	Failure

Table 3-68. Returns for Algorithm Control API ***MW_IP_GET_SCENE_MODE_INFO()****.*

Example:

None

See Also:

MW_IP_SET_SCENE_MODE_INFO()

3.4.8 MW_IP_SET_SCENE_MODE_INFO

API Syntax:

MW_IP_SET_SCENE_MODE_INFO (int sceneMode, SCENE_DATA_s *info)

Function Description:

- This API is used to setup the scene mode information.

Parameters:

Type	Parameter	Description
int	**sceneMode**	Input scene mode
SCENE_DATA_s *	**info**	Pointer to the scene mode information.

Table 3-69. Parameters for Algorithm control API ***MW_IP_SET_SCENE_MODE_INFO()****.*

Returns:

Return	Description
0	Success
- 1	Failure

Table 3-70. Returns for Algorithm Control API ***MW_IP_SET_SCENE_MODE_INFO()****.*

Example:

None

See Also:

MW_IP_GET_SCENE_MODE_INFO()
For more details on **SCENE_DATA_s**, please refer to *AMBARELLA_SDK6_AN_IQ_Tuning*.

3.4.9 MW_IP_GET_AEB_INFO

API Syntax:

MW_IP_GET_AEB_INFO (AEB_INFO_s *aebInfo)

Function Description:

- This API is used to retrieve the current AEB setting.

Parameters:

Type	Parameter	Description
AEB_INFO_s *	**aebInfo**	Please refer to Section 3.4.9.1 for more details.

Table 3-71. Parameters for Algorithm Control API ***MW_IP_GET_AEB_INFO()***.

Returns:

Return	Description
0	Success
-1	Failure

Table 3-72. Returns for Algorithm Control API ***MW_IP_GET_AEB_INFO()***.

Example:

None

See Also:

MW_IP_SET_AEB_INFO()

3.4.9.1 MW_IP_GET_AEB_INFO > AEB_INFO_s

Type	Field	Description
UINT8	**Num**	Number of pictures taken in AEB mode is restricted to 1 - 5.
INT8	**EvBias[AEB_MAX_NUM]**	The amount of EV bias value of each picture respectively. The unit of EV is 32, so the EV bias is ranged from +3 31/32 EV ~ -3 31/32 EV. **AEB_MAX_NUM** is 9.

Table 3-73. Definition of ***AEB_INFO_s*** *for Algorithm Control API* ***MW_IP_GET_AEB_INFO()***.

3.4.10 MW_IP_SET_AEB_INFO

API Syntax:

MW_IP_SET_AEB_INFO (AEB_INFO_s * aebInfo)

Function Description:

- This API is used to configure the camera behavior in the AEB mode. Note that **MW_IP_SET_AEB_INFO** is used to configure AEB behavior only. Calling this API will not change the camera mode to the AEB mode.

Parameters:

Type	Parameter	Description
AEB_INFO_s *	**aebInfo**	Please refer to Section 3.4.9.1 for more details.

*Table 3-74. Parameters for Algorithm Control API **MW_IP_SET_AEB_INFO()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-75. Returns for Algorithm Control API **MW_IP_SET_AEB_INFO()**.*

Example:

None

See Also:

MW_IP_GET_AEB_INFO()

3.4.11 MW_IP_GET_AE_INFO

API Syntax:

MW_IP_GET_AE_INFO (UINT32 channelNo, UINT8 mode, AMBA_AE_INFO_S *pAeInfo)

Function Description:

- This API get the sensor AGC gain, sensor shutter width, and DSP digital gain.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
int	**mode**	0: **IP_MODE_VIDEO** 1: **IP_MODE_STILL**
AMBA_AE_INFO_s *	**pAeInfo**	Pointer to the AE information.

*Table 3-76. Parameters for API **MW_IP_GET_AE_INFO()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-77. Returns for API **MW_IP_GET_AE_INFO()**.*

Example:

```
AMBA_AE_INFO_s StillAeInfo[AEB_MAX_NUM];
//AEB_MAX_NUM, This symbol is defined in AmbaImg_Adjustment_A9.h

AMBA_AE_INFO_s VideoAeInfo;
UINT32 channelNo = 0;

//1. For IP_MODE_STILL,
AmbaImg_Proc_Cmd(MW_IP_GET_AE_INFO, channelNo, IP_MODE_STILL, (UINT32)Stil-
lAeInfo);

//2. For IP_MODE_VIDEO,
AmbaImg_Proc_Cmd(MW_IP_GET_AE_INFO, channelNo, IP_MODE_VIDEO,
(UINT32)&VideoAeInfo);
```

See Also:

MW_IP_SET_AE_INFO()

3.4.12 MW_IP_SET_AE_INFO

API Syntax:

MW_IP_SET_AE_INFO (UINT32 channelNo, UINT8 mode, AMBA_AE_INFO_S *pAeInfo)

Function Description:

- This API sets the sensor AGC gain, sensor shutter width, and DSP digital gain.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
int	**mode**	0: **IP_MODE_VIDEO** 1: **IP_MODE_STILL**
AMBA_AE_INFO_s *	**pAeInfo**	Pointer to the AE information.

*Table 3-78. Parameters for Algorithm Control API **MW_IP_SET_AE_INFO()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-79. Returns for Algorithm Control API **MW_IP_SET_AE_INFO()**.*

Example:

None

See Also:

MW_IP_GET_AE_INFO()

3.4.13 MW_IP_AMBA_AE_CONTROL

API Syntax:

MW_IP_AMBA_AE_CONTROL (UINT32 channelNo, AMBA_3A_STATUS_s *aaaVideoStatus, AMBA_3A_STATUS_s *aaaStillStatus)

Function Description:

- This API is the entry point for AE algorithm of Ambarella.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AMBA_3A_STATUS_s *	**aaaVideoStatus**	Please refer to Section 3.3.4.1 for more details.
AMBA_3A_STATUS_s *	**aaaStillStatus**	Please refer to Section 3.3.4.1 for more details.

*Table 3-80. Parameters for Algorithm Control API **MW_IP_AMBA_AE_CONTROL()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-81. Returns for Algorithm Control API **MW_IP_AMBA_AE_CONTROL()**.*

Example:

None

See Also:

MW_IP_GET_3A_STATUS()
MW_IP_SET_3A_STATUS()

3.4.14 MW_IP_SET_MULTI_AE_DEF_SETTING

API Syntax:

MW_IP_SET_MULTI_AE_DEF_SETTING (UINT32 channelNo, AE_DEF_SETTING_s *defSetting)

Function Description:

- This API is used to set the AE default settings.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AE_DEF_ SETTING_s	***defSetting**	Pointer to the AE default settings.

*Table 3-82. Parameters for Algorithm Control API **MW_IP_SET_MULTI_AE_DEF_SETTING()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-83. Returns for Algorithm Control API **MW_IP_SET_MULTI_AE_DEF_SETTING()**.*

Example:

None

See Also:

For more details on **AE_DEF_SETTING_s**, please refer to *AMBARELLA_SDK6_AN_IQ_Tuning.*

3.4.15 MW_IP_GET_MULTI_AE_DEF_SETTING

API Syntax:

MW_IP_GET_MULTI_AE_DEF_SETTING (UINT32 channelNo, AE_DEF_SETTING_s *defSetting)

Function Description:

- This API is used to get the AE default settings.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AE_DEF_SETTING_s	***defSetting**	Pointer to the AE default settings.

*Table 3-84. Parameters for Algorithm Control API **MW_IP_GET_MULTI_AE_DEF_SETTING()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-85. Returns for Algorithm Control API **MW_IP_GET_MULTI_AE_DEF_SETTING()**.*

Example:

None

See Also:

MW_IP_SET_MULTI_AE_DEF_SETTING()

3.4.16 MW_IP_SET_MULTI_AE_EV_LUT

API Syntax:

MW_IP_SET_MULTI_AE_EV_LUT (UINT32 channelNo, AE_EV_LUT_s *evLut)

Function Description:

- This API is used to setup the AE Ev lookup table.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AE_EV_LUT_s	***evLut**	Pointer to the AE Ev lookup table.

*Table 3-86. Parameters for Algorithm Control API **MW_IP_SET_MULTI_AE_EV_LUT()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-87. Returns for Algorithm Control API **MW_IP_SET_MULTI_AE_EV_LUT()**.*

Example:

None

See Also:

For more details on **AE_EV_LUT_s**, please refer to *AMBARELLA_SDK6_AN_IQ_Tuning*.

3.4.17 MW_IP_GET_MULTI_AE_EV_LUT

API Syntax:

MW_IP_GET_MULTI_AE_EV_LUT (UINT32 channelNo, AE_EV_LUT_s *evLut)

Function Description:

- This API is used to setup the AE Ev lookup table.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AE_EV_LUT_s	***evLut**	Pointer to the AE Ev lookup table.

Table 3-88. Parameters for Algorithm Control API ***MW_IP_GET_MULTI_AE_EV_LUT()****.*

Returns:

Return	Description
0	Success
-1	Failure

Table 3-89. Returns for Algorithm Control API ***MW_IP_GET_MULTI_AE_EV_LUT()****.*

Example:

None

See Also:

MW_IP_SET_MULTI_AE_EV_LUT()

3.4.18 MW_IP_SET_DGAIN

API Syntax:

MW_IP_SET_DGAIN (UINT32 channelNo, UINT32 dgain)

Function Description:

- This API is used to setup the digital gain for certain channels.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT32	**dgain**	Input digital gain. Uint is 4096.

*Table 3-90. Parameters for Algorithm Control API **MW_IP_SET_DGAIN()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-91. Returns for Algorithm Control API **MW_IP_SET_DGAIN()**.*

Example:

None

See Also:

None

3.4.19 MW_IP_GET_DGAIN

API Syntax:

MW_IP_GET_DGAIN (UINT32 channelNo, UINT32 *dgain)

Function Description:

- This API is used to get the digital gain for certain channel.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT32	***dgain**	Pointer to the input digital gain. Uint is 4096.

*Table 3-92. Parameters for Algorithm Control API **MW_IP_GET_DGAIN()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-93. Returns for Algorithm Control API **MW_IP_GET_DGAIN()**.*

Example:

None

See Also:

None

3.4.20 MW_IP_SET_GLOBAL_DGAIN

API Syntax:

MW_IP_SET_GLOBAL_DGAIN (UINT32 channelNo, UINT32 gDgain)

Function Description:

- This API is used to setup the global digital gain for certain channels.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT32	**gDgain**	Input global digital gain. Uint is 4096.

*Table 3-94. Parameters for Algorithm Control API **MW_IP_SET_GLOBAL_DGAIN()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-95. Returns for Algorithm Control API **MW_IP_SET_GLOBAL_DGAIN()**.*

Example:

None

See Also:

None

3.4.21 MW_IP_GET_GLOBAL_DGAIN

API Syntax:

MW_IP_GET_GLOBAL_DGAIN (UINT32 channelNo, UINT32 *gDgain)

Function Description:

- This API is used to get the global digital gain for certain channels.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT32	***gDgain**	Pointer to the global digital gain. Uint is 4096.

Table 3-96. Parameters for Algorithm Control API ***MW_IP_GET_GLOBAL_DGAIN()***.

Returns:

Return	Description
0	Success
-1	Failure

Table 3-97. Returns for Algorithm Control API ***MW_IP_GET_GLOBAL_DGAIN()***.

Example:

None

See Also:

None

3.4.22 MW_IP_EXPS_TO_EV_IDX

API Syntax:

MW_IP_EXPS_TO_EV_IDX (UINT32 channelNo, UINT8 type, AMBA_AE_INFO_s *aeInfo)

Function Description:

- This API is used to get the EvIndex of certain Ae information.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT8	**type**	Type: IP_MODE_VIDEO / IP_MODE_STILL
AMBA_AE_INFO_s	***aeInfo**	Pointer to Ae information

*Table 3-98. Parameters for Algorithm Control API **MW_IP_EXPS_TO_EV_IDX()**.*

Returns:

Return	Description
int EvIndex;	Return the ev_index of input ae settings.

*Table 3-99. Returns for Algorithm Control API **MW_IP_EXPS_TO_EV_IDX()**.*

Example:

None

See Also:

None

3.4.23 MW_IP_EXPS_TO_NF_IDX

API Syntax:

MW_IP_EXPS_TO_NF_IDX (UINT32 channelNo, UINT8 type, AMBA_AE_INFO_s *aeInfo)

Function Description:

- This API is used to get the NfIndex of certain Ae information.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT8	**type**	Type: IP_MODE_VIDEO / IP_MODE_STILL
AMBA_AE_INFO_s	***aeInfo**	Pointer to Ae information

*Table 3-100. Parameters for Algorithm Control API **MW_IP_EXPS_TO_NF_IDX()**.*

Returns:

Return	Description
int NfIndex;	Return the nf_index of input ae settings.

*Table 3-101. Returns for Algorithm Control API **MW_IP_EXPS_TO_NF_IDX()**.*

Example:

None

See Also:

None

3.4.24 MW_IP_GET_ENVIRONMENT_INFO

API Syntax:

MW_IP_GET_ENVIRONMENT_INFO (UINT32 channelNo, UINT32 *value)

Function Description:

- This API is used to get the current Ae index of certain Ae information.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT32	***value**	Pointer to Ae index

*Table 3-102. Parameters for Algorithm Control API **MW_IP_GET_ENVIRONMENT_INFO()**.*

Returns:

Return	Description
0	Success
- 1	Failure

*Table 3-103. Returns for Algorithm Control API **MW_IP_GET_ENVIRONMENT_INFO()**.*

Example:

None

See Also:

None

3.4.25 MW_IP_SET_ENVIRONMENT_INFO

API Syntax:

MW_IP_SET_ENVIRONMENT_INFO (UINT32 channelNo, UINT32 value)

Function Description:

- This API is used to set the Ae index of certain channel.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT32	**value**	Ae index

*Table 3-104. Parameters for Algorithm Control API **MW_IP_SET_ENVIRONMENT_INFO()**.*

Returns:

Return	Description
0	Success
- 1	Failure

*Table 3-105. Returns for Algorithm Control API **MW_IP_SET_ENVIRONMENT_INFO()**.*

Example:

None

See Also:

None

3.4.26 MW_IP_GET_FLICKER_CMD

API Syntax:

MW_IP_GET_FLICKER_CMD (UINT8 *enable)

Function Description:

- This API is used to get the flicker command.

Parameters:

Type	Parameter	Description
UINT8	***enable**	UINT8 *enable: Pointer to flicker command, Enable(1), Disable(0)

*Table 3-106. Parameters for Algorithm Control API **MW_IP_GET_FLICKER_CMD()**.*

Returns:

Return	Description
0	Success
- 1	Failure

*Table 3-107. Returns for Algorithm Control API **MW_IP_GET_FLICKER_CMD()**.*

Example:

None

See Also:

None

3.4.27 MW_IP_SET_FLICKER_CMD

API Syntax:

MW_IP_SET_FLICKER_CMD (UINT8 enable)

Function Description:

- This API is used to set the flicker command.

Parameters:

Type	Parameter	Description
UINT8	**enable**	UINT8 enable: Enable(1), Disable(0)

*Table 3-108. Parameters for Algorithm Control API **MW_IP_SET_FLICKER_CMD()**.*

Returns:

Return	Description
0	Success
- 1	Failure

*Table 3-109. Returns for Algorithm Control API **MW_IP_SET_FLICKER_CMD()**.*

Example:

None

See Also:

None

3.5 Algorithm Control: AWB Algorithm

This section describes all commands to control the behavior of the AWB algorithm in the Image Proc Module.

3.5.1 MW_IP_GET_MULTI_AWB_CONTROL_CAPABILITY

API Syntax:

MW_IP_GET_MULTI_AWB_CONTROL_CAPABILITY (UINT32 channelNo, AWB_CONTROL_s *awbControl-Mode)

Function Description:

- This API is used for retrieving the special light source for the AWB region.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel num-ber, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AWB_ CONTROL_s*	**awbControlMode**	Pointer to the AWB Control Capability.

*Table 3-110. Parameters for Algorithm Control API **MW_IP_GET_MULTI_AWB_CONTROL_CAPABILITY()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-111. Returns for Algorithm Control API API **MW_IP_GET_MULTI_AWB_CONTROL_CAPABILITY()**.*

Example:

None

See Also:

MW_IP_SET_MULTI_AWB_CONTROL_CAPABILITY()
For more details on **AWB_CONTROL_s**, please refer to *AMBARELLA_SDK6_AN_IQ_Tuning*.

3.5.2 MW_IP_SET_MULTI_AWB_CONTROL_CAPABILITY

API Syntax:

MW_IP_SET_MULTI_AWB_CONTROL_CAPABILITY (UINT32 channelNo, AWB_CONTROL_s *awbControlMode)

Function Description:

- This API is used for setting the special light source for the AWB region. Two methods of white balance algorithms are supported. One is the gray-world like algorithm and the other is white-patch method. These two methods estimate the optimum color balance gains from the AWB statistics. Color balance gains are interactively updated based on a speed control parameter.
- NewGain = (OldGain * (64 - speed) + EstimatedGain * speed) / 64.
- The higher the speed is, the more frequent are the updates to the color gains. For speed 0, the color gains are not updated.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AWB_CONTROL_s*	**awbControlMode**	Pointer to the AWB Control Capability.

*Table 3-112. Parameters for Algorithm Control API **MW_IP_SET_MULTI_AWB_CONTROL_CAPABILITY()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-113. Returns for Algorithm Control API **MW_IP_SET_MULTI_AWB_CONTROL_CAPABILITY()**.*

Example:

None

See Also:

MW_IP_GET_MULTI_AWB_CONTROL_CAPABILITY()
For more details on **AWB_CONTROL_s**, please refer to *AMBARELLA_SDK6_AN_IQ_Tuning*.

3.5.3 MW_IP_GET_MULTI_AWB_ALGO_PARAMS

API Syntax:

MW_IP_GET_MULTI_AWB_ALGO_PARAMS (UINT32 channelNo, AWB_ALGO_INFO_s *awbAlgoInfo)

Function Description:

- This API is used for retrieving the AWB algorithm parameters.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AWB_ALGO_INFO_s *	**awbAlgoInfo**	Pointer to the AWB algorithm parameters.

*Table 3-114. Parameters for Algorithm Control API **MW_IP_GET_MULTI_AWB_ALGO_PARAMS()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-115. Returns for Algorithm Control API **MW_IP_GET_MULTI_AWB_ALGO_PARAMS()**.*

Example:

None

See Also:

MW_IP_SET_MULTI_AWB_ALGO_PARAMS()

3.5.4 MW_IP_SET_MULTI_AWB_ALGO_PARAMS

API Syntax:

MW_IP_SET_MULTI_AWB_ALGO_PARAMS (UINT32 channelNo, AWB_ALGO_INFO_s *awbAlgoInfo)

Function Description:

- This API is used to configure the parameters for the AWB algorithm.
- Each candidate white region is bounded by the thick blue lines as shown in Figure 3-1 below. Those data whose (G/R, G/B) is located in the candidate region are recognized as possible white samples. The candidate region is specified by the following conditions.
    - (1) (G/R)min <= (G/R) <= (G/R)max
    - (2) (G/B)min <= (G/B) <= (G/B)max
    - (3) Yamin <= (G/B) <= Yamax, where Yamin = Y_a_min - Y_a_min_slope * (G/R) , Yamax = Y_a_max - Y_a_max_slope * (G/R) as the green lines in Figure 3-1.
    - (4) Ybmin <= (G/B) <= Ybmax, where Ybmin = Y_b_min + Y_b_min_slope * (G/R) , Ybmax = Y_b_max + Y_b_max_slope * (G/R) as the blue lines in Figure 3-1.

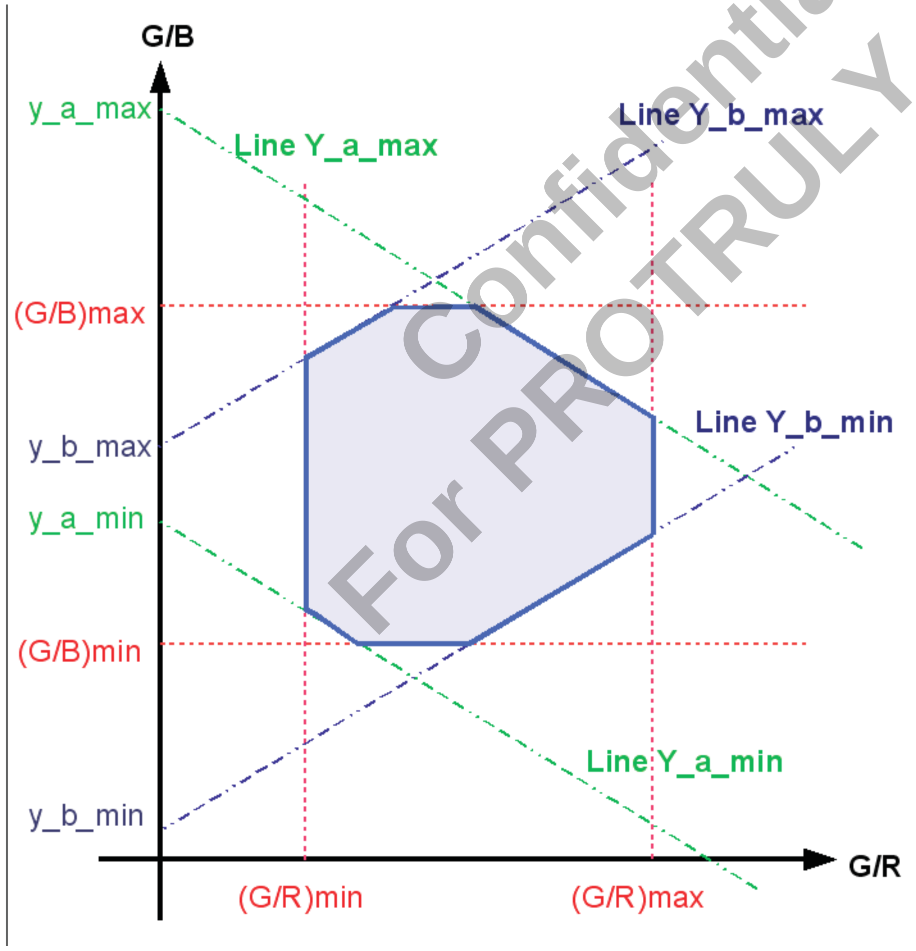

Figure 3-1. Candidate White Region.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AWB_ALGO_INFO_s *	**awbAlgoInfo**	Pointer to the AWB algo params.

Table 3-116. Parameters for Algorithm Control API ***MW_IP_SET_MULTI_AWB_ALGO_PARAMS ()***.

Returns:

Return	Description
0	Success
-1	Failure

Table 3-117. Returns for Algorithm Control API ***MW_IP_SET_MULTI_AWB_ALGO_PARAMS()***.

Example:

None

See Also:

MW_IP_GET_MULTI_AWB_ALGO_PARAMS()
For more details on **AWB_ALGO_INFO_s**, please refer to *AMBARELLA_SDK6_AN_IQ_Tuning*.

3.5.5 MW_IP_GET_PIPE_WB_GAIN

API Syntax:

MW_IP_GET_PIPE_WB_GAIN (UINT32 channelNo, UINT8 mode, AMBA_DSP_IMG_WB_GAIN_s *awb-Gain)

Function Description:

- This API is used to get the current WB gain.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT8	**mode**	**IP_MODE_VIDEO**: Image Video Mode **IP_MODE_STILL**: Image Still Mode
AMBA_DSP_IMG_WB_GAIN_s *	**awbGain**	Input WbGain.

Table 3-118. Parameters for Algorithm Control API ***MW_IP_GET_PIPE_WB_GAIN()***.

Returns:

Return	Description
0	Success
- 1	Failure

Table 3-119. Returns for Algorithm Control API ***MW_IP_GET_PIPE_WB_GAIN()***.

Example:

None

See Also:

MW_IP_SET_PIPE_WB_GAIN()

3.5.6 MW_IP_SET_PIPE_WB_GAIN

API Syntax:

MW_IP_SET_PIPE_WB_GAIN (UINT32 channelNo, UINT8 mode, AMBA_DSP_IMG_WB_GAIN_s *awb-Gain)

Function Description:

- This API is used to set the WB gain.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT8	**mode**	**IP_MODE_VIDEO**: Image Video Mode **IP_MODE_STILL**: Image Still Mode
AMBA_DSP_IMG_WB_GAIN_s t *	**awbGain**	Input WbGain.

Table 3-120. Parameters for Algorithm Control API ***MW_IP_SET_PIPE_WB_GAIN()***.

Returns:

Return	Description
0	Success
- 1	Failure

Table 3-121. Returns for Algorithm Control API ***MW_IP_SET_PIPE_WB_GAIN()***.

Example:

None

See Also:

MW_IP_GET_PIPE_WB_GAIN()

3.5.7 MW_IP_AMBA_AWB_CONTROL

API Syntax:

MW_IP_AMBA_AWB_CONTROL (UINT32 channelNo, AMBA_3A_STATUS_s *aaaVideoStatus, AMBA_3A_STATUS_s *aaaStillStatus)

Function Description:

- This API is the entry point of AWB algorithm of Ambarella.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AMBA_3A_STATUS_s *	**aaaVideoStatus**	Please refer to Section 3.3.4.1 for more details.
AMBA_3A_STATUS_s *	**aaaStillStatus**	Please refer to Section 3.3.4.1 for more details.

*Table 3-122. Parameters for Algorithm Control API **MW_IP_AMBA_AWB_CONTROL()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 3-123. Returns for Algorithm Control API **MW_IP_AMBA_AWB_CONTROL()**.*

Example:

None

See Also:

MW_IP_GET_3A_STATUS()
MW_IP_SET_3A_STATUS()

4 Image Quality Control

4.1 Image Quality Control: Overview

This chapter provides the functions to set and query Image Properties, Digital Effects, and Color Styles.

4.2 Image Quality Control: List of APIs

4.3 Image Quality Control: Image Property

This section describes all commands to control the Image Property.

4.3.1 MW_IP_RESET_VIDEO_PIPE_CTRL_PARAMS

API Syntax:

MW_IP_RESET_VIDEO_PIPE_CTRL_PARAMS (UINT32 channelNo, UINT8 type)

Function Description:

- This API is used to reset the video pipeline parameters.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel num-ber, such as the VIN channel number. For single sensor application, the **channelNo** is 0.
UINT8	**type**	0: Normal filters 1: HDR filters

*Table 4-1. Parameters for Image Quality Control API **MW_IP_RESET_VIDEO_PIPE_CTRL_PARAMS()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 4-2. Returns for Image Quality Control API **MW_IP_RESET_VIDEO_PIPE_CTRL_PARAM()**.*

Example:

None

See Also:

None

4.3.2 MW_IP_RESET_STILL_PIPE_CTRL_PARAMS

API Syntax:

MW_IP_RESET_STILL_PIPE_CTRL_PARAMS (UINT32 channelNo)

Function Description:

- This API is used to reset the still pipeline parameters.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For single sensor application, the **channelNo** is 0.

*Table 4-3. Parameters for Image Quality Control API **MW_IP_RESET_STILL_PIPE_CTRL_PARAMS()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 4-4. Returns for Image Quality Control API **MW_IP_RESET_STILL_PIPE_CTRL_PARAMS()**.*

Example:

None

See Also:

None

4.3.3 MW_IP_SET_VIDEO_PIPE_CTRL_PARAMS

API Syntax:

MW_IP_SET_VIDEO_PIPE_CTRL_PARAMS (UINT32 channelNo)

Function Description:

- This API is used to setup video pipeline parameters calculated by the Ambarella Adj algorithm.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.

*Table 4-5. Parameters for Image Quality Control API **MW_IP_SET_VIDEO_PIPE_CTRL_PARAMS()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 4-6. Returns for Image Quality Control API **MW_IP_SET_VIDEO_PIPE_CTRL_PARAMS()**.*

Example:

None

See Also:

None

4.3.4 MW_IP_SET_STILL_PIPE_CTRL_PARAMS

API Syntax:

MW_IP_SET_STILL_PIPE_CTRL_PARAMS (UINT32 channelNo, AMBA_DSP_IMG_MODE_CFG_s *mode)

Function Description:

- This API is used to setup still pipeline parameters calculated by the Ambarella Adj algorithm.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AMBA_DSP_IMG_MODE_CFG_s	***mode**	Pointer to the DSP image mode configuration.

Table 4-7. Parameters for Image Quality Control API ***MW_IP_SET_STILL_PIPE_CTRL_PARAMS()****.*

Returns:

Return	Description
0	Success
-1	Failure

Table 4-8. Returns for Image Quality Control API ***MW_IP_SET_STILL_PIPE_CTRL_PARAMS()****.*

Example:

None

See Also:

For detailed definition of **AMBA_DSP_IMG_MODE_CFG_s**, please refer to *AMBARELLA_SDK6_API_Image_Kernel*.

4.3.5 MW_IP_SET_ IMAGE_BRIGHTNESS

API Syntax:

MW_IP_SET_ IMAGE_BRIGHTNESS (UINT8 channelNo, AMBA_DSP_IMG_MODE_CFG_s *mode Cfg, INT16 Brightness)

Function Description:

- This API is used to adjust brightness of the image.

Parameters:

Type	Parameter	Description
UINT8	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AMBA_DSP_ IMG_MODE_ CFG_s	***modeCfg**	Pointer to the DSP image mode configuration.
INT16	**Brightness**	Brightness adjustment ranging from - 256 to 256. 0: Default value without adjustment.

*Table 4-9. Parameters for Image Quality Control API **MW_IP_SET_ IMAGE_BRIGHTNESS()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 4-10. Returns for Image Quality Control API **MW_IP_SET_ IMAGE_BRIGHTNESS()**.*

Example:

```
AMBA_DSP_IMG_MODE_CFG_s ModeCfg;
INT16 value;
value = 100;  //Adjust. Brightness +100
memset(&ModeCfg, 0, sizeof(ModeCfg));
AmbaImg_Proc_Cmd(MW_IP_SET_IMAGE_BRIGHTNESS, 0, (UINT32)&ModeCfg, (UINT32)
value);
```

See Also:

For more details on **AMBA_DSP_IMG_MODE_CFG_s**, please refer to *AMBARELLA_SDK6_API_Image_ Kernel*.

4.3.6 MW_IP_SET_IMAGE_CONTRAST

API Syntax:

MW_IP_SET_ IMAGE_CONTRAST (UINT8 channelNo, AMBA_DSP_IMG_MODE_CFG_s *mode Cfg, UINT16 Contrast)

Function Description:

- This API is used to adjust the contrast of the image.

Parameters:

Type	Parameter	Description
UINT8	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AMBA_DSP_IMG_MODE_CFG_s	***modeCfg**	Pointer to the DSP image mode configuration.
UINT16	**Contrast**	Contrast adjustment ranging from 0 to 256. 64: Default value without change.

*Table 4-11. Parameters for Image Quality Control API **MW_IP_SET_ IMAGE_CONTRAST()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 4-12. Returns for Image Quality Control API **MW_IP_SET_ IMAGE_CONTRAST()**.*

Example:

```
AMBA_DSP_IMG_MODE_CFG_s ModeCfg;
UINT16 value;
value = 64;  //No change
memset(&ModeCfg, 0, sizeof(ModeCfg));
AmbaImg_Proc_Cmd(MW_IP_SET_IMAGE_CONTRAST, 0, (UINT32)&ModeCfg, (UINT32)
value);
```

See Also:

For more details on **AMBA_DSP_IMG_MODE_CFG_s**, please refer to *AMBARELLA_SDK6_API_Image_Kernel*.

4.3.7 MW_IP_SET_IMAGE_SATURATION

API Syntax:

MW_IP_SET_ IMAGE_SATURATION (UINT8 channelNo, AMBA_DSP_IMG_MODE_CFG_s *mode Cfg, UINT16 Saturation)

Function Description:

- This API is used to adjust the saturation of the image.

Parameters:

Type	Parameter	Description
UINT8	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AMBA_DSP_IMG_MODE_CFG_s	***modeCfg**	Pointer to the DSP image mode configuration.
UINT16	**Saturation**	Saturation adjustment ranging from 0 to 256. 64: Default value without change.

*Table 4-13. Parameters for Image Quality Control API **MW_IP_SET_IMAGE_SATURATION()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 4-14. Returns for Image Quality Control API **MW_IP_SET_IMAGE_SATURATION()**.*

Example:

```
AMBA_DSP_IMG_MODE_CFG_s ModeCfg;
UINT16 value;
value = 64;  //No change
memset(&ModeCfg, 0, sizeof(ModeCfg));
AmbaImg_Proc_Cmd(MW_IP_SET_IMAGE_SATURATION, 0, (UINT32)&ModeCfg, (UINT32)
value);
```

See Also:

For more details of **AMBA_DSP_IMG_MODE_CFG_s**, please refer to *AMBARELLA_SDK6_API_Image_Kernel*.

4.3.8 MW_IP_SET_IMAGE_HUE

API Syntax:

MW_IP_SET_IMAGE_HUE (UINT8 channelNo, AMBA_DSP_IMG_MODE_CFG_s *mode Cfg, INT16 Hue)

Function Description:

- This API is used to adjust the hue of the image.

Parameters:

Type	Parameter	Description
UINT8	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AMBA_DSP_IMG_MODE_CFG_s	***modeCfg**	Pointer to the DSP image mode configuration.
INT16	**Hue**	Hue adjustment ranging from -15 to 15. 0: Default value without change.

*Table 4-15. Parameters for Image Quality Control API **MW_IP_SET_IMAGE_HUE()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 4-16. Returns for Image Quality Control API **MW_IP_SET_IMAGE_HUE()**.*

Example:

```
AMBA_DSP_IMG_MODE_CFG_s ModeCfg;
INT16 value;
value = 0;  //No change
memset(&ModeCfg, 0, sizeof(ModeCfg));
AmbaImg_Proc_Cmd(MW_IP_SET_IMAGE_HUE, 0, (UINT32)&ModeCfg, (UINT32)value);
```

See Also:

For more details on **AMBA_DSP_IMG_MODE_CFG_s**, please refer to *AMBARELLA_SDK6_API_Image_Kernel*.

4.3.9 MW_IP_SET_IMAGE_SHARPNESS

API Syntax:

MW_IP_SET_IMAGE_SHARPNESS (UINT8 channelNo, AMBA_DSP_IMG_MODE_CFG_s *mode Cfg, UINT16 Sharpness)

Function Description:

- This API is used to adjust the sharpness of the image.

Parameters:

Type	Parameter	Description
UINT8	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AMBA_DSP_IMG_MODE_CFG_s	***modeCfg**	Pointer to the DSP image mode configuration.
UINT16	**Sharpness**	Sharpness adjustment ranging from 0 to 6. 3: Default value without change.

Table 4-17. Parameters for Image Quality Control API ***MW_IP_SET_IMAGE_SHARPNESS()****.*

Returns:

Return	Description
0	Success
-1	Failure

Table 4-18. Returns for Image Quality Control API ***MW_IP_SET_IMAGE_SHARPNESS()****.*

Example:

```
AMBA_DSP_IMG_MODE_CFG_s  ModeCfg;
UINT16 value;
value = 3;  //No change
memset(&ModeCfg,  0, sizeof(ModeCfg));
AmbaImg_Proc_Cmd(MW_IP_SET_IMAGE_SHARPNESS,  0, (UINT32)&ModeCfg,  (UINT32)
value);
```

See Also:

For more details on **AMBA_DSP_IMG_MODE_CFG_s**, please refer to *AMBARELLA_SDK6_API_Image_Kernel*.

4.3.10 MW_IP_SET_DIGITAL_EFFECT

API Syntax:

MW_IP_SET_DIGITAL_EFFECT (UINT8 channelNo, UINT8 effect)

Function Description:

- This API is used to set the digital effect to apply to the image.

Parameters:

Type	Parameter	Description
UINT8	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT8	**effect**	0: **DIGITAL_NO_EFFECT** 1: **DIGITAL_ART** 2: **DIGITAL_SEPIA** 3: **DIGITAL_NEGATIVE** 4: **DIGITAL_BW** 5: **DIGITAL_VIVID** 6: **DIGITAL_70FILM** 18: **DIGITAL_CUSTOMER_0** 19: **DIGITAL_CUSTOMER_1** 20: **DIGITAL_CUSTOMER_2** 21: **DIGITAL_CUSTOMER_3** 22: **DIGITAL_CUSTOMER_4** 23: **DIGITAL_CUSTOMER_5** 24: **DIGITAL_LAST**

*Table 4-19. Parameters for Image Quality Control API **MW_IP_SET_DIGITAL_EFFECT()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 4-20. Returns for Image Quality Control API **MW_IP_SET_DIGITAL_EFFECT()**.*

Example:

```
UINT8 DeTmp = 0;
UINT32 ChNo = 0;

DeTmp = DIGITAL_NO_EFFECT; //DIGITAL_ART/DIGITAL_SEPIA/....
AmbaIQParam_DigitalEffect_Load_Color_Table((int)DeTmp, 0, 1); // If user
switch digital effect then it need to load cc table
AmbaImg_Proc_Cmd(MW_IP_SET_DIGITAL_EFFECT, ChNo, (UINT32)DeTmp, 0);
```

See Also:

None

4.3.11 MW_IP_GET_DIGITAL_EFFECT

API Syntax:

MW_IP_GET_DIGITAL_EFFECT (UINT8 channelNo, UINT8 * effect)

Function Description:

- This API is used to get the digital effect to apply to the image.

Parameters:

Type	Parameter	Description
UINT8	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT8	*** effect**	0: **DIGITAL_NO_EFFECT** 1: **DIGITAL_ART** 2: **DIGITAL_SEPIA** 3: **DIGITAL_NEGATIVE** 4: **DIGITAL_BW** 5: **DIGITAL_VIVID** 6: **DIGITAL_70FILM** 18: **DIGITAL_CUSTOMER_0** 19: **DIGITAL_CUSTOMER_1** 20: **DIGITAL_CUSTOMER_2** 21: **DIGITAL_CUSTOMER_3** 22: **DIGITAL_CUSTOMER_4** 23: **DIGITAL_CUSTOMER_5** 24: **DIGITAL_LAST**

*Table 4-21. Parameters for Image Quality Control API **MW_IP_GET_DIGITAL_EFFECT()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 4-22. Returns for Image Quality Control API **MW_IP_GET_DIGITAL_EFFECT()**.*

Example:

```
UINT8 DeTmp = 0;
UINT32 ChNo = 0;

AmbaImg_Proc_Cmd(MW_IP_GET_DIGITAL_EFFECT, ChNo, (UINT32)&DeTmp, 0);
```

See Also:

None

4.3.12 MW_IP_SET_DE_PARAM

API Syntax:

MW_IP_SET_DE_PARAM (UINT8 mode, DE_PARAM_s * pDeParam)

Function Description:

- This API is used to setup certain digital effect parameters for the Video/Still mode.

Parameters:

Type	Parameter	Description
UINT8	**mode**	0: **IP_MODE_VIDEO** 1: **IP_MODE_STILL**
DE_PARAM_s	***pDeParam**	Pointer to the digital effect parameters.

*Table 4-23. Parameters for Image Quality Control API **MW_IP_SET_DE_PARAM()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 4-24. Returns for Image Quality Control API **MW_IP_SET_DE_PARAM()**.*

Example:

```
DE_PARAM_s DeParam;

UINT8 Mode = IP_MODE_VIDEO;// IP_MODE_STILL

AmbaImg_Proc_Cmd(MW_IP_GET_DE_PARAM, IP_MODE_VIDEO, (UINT32)& DeParam, 0);

AmbaImg_Proc_Cmd(MW_IP_SET_DE_PARAM, IP_MODE_VIDEO, (UINT32)& DeParam, 0);
```

See Also:

For detailed definition of **DE_PARAM_s**, please refer to *AMBARELLA_SDK6_AN_IQ_Tuning*.

4.3.13 MW_IP_GET_DE_PARAM

API Syntax:

MW_IP_GET_DE_PARAM (UINT8 mode, DE_PARAM_s *pDeParam)

Function Description:

- This API is used to get the certain digital effect parameters for the Video/Still mode.

Parameters:

Type	Parameter	Description
UINT8	**mode**	0: **IP_MODE_VIDEO** 1: **IP_MODE_STILL**
DE_PARAM_s	***pDeParam**	Pointer to the digital effect parameters.

*Table 4-25. Parameters for Image Quality Control API **MW_IP_GET_DE_PARAM()**.*

Returns:

Return	Description
0	Success
-1	Failure

*Table 4-26. Returns for Image Quality Control API **MW_IP_GET_DE_PARAM()**.*

Example:

```
DE_PARAM_s DeParam;

UINT8 Mode = IP_MODE_VIDEO;// IP_MODE_STILL

AmbaImg_Proc_Cmd(MW_IP_GET_DE_PARAM, IP_MODE_VIDEO, (UINT32)& DeParam, 0);

AmbaImg_Proc_Cmd(MW_IP_SET_DE_PARAM, IP_MODE_VIDEO, (UINT32)& DeParam, 0);
```

See Also:

For the detailed definition of **DE_PARAM_s**, please refer to *AMBARELLA_SDK6_AN_IQ_Tuning*.

5 AAA Statistics

5.1 AAA Statistics: Overview

This chapter provides the functions to get the AAA Statistics from the hardware and use them for calculations (e.g., average for a tile). In addition to directly getting hardware statistics data, the Image Proc Module also provides utility commands to get the average values for the AE/AWB/AF tiles.

5.2 AAA Statistics: List of APIs

5.2.1 MW_IP_SET_CFA_3A_STAT

API Syntax:

MW_IP_SET_CFA_3A_STAT (UINT32 channelNo, AMBA_DSP_EVENT_CFA_3A_DATA_s *pCfaStat)

Function Description:

- This API is used to set the CFA domain AE/AWB/AF(AAA) statistics to the Imgproc module.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AMBA_DSP_EVENT_CFA_3A_DATA_s *	**pCfaSta**	Pointer to the CFA stat data.

Table 5-1. Parameters for AAA Statistics API **MW_IP_SET_CFA_3A_STAT()**.

Returns:

Return	Description
0	Success
1	Failure

Table 5-2. Returns for AAA Statistics API ***MW_IP_SET_CFA_3A_STAT()***.

Example:

None

See Also:

For the detailed definition of **AMBA_DSP_EVENT_CFA_3A_DATA_s**, please refer to *AMBARELLA_SDK6_API_Image_Kernel*.

5.2.2 MW_IP_GET_CFA_3A_STAT

API Syntax:

MW_IP_GET_CFA_3A_STAT (UINT32 channelNo, AMBA_DSP_EVENT_CFA_3A_DATA_s *pCfaStat)

Function Description:

- This API is used to get the CFA domain AE/AWB/AF(AAA) statistics from the Imgproc module.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AMBA_DSP_EVENT_CFA_3A_DATA_s *	**pCfaStat**	Pointer to the CFA stat data.

Table 5-3. Parameters for AAA Statistics API **MW_IP_GET_CFA_3A_STAT()**.

Returns:

Return	Description
0	Success
- 1	Failure

Table 5-4. Returns for AAA Statistics API ***MW_IP_GET_CFA_3A_STAT()***.

Example:

None

See Also:

None

5.2.3 MW_IP_SET_RGB_3A_STAT

API Syntax:

MW_IP_SET_RGB_3A_STAT (UINT32 channelNo, AMBA_DSP_EVENT_RGB_3A_DATA_s *pRgbStat)

Function Description:

- This API is used to set the RGB domain AE/AWB (AAA) statistics to the Imgproc module.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AMBA_DSP_EVENT_RGB_3A_DATA_s *	**pRgbStat**	Pointer to the RGB stat data.

*Table 5-5. Parameters for AAA Statistics API **MW_IP_SET_RGB_3A_STAT()**.*

Returns:

Return	Description
0	Success
- 1	Failure

*Table 5-6. Returns for AAA Statistics API **MW_IP_SET_RGB_3A_STAT()**.*

Example:

None

See Also:

For detailed definition of **AMBA_DSP_EVENT_RGB_3A_DATA_s**, please refer to *AMBARELLA_SDK6_API_Image_Kernel*.

5.2.4 MW_IP_GET_RGB_3A_STAT

API Syntax:

MW_IP_GET_RGB_3A_STAT (UINT32 channelNo, AMBA_DSP_EVENT_RGB_3A_DATA_s *pRgbStat)

Function Description:

- This API is used to get the RGB domain AE/AWB (AAA) statistics from the Imgproc module.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
AMBA_DSP_ EVENT_ RGB_3A_ DATA_s *	**pRgbStat**	Pointer to the RGB stat data.

*Table 5-7. Parameters for AAA Statistics API **MW_IP_GET_RGB_3A_STAT()**.*

Returns:

Return	Description
0	Success
- 1	Failure

*Table 5-8. Returns for AAA Statistics API **MW_IP_GET_RGB_3A_STAT()**.*

Example:

None

See Also:

None

5.2.5 MW_IP_GET_AE_TILE_INFO

API Syntax:

MW_IP_GET_AE_TILE_INFO (UINT32 channelNo, UINT8 mode, AMBA_AE_TILES_INFO_s *aeTilesInfo)

Function Description:

- This API is used to get the AE statistics tile information.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT8	**mode**	Specify the AE stat. mode. 0: RGB mode 1: CFA mode
AMBA_AE_TILES_INFO_s *	**aeTilesInfo**	Pointer to the **AMBA_AE_TILES_INFO_s**. Please refer to Section 5.2.5.1 for more details.

*Table 5-9. Parameters for AAA Statistics API **MW_IP_GET_AE_TILE_INFO()**.*

Returns:

Return	Description
0	Success
- 1	Failure

*Table 5-10. Returns for AAA Statistics API **MW_IP_GET_AE_TILE_INFO()**.*

Example:

None

See Also:

None

5.2.5.1 MW_IP_GET_AE_TILE_INFO > AMBA_AE_TILES_INFO_s

Type	Field	Description
UINT16	**Rows**	Number of tile rows
UINT16	**Cols**	Number of tile columns
UINT16	**TilesValues[1024]**	AE tiles values

*Table 5-11. Definition of **AMBA_AE_TILES_INFO_s** for AAA Statistics API **MW_IP_GET_AE_TILE_INFO()**.*

5.2.6 MW_IP_GET_AWB_TILE_INFO

API Syntax:

MW_IP_GET_AWB_TILE_INFO (UINT32 channelNo, UINT8 mode, AMBA_AWB_TILES_INFO_s *awbTilesInfo)

Function Description:

- This API is used to get AWB statistics tile information.

Parameters:

Type	Parameter	Description
UINT32	**channelNo**	The parameter is used to indicate the current channel number, such as the VIN channel number. For the single sensor application, the **channelNo** is 0.
UINT8	**mode**	Specify the AE stat. mode. 0: RGB mode 1: CFA mode
AMBA_AWB_TILES_INFO_s *	**awbTilesInfo**	Pointer to the **AMBA_AWB_TILES_INFO_s**. Please refer to Section 5.2.6.1 for more details.

*Table 5-12. Parameters for AAA Statistics API **MW_IP_GET_AWB_TILE_INFO()**.*

Returns:

Return	Description
0	Success
- 1	Failure

*Table 5-13. Returns for AAA Statistics API **MW_IP_GET_AWB_TILE_INFO()**.*

Example:

None

See Also:

None

5.2.6.1 MW_IP_GET_AWB_TILE_INFO > AMBA_AWB_TILES_INFO_s

Type	Field	Description
UINT16	**Rows**	Number of tile rows
UINT16	**Cols**	Number of tile columns
AMBA_AWB_TILES_VALUE_s	***pTilesValue**	Pointer to AWB statistics tile information. Please refer to Section 5.2.6.2 for more details.

*Table 5-14. Definition of **AMBA_AWB_TILES_INFO_s** for AAA Statistics API **MW_IP_GET_AWBTILE_INFO()**.*

5.2.6.2 MW_IP_GET_AWB_VALUE_INFO > AMBA_AWB_TILES_INFO_s

Type	Field	Description
UINT16	**R**	Tile value of **R**
UINT16	**G**	Tile value of **G**
UINT16	**B**	Tile value of **B**
UINT16	**Y**	Tile value of **Y**

*Table 5-15. Definition of **AMBA_AWB_TILES_VALUE_s** for AAA Statistics API **MW_IP_GET_AWBTILE_INFO()**.*

Appendix 1 Additional Resources

Other Ambarella documents of potential interest include:

- *Ambarella uITRON AN: Camera*
- *Ambarella uITRON AN: Middleware*
- *Ambarella uITRON AN: System*
- *Ambarella uITRON API: Camera*
- *Ambarella uITRON API: Middleware*
- *Ambarella uITRON API: System*

Please contact an Ambarella representative for digital copies.

Appendix 2 Important Notice

All Ambarella design specifications, datasheets, drawings, files, and other documents (together and separately, “materials”) are provided on an “*as is*” basis, and Ambarella makes no warranties, expressed, implied, statutory, or otherwise with respect to the materials, and expressly disclaims all implied warranties of noninfringement, merchantability, and fitness for a particular purpose. The information contained herein is believed to be accurate and reliable. However, Ambarella assumes no responsibility for the consequences of use of such information.

Ambarella Incorporated reserves the right to correct, modify, enhance, improve, and otherwise change its products and services at any time and to discontinue any product or service without notice. Customers should obtain the latest relevant information before placing orders and should verify that such information is current and complete.

All products are sold subject to Ambarella’s terms and conditions of sale supplied at the time of order acknowledgment. Ambarella warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with its standard warranty. Testing and other quality control techniques are used to the extent Ambarella deems necessary to support this warranty.

Ambarella assumes no liability for applications assistance or customer product design. Customers are responsible for their products and applications using Ambarella components. To minimize the risks associated with customer products and applications, customers should provide adequate design and operating safeguards.

Ambarella does not warrant or represent that any license, either expressed or implied, is granted under any Ambarella patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which Ambarella products or services are used. Information published by Ambarella regarding third-party products or services does not constitute a license from Ambarella to use such products or services or a warranty or endorsement thereof. Use of such information may require a license from a third party under the patents or other intellectual property of the third party, or a license from Ambarella under the patents or other intellectual property of Ambarella.

Reproduction of information from Ambarella documents is not permissible without prior approval from Ambarella.

Ambarella products are not authorized for use in safety-critical applications (such as life support) where a failure of the product would reasonably be expected to cause severe personal injury or death, unless officers of the parties have executed an agreement specifically governing such use. Customers acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of Ambarella products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by Ambarella. Further, Customers must fully indemnify Ambarella and its representatives against any damages arising out of the use of Ambarella products in such safety-critical applications.

Ambarella products are neither designed nor intended for use in automotive and military/aerospace applications or environments. Customers acknowledge and agree that any such use of Ambarella products is solely at the Customer’s risk, and they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

Appendix 3 Revision History

NOTE: Page numbers for previous drafts may differ from page numbers in the current version.

Version	Date	Comments
0.1	28 November 2013	Formatting
0.2	24 February 2014	Update in Chapter 3, Algorithm Control. Update in Chapte 4, Image Quality Control.
0.3	6 March 2014	Add new APIs in Chapter 4, Image Quality Control
0.4	8 April 2014	Update Algorithm Control and Image Quality Control.
0.5	19 May 2014	Update in MW_IP_SET_ADJ_PARAMS_ADD, MW_IP_GET_ADJ_PARAMS_ADD, add MW_IP_CHK_IQ_PARAM_VER and MW_IP_SET_IMAGE_SHARPNESS.
0.6	15 September 2014	Formatted to SDK6
0.7	29 June 2015	Update in Sections 3.4.5 and 3.4.6.
0.8	21 April 2016	Update in Sections 3.3.1, 3.3.9 and 4.3.1. Delete Section 3.3.9 MW_IP_GET_SENSOR_STATUS. Add Sections 2.4.7, 2.4.8, 3.3.10 ~ 3.3.18 and 3.4.22 ~3.4.27.

Table 3-1. Revision History.