Crystal Image through
Imaging Innovation
PIXELPLUS

SURROUND VIEW MONITORING SYSTEM

PI5008K SVM Driver

Rev 0.1

Last Update : 2018.06.08

6th Floor, 105, Gwanggyo-ro, Yeongtong-gu,
Suwong-si, Gyeonggi-do, 16229, Korea
Tel : +82-31-888-5300, FAX : +82-31-888-5399

Contents

Figure

1. Overview

1.1. Introduction

This documentation explains device driver for SVM block.

1.2. Definitions

1.2.1. FB LUT (Front & Back LookUp Table)

Lookup table for input data for Front and Back channel

1.2.2. LR LUT (Left & Right LookUp Table)

Lookup table for input data for Left and Right channel

1.2.3. BC LUT (Brightness Control LookUp Table)

Lookup table for blending of FB LUT and LR LUT and brightness control

1.2.4. BC ADD LUT (Brightness Control ADDitional LookUp Table) [Option]

Additional Lookup table to compensate curve of boundaries of BC LUT.

2. SVM Driver API

2.1. PPDRV_SVM_CTRL_GetVersion

Prototype	PP_U32 PPDRV_SVM_CTRL_GetVersion(PP_VOID);
Description	SVM Block version
Argument	
Return value	SVM Block version
Example	

2.2. PPDRV_SVM_Initialize

Prototype	PP_RESULT_E PPDRV_SVM_Initialize(_VID_RESOL enInput, _VID_RESOL enOutput, PP_U8 u8InFrameBufCnt, PP_U8 u8OutFrameBufCnt);
Description	Initialize SVM Block
Argument	enInput : Input resolution enOutput : Output resolution u8InFrameBufCnt : Number of buffer to store input data of each input cahnnel(3 or above). Default value is 3 u8OutFrameBufCnt : Number of buffer to store output data (2~4). Default value is 2
Return value	eSUCCESS eERROR_INVALID_ARGUMENT : Argument value error eERROR_SVM_INOUT_FPS : Output frame rate should be same or half of input frame rate.

Example	PP_S32 s32VinWidth, s32VinHeight, s32VoutWidth, s32VoutHeight;
	_VID_RESOL enVinResol, enVoutResol;
	
	VINAPI_GetResol(BD_SVM_IN_FMT, &s32VinWidth, &s32VinHeight, &enVinResol);
	VINAPI_GetResol(BD_SVM_OUT_FMT, &s32VoutWidth, &s32VoutHeight, &enVoutResol);

	PPDRV_SVM_Initialize(enVinResol, enVoutResol, 3, 2);

2.3. PPDRV_SVM_IN_SetAddress

Prototype	PP_RESULT_E PPDRV_SVM_IN_SetAddress(PP_CHANNEL_E IN enChannel, PP_U32* IN pu32Addr);
Description	Set the address of input buffer
Argument	enChannel : Input channel typedef enum ppCHANNEL_E eCHANNEL_FRONT eCHANNEL_LEFT, eCHANNEL_RIGHT, eCHANNEL_REAR } PP_CHANNEL_E; pu32Addr : Memory base address (16byte align)
Return value	eSUCCESS eERROR_INVALID_ARGUMENT : Argument value error eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialized eERROR_INVALID_ALIGN : address의 16byte align error
Example	

2.4. PPDRV_SVM_IN_GetAddress

Prototype	PP_U32* PPDRV_SVM_IN_GetAddress(PP_CHANNEL_E IN enChannel);
Description	Get input buffer addess
Argument	enChannel : Input channel (see. 2.3)
Return value	Base address of input buffer PP_NULL
Example	

2.5. PPDRV_SVM_IN_SetMirroring

Prototype	PP_RESULT_E PPDRV_SVM_IN_SetMirroring(

	PP_CHANNEL_E IN enChannel, PP_BOOL IN bHorizontal, PP_BOOL IN bVertical);
Description	Set left/right and top/bottom mirroring of each channel
Argument	enChannel : Input channel (see. 2.3) bHorizontal : Set left/right mirroring bVertical : Set top/bottom mirroring
Return value	eSUCCESS eERROR_INVALID_ARGUMENT : Argument value error
Example	

2.6. PPDRV_SVM_IN_SetAntiAliasing

Prototype	PP_RESULT_E PPDRV_SVM_IN_SetAntiAliasing(PP_CHANNEL_E IN enChannel, PP_SVMDRV_ANTI_ALIASING_STRENGTH_H_E IN enHorizotal, PP_SVMDRV_ANTI_ALIASING_STRENGTH_V_E IN enVertical);
Description	Set anti-aliasing for each input channel
Argument	enChannel : Input channel (see. 2.3) enHorizotal : Horizontal anti-aliasing value typedef enum ppSVMDRV_ANTI_ALIASING_STRENGTH_H_E { eSVMDRV_AA_H_1 = 0, eSVMDRV_AA_H_2, eSVMDRV_AA_H_3, eSVMDRV_AA_H_4, eSVMDRV_AA_H_5, eSVMDRV_AA_H_6, eSVMDRV_AA_H_7, eSVMDRV_AA_H_MAX, } PP_SVMDRV_ANTI_ALIASING_STRENGTH_H_E; enVertical: Vertical anti-aliasing value typedef enum ppSVMAPI_ANTI_ALIASING_STRENGTH_V_E { eSVMDRV_AA_V_1 = 0, eSVMDRV_AA_V_2,

	eSVMDRV_AA_V_3, eSVMDRV_AA_V_4, eSVMDRV_AA_V_MAX, } PP_SVMDRV_ANTI_ALIASING_STRENGTH_V_E;
Return value	eSUCCESS eERROR_INVALID_ARGUMENT : Argument value error
Example	

2.7. PPDRV_SVM_INOUT_SetReplaceColor

Prototype	PP_VOID PPDRV_SVM_INOUT_SetReplaceColor(PP_U8 IN u8Y, PP_U8 IN u8Cb, PP_U8 IN u8Cr);
Description	Set replace color which is used instead of input output data
Argument	u8Y u8Cb u8Cr
Return value	
Example	

2.8. PPDRV_SVM_IN_SetReplaceColorEnable

Prototype	PP_VOID PPDRV_SVM_IN_SetReplaceColorEnable(PP_CHANNEL_E IN enChannel, PP_BOOL IN bOn);
Description	Decide whether to use replace color instead of input data
Argument	enChannel : Input channel (see. 2.3) bOn : PP_TRUE or PP_FALSE
Return value	
Example	

2.9. PPDRV_SVM_IN_GetReplaceColorEnable

Prototype	PP_BOOL PPDRV_SVM_IN_GetReplaceColorEnable(PP_CHANNEL_E IN enChannel);

Description	Get the setting whether to use replace color which is used instead of input data
Argument	enChannel : Input channel (see. 2.3)
Return value	PP_TRUE PP_FALSE
Example	

2.10.PPDRV_SVM_IN_SetEnable

Prototype	PP_RESULT_E PPDRV_SVM_IN_SetEnable(PP_BOOL IN bFrontEnable, PP_BOOL IN bLeftEnable, PP_BOOL IN bRightEnable, PP_BOOL IN bRearEnable);
Description	Decide whether to use each input channel
Argument	enChannel : Input channel (see. 2.3)
Return value	eSUCCESS eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialized
Example	

2.11.PPDRV_SVM_CTRL_SetImage

Prototype	PP_RESULT_E PPDRV_SVM_CTRL_SetImage(PP_SVMDRV_IMG_NUMBER_E IN enImgNum, PP_RECT_S IN stRect, PP_U8 IN u8Alpha);
Description	Set transparency and rectangle area of YUV422(YUYV) image
Argument	enImgNum : image number (Currently 2 images are supported.) typedef enum ppSVMDRV_IMG_NUMBER_E { eSVMDRV_IMG_NUM_0 = 0, eSVMDRV_IMG_NUM_1, eSVMDRV_IMG_NUM_MAX, } PP_SVMDRV_IMG_NUMBER_E; stRect : Rectangle area typedef struct ppRECT_S

	{
	PP_U16 u16X;
	PP_U16 u16Y;
	PP_U16 u16Width;
	PP_U16 u16Height;
	} PP_RECT_S;
	u8Alpha : Transparency (0~63)

Return value	eSUCCESS eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialized eERROR_INVALID_ARGUMENT : Argument value error
Example	

2.12.PPDRV_SVM_CTRL_SetImageMaskColor

Prototype	PP_VOID PPDRV_SVM_CTRL_SetImageMaskColor(PP_SVMDRV_IMG_NUMBER_E IN enImgNum, PP_U8 IN u8Y, PP_U8 IN u8Cb, PP_U8 IN u8Cr);
Description	Set mask color of YUV422(YUYV) image Mask color will not be displayed.
Argument	enImgNum : image number (see. 2.11) u8Y u8Cb u8Cr
Return value	
Example	

2.13.PPDRV_SVM_CTRL_SetImageAddr

Prototype	PP_RESULT_E PPDRV_SVM_CTRL_SetImageAddr(PP_SVMDRV_IMG_NUMBER_E IN enImgNum, PP_U32* IN pu32Addr4odd, PP_U32* IN pu32Addr4even);
Description	Set memory base address of YUV422(YUYV) image data One image is needed if camera input is Progressive. Odd and even image

	are required for intelace mode.
Argument	enImgNum : image number (see. 2.11) pu32Addr4odd : Memory base address for progressive or odd field for intelace mode (16byte align) pu32Addr4even : Memory base address for even field for Interace mode (16byte align). PP_NULL is used for progressive.
Return value	eSUCCESS eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialized eERROR_INVALID_ARGUMENT : Argument value error eERROR_INVALID_ALIGN : 16byte align error (Address is must be aligned by 16 bytes)
Example	

2.14.PPDRV_SVM_CTRL_GetImageAddr

Prototype	PP_U32* PPDRV_SVM_CTRL_GetImageAddr(PP_SVMDRV_IMG_NUMBER_E IN enImgNum, PP_FIELD_E IN enField);
Description	Obtain address of YUV422(YUYV) image

Argument	enImgNum : image number (see. 2.11)
	enField : input field
	typedef enum ppFIELD_E
	{
	eFIELD_NONE = 0, // progressive
	eFIELD_ODD = 0, // odd field of interlace
	eFIELD_EVEN, // even field of interlace
	eFIELD_MAX,
	} PP_FIELD_E;

Return value	Base address of image buffer PP_NULL
Example	

2.15.PPDRV_SVM_CTRL_SetImageEnable

Prototype	PP_RESULT_E PPDRV_SVM_CTRL_SetImageEnable(PP_SVMDRV_IMG_NUMBER_E IN enImgNum,

	PP_BOOL IN bEnable);
Description	Set whether to use YUV422(YUYV) image
Argument	enImgNum : image number (see. 2.11) bEnable : PP_TRUE or PP_FALSE
Return value	eSUCCESS eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialzed eERROR_INVALID_ARGUMENT : Argument value error
Example	

2.16.PPDRV_SVM_CTRL_SetEdgeEnhancementFixedGain

Prototype	PP_VOID PPDRV_SVM_CTRL_SetEdgeEnhancementFixedGain(PP_U8 IN u8Gain);
Description	Set gain of Edge enhancement
Argument	u8Gain
Return value	
Example	

2.17.PPDRV_SVM_CTRL_SetEdgeEnhancementEnable

Prototype	PP_RESULT_E PPDRV_SVM_CTRL_SetEdgeEnhancementEnable(PP_BOOL IN bEnable);
Description	Decide whether to use Edge enhancement
Argument	bEnable : PP_TRUE or PP_FALSE
Return value	eSUCCESS eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialized
Example	

2.18.PPDRV_SVM_CTRL_SetDynamicblendingCoefficient

Prototype	PP_VOID PPDRV_SVM_CTRL_SetDynamicblendingCoefficient(PP_U8 u8Coef1, PP_U8 u8Coef3, PP_U8 u8Coef6, PP_U8 u8Coef8);
Description	Set coeffieient value of dynamic blending (see. Fig 1)

Argument	u8Coef1 : Blend ratio value for the area where Front and Left are overlapped u8Coef3 : Blend ratio value for the area where Front and Right are overlapped u8Coef6 : Blend ratio value for the area where Rear and Left are overlapped u8Coef8 : Blend ratio value where Rear and Right are overlapped
Return value	
Example	

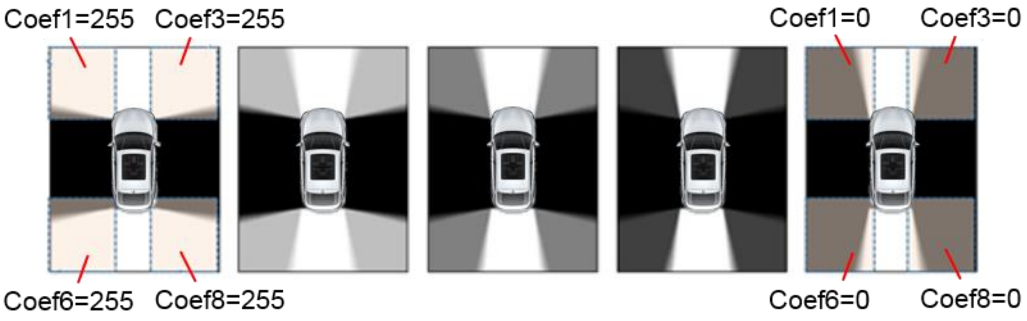

Figure 1 Coefficient of dynamic blending

2.19.PPDRV_SVM_CTRL_SetDynamicblendingEnable

Prototype	PP_RESULT_E PPDRV_SVM_CTRL_SetDynamicblendingEnable(PP_BOOL IN bEnable);
Description	Decide whether to use Dynamic blending
Argument	bEnable : PP_TRUE or PP_FALSE
Return value	eSUCCESS eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialized
Example	

2.20.PPDRV_SVM_CTRL_GetRGBStatistics

Prototype	PP_VOID PPDRV_SVM_CTRL_GetRGBStatistics(PP_SVMDRV_RGB_STATISTICS_S* OUT pstStats);
Description	Obtain the statistical value of RGB color for each area It will be used for Brightness control

Argument	pstStats : Structure for RGB statistical value (see. Fig 2) typedef enum ppSVMDRV_OUTPUT_PART_E { eSVMDRV_OUTPUT_PART_FB = 0, eSVMDRV_OUTPUT_PART_LR, eSVMDRV_OUTPUT_PART_MAX, } PP_SVMDRV_OUTPUT_PART_E; typedef enum ppSVMDRV_SUBPART_E { eSVMDRV_SUBPART_FL = 0, eSVMDRV_SUBPART_FR, eSVMDRV_SUBPART_BR, eSVMDRV_SUBPART_BL, eSVMDRV_SUBPART_MAX, } PP_SVMDRV_SUBPART_E; typedef enum ppSVMDRV_COLOR_E { eSVMDRV_COLOR_R = 0, eSVMDRV_COLOR_G, eSVMDRV_COLOR_B, eSVMDRV_COLOR_MAX, } PP_SVMDRV_COLOR_E; typedef struct ppSVMDRV_RGB_STATISTICS_S { PP_U32 u32Value[eSVMDRV_OUTPUT_PART_MAX] [eSVMDRV_SUBPART_MAX][eSVMDRV_COLOR_MAX]; } PP_SVMDRV_RGB_STATISTICS_S;
Return value	eSUCCESS eERROR_INVALID_ARGUMENT : Argument value error
Example	

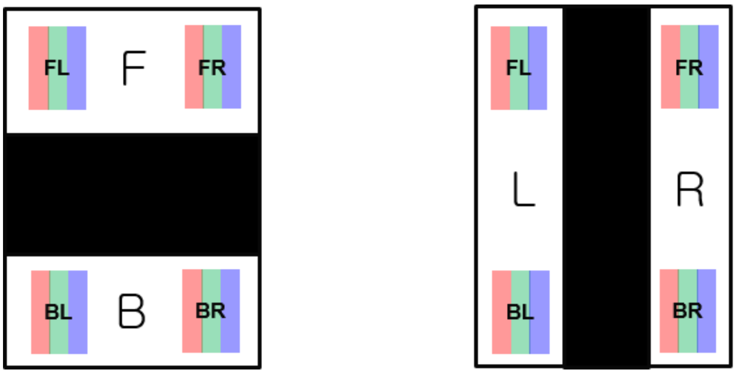

Figure 2 Area of RGB Statistics

2.21.PPDRV_SVM_CTRL_SetBCCoefficient

Prototype	PP_RESULT_E PPDRV_SVM_CTRL_SetBCCoefficient(PP_SVMDRV_BCCOEF_S* IN pstCoefBC);
Description	Set coefficient value of RGB of each area for Brightness control (see. Fig 3)
Argument	pstCoefBC : Coefficient structure for BC typedef enum ppSVMDRV_COLOR_E { eSVMDRV_COLOR_R = 0, eSVMDRV_COLOR_G, eSVMDRV_COLOR_B, eSVMDRV_COLOR_MAX, } PP_SVMDRV_COLOR_E; typedef enum ppSVMDRV_FB_PART_E { eSVMDRV_FB_PART_F = 0, eSVMDRV_FB_PART_B, eSVMDRV_FB_PART_MAX, } PP_SVMDRV_FB_PART_E; typedef enum ppSVMDRV_LR_PART_E { eSVMDRV_LR_PART_L, eSVMDRV_LR_PART_R, eSVMDRV_LR_PART_MAX,

	} PP_SVMDRV_LR_PART_E;
	
	typedef enum ppSVMDRV_SUBPART_E
	{
	eSVMDRV_SUBPART_FL = 0,
	eSVMDRV_SUBPART_FR,
	eSVMDRV_SUBPART_BR,
	eSVMDRV_SUBPART_BL,
	eSVMDRV_SUBPART_MAX,
	} PP_SVMDRV_SUBPART_E;
	
	typedef struct ppSVMDRV_BCCOEF_S
	{
	PP_U8 u8Coef1[eSVMDRV_COLOR_MAX][eSVMDRV_SUBPART_MAX];
	PP_U8 u8Coef2[eSVMDRV_COLOR_MAX][eSVMDRV_FB_PART_MAX];
	PP_U8 u8Coef3[eSVMDRV_COLOR_MAX][eSVMDRV_SUBPART_MAX];
	PP_U8 u8Coef4[eSVMDRV_COLOR_MAX][eSVMDRV_LR_PART_MAX];
	} PP_SVMDRV_BCCOEF_S;

Return value	eSUCCESS eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialzed eERROR_INVALID_ARGUMENT : Argument value error
Example	

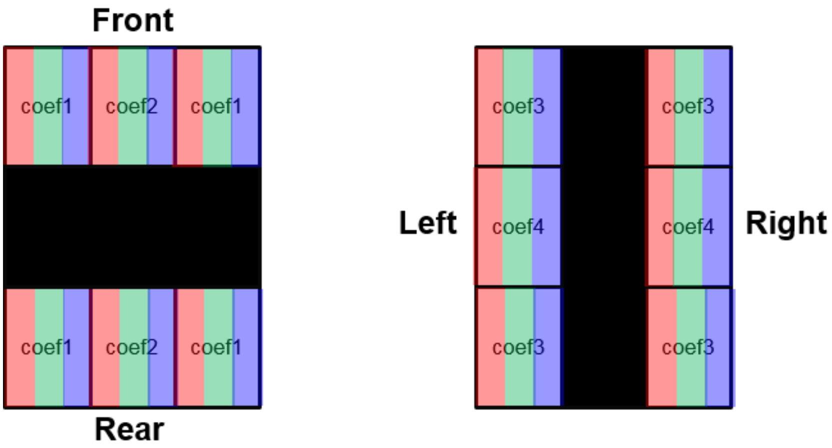

Figure 3 Area of BC RGB Coefficient

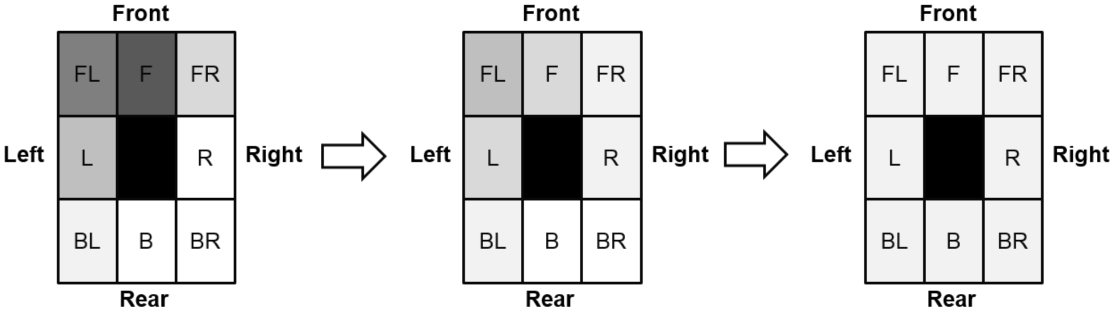

Figure 4 BC Flow

2.22.PPDRV_SVM_CTRL_SetMorphingRatio

Prototype	PP_VOID PPDRV_SVM_CTRL_SetMorphingRatio(PP_U8 IN u8FBLUTRatio, PP_U8 IN u8LRLUTRatio);
Description	Set Morphing ratio (see. Fig 5)
Argument	u8FBLUTRatio : FB LUT의 Morphing ratio u8LRLUTRatio : LR LUT의 Morphing ratio
Return value	
Example	

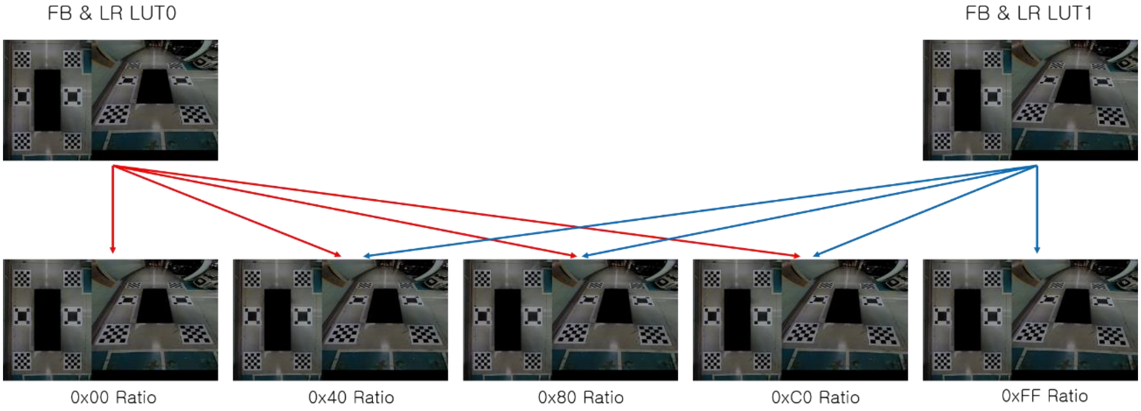

Figure 5 Morphing Ratio

2.23.PPDRV_SVM_CTRL_SetFBLUTAddress

Prototype	PP_RESULT_E PPDRV_SVM_CTRL_SetFBLUTAddress(PP_U32* IN pu32Addr4odd,

	PP_U32* IN pu32Addr4even, PP_SVMDRV_MORPHING_NUM_E enNum);
Description	Set memory base address of FB LUT

Argument	pu32Addr4odd : Memory base address for progressive or odd fiel for intelace mode (16byte align)
	pu32Addr4even : Memory base address for even field for Interace mode (16byte align). PP_NULL is used for progressive mode
	enNum : Address when Morphing is used (see. Fig 5)
	typedef enum ppSVMDRV_MORPHING_NUM_E
	{
	eSVMDRV_MORPING_NUM_0 = 0, // FB & LR LUT0 of Fig 5
	eSVMDRV_MORPING_NUM_1, // FB & LR LUT1 of Fig 5
	eSVMDRV_MORPING_NUM_MAX,
	} PP_SVMDRV_MORPHING_NUM_E;

Return value	eSUCCESS eERROR_SVM_NOT_INITIALIZE : SVM Block is not initializedeERROR_INVALID_ARGUMENT : Argument value error
Example	

2.24.PPDRV_SVM_CTRL_GetFBLUTAddress

Prototype	PP_RESULT_E PPDRV_SVM_CTRL_GetFBLUTAddress(PP_U32** OUT pu32Addr4odd, PP_U32** OUT pu32Addr4even, PP_SVMDRV_MORPHING_NUM_E enNum);
Description	Obtain memory base address of FB LUT
Argument	pu32Addr4odd : Memory base address for progressive or odd fiel for intelace mode pu32Addr4even : Memory base address for even field for Interace mode enNum : Address when morphing is used (see. 2.23)
Return value	eSUCCESS eERROR_INVALID_ARGUMENT : Argument value error
Example	

2.25.PPDRV_SVM_CTRL_SetLRLUTAddress

Prototype	PP_RESULT_E PPDRV_SVM_CTRL_SetLRLUTAddress(PP_U32* IN pu32Addr4odd, PP_U32* IN pu32Addr4even, PP_SVMDRV_MORPHING_NUM_E enNum);
Description	Set base address of LR LUT
Argument	pu32Addr4odd : Memory base address for progressive or odd fiel for intelace mode (16byte align) pu32Addr4even : Memory base address for even field for Interace mode (16byte align). PP_NULL is used for progressive mode enNum : Address when morphing is used (see. 2.23)
Return value	eSUCCESS eERROR_SVM_NOT_INITIALIZE : SVM Block is not intialized eERROR_INVALID_ARGUMENT : Argument value error
Example	

2.26.PPDRV_SVM_CTRL_GetLRLUTAddress

Prototype	PP_RESULT_E PPDRV_SVM_CTRL_GetLRLUTAddress(PP_U32** OUT pu32Addr4odd, PP_U32** OUT pu32Addr4even, PP_SVMDRV_MORPHING_NUM_E enNum);
Description	Obtain memory base address of LR LUT의 memory base address
Argument	pu32Addr4odd : Memory base address for progressive or odd fiel for intelace mode pu32Addr4even : Memory base address for even field for Interace mode enNum : Address when morphing is used (see. 2.23)
Return value	eSUCCESS eERROR_INVALID_ARGUMENT : Argument value error
Example	

2.27.PPDRV_SVM_CTRL_SetBCLUTAddress

Prototype	PP_RESULT_E PPDRV_SVM_CTRL_SetLRLUTAddress(

	PP_U32* IN pu32Addr4odd, PP_U32* IN pu32Addr4even);
Description	Set memory address of BC LUT
Argument	pu32Addr4odd : Memory base address for progressive or odd fiel for intelace mode (16byte align) pu32Addr4even : Memory base address for even field for Interace mode (16byte align). PP_NULL is used for progressive mode
Return value	eSUCCESS eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialized. eERROR_INVALID_ARGUMENT : Argument value error
Example	

2.28.PPDRV_SVM_CTRL_GetBCLUTAddress

Prototype	PP_RESULT_E PPDRV_SVM_CTRL_GetLRLUTAddress(PP_U32** OUT pu32Addr4odd, PP_U32** OUT pu32Addr4even);
Description	Get memory base address of BC LUT
Argument	pu32Addr4odd : Memory base address for progressive or odd fiel for intelace mode pu32Addr4even : Memory base address for even field for Interace mode
Return value	eSUCCESS eERROR_INVALID_ARGUMENT : Argument value error
Example	

2.29.PPDRV_SVM_CTRL_SetBCAdditionalLUT [Option]

Prototype	PP_RESULT_E PPDRV_SVM_CTRL_SetBCAdditionalLUT(PP_SVMDRV_BC_ADDITIONAL_LUT_E enType, PP_SVMDRV_BC_ADDITIONAL_LUT_S* IN stBCAdd);
Description	Set memory base address of BC Additional LUT
Argument	enType: BC Additional LUT types typedef enum ppSVMDRV_BC_ADDITIONAL_LUT_E { eSVMDRV_BC_ADD_LUT_ALPHA_0 = 0, eSVMDRV_BC_ADD_LUT_ALPHA_1,

	eSVMDRV_BC_ADD_LUT_GRADIENT, eSVMDRV_BC_ADD_LUT_MAX, } PP_SVMDRV_BC_ADDITIONAL_LUT_E; stBCAdd : BC additional LUT의 size와 memory base address typedef enum ppSVMDRV_BC_ADDITIONAL_LUT_SUBCORE_E { eSVMDRV_BC_ADD_LUT_SUBCORE_0 = 0, eSVMDRV_BC_ADD_LUT_SUBCORE_1, eSVMDRV_BC_ADD_LUT_SUBCORE_MAX, } PP_SVMDRV_BC_ADDITIONAL_LUT_SUBCORE_E; typedef struct ppSVMDRV_BC_ADDITIONAL_LUT_S { PP_U16 u16TotalCnt[eFIELD_MAX] [eSVMDRV_BC_ADD_LUT_SUBCORE_MAX]; PP_U32* pu32Addr[eFIELD_MAX] [eSVMDRV_BC_ADD_LUT_SUBCORE_MAX]; } PP_SVMDRV_BC_ADDITIONAL_LUT_S;
Return value	eSUCCESS eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialized eERROR_INVALID_ARGUMENT : Argument value error
Example	

2.30.PPDRV_SVM_CTRL_SetOutputMode

Prototype	PP_RESULT_E PPDRV_SVM_CTRL_SetOutputMode(PP_SVMDRV_OUTPUTMODE_E enOutputMode);
Description	Set output mode

Argument	enOutputMode : Output mode
	typedef enum ppSVMDRV_OUTPUTMODE_E
	{
	eSVMDRV_OUTPUTMODE_BYPASS_FRONT = 0,
	// Output front input
	eSVMDRV_OUTPUTMODE_BYPASS_LEFT,
	// Output left input
	eSVMDRV_OUTPUTMODE_BYPASS_RIGHT,

	// Output right input
	eSVMDRV_OUTPUTMODE_BYPASS_REAR,
	// Output Rear input
	eSVMDRV_OUTPUTMODE_LUT,
	// Output using FB, LR, BC LUT
	eSVMDRV_OUTPUTMODE_QUAD,
	// Output Quad image from Front, Left, Right, Rear input
	eSVMDRV_OUTPUTMODE_MAX,
	} PP_SVMDRV_OUTPUTMODE_E;

Return value	eSUCCESS eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialized eERROR_INVALID_ARGUMENT : Argument value error
Example	

2.31.PPDRV_SVM_CTRL_GetLuminanceAverage

Prototype	PP_VOID PPDRV_SVM_CTRL_GetLuminanceAverage(PP_U16* OUT pu16Front, PP_U16* OUT pu16Left, PP_U16* OUT pu16Right, PP_U16* OUT pu16Rear);
Description	Obtain luminance average value of input channels
Argument	pu16Front : Luminance average value of front channel pu16Left : Luminance average value of left channel pu16Right : Luminance average value of right channel pu16Rear : Luminance average value of rear channel
Return value	
Example	

2.32.PPDRV_SVM_CTRL_SetEnable

Prototype	PP_RESULT_E PPDRV_SVM_CTRL_SetEnable(PP_BOOL IN bEnable);
Description	SVM Block On or Off
Argument	bEnable : PP_TRUE or PP_FALSE
Return value	eSUCCESS

	eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialized
Example	

2.33.PPDRV_SVM_CTRL_GetEnable

Prototype	PP_BOOL PPDRV_SVM_CTRL_GetEnable(PP_VOID);
Description	Obtain on/off state of SVM Block
Argument	
Return value	PP_TRUE : SVM Block On PP_TRUE : SVM Block Off
Example	

2.34.PPDRV_SVM_OUT_SetAddress

Prototype	PP_RESULT_E PPDRV_SVM_OUT_SetAddress(
	    PP_SVMDRV_OUT_FRAMEBUF_NUM_E IN enOutFBNum,
	    PP_U32* IN pu32Addr);

Description	Set address of output buffer

Argument	enOutFBNum : Number of output buffer
	    typedef enum ppSVMDRV_OUT_FRAMBUF_NUM_E
	    {
	        eSVMDRV_OUT_FB_NUM_0 = 0,
	        eSVMDRV_OUT_FB_NUM_1,
	        eSVMDRV_OUT_FB_NUM_2,
	        eSVMDRV_OUT_FB_NUM_3,
	        eSVMDRV_OUT_FB_NUM_MAX,
	    } PP_SVMDRV_OUT_FRAMEBUF_NUM_E;
	pu32Addr : Memory base address (16byte align)

Return value	eSUCCESS eERROR_INVALID_ARGUMENT : Argument value error eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialized eERROR_INVALID_ALIGN : address의 16byte align error
Example	

2.35.PPDRV_SVM_OUT_GetAddress

Prototype	PP_U32* PPDRV_SVM_OUT_GetAddress(PP_SVMDRV_OUT_FRAMEBUF_NUM_E IN enOutFBNum);
Description	Obtain memory base address of Output buffer
Argument	enOutFBNum : Output buffer number
Return value	Output buffer base address or PP_NULL
Example	

2.36.PPDRV_SVM_OUT_SetReplaceColorEnable

Prototype	PP_VOID PPDRV_SVM_OUT_SetReplaceColorEnable(PP_BOOL IN bOn);
Description	Decide whether to use replace color which is used instead of output data Replace color : (see. 2.8)
Argument	bOn : PP_TRUE or PP_FALSE
Return value	
Example	

2.37.PPDRV_SVM_IN_GetReplaceColorEnable

Prototype	PP_BOOL PPDRV_SVM_OUT_GetReplaceColorEnable(PP_VOID);
Description	Get the setting whether to use replace color which is used instead of output data
Argument	
Return value	PP_TRUE PP_FALSE
Example	

2.38.PPDRV_SVM_CTRL_SetBackgroundColor

Prototype	PP_VOID PPDRV_SVM_CTRL_SetBackgroundColor(PP_U8 IN u8Y, PP_U8 IN u8Cb, PP_U8 IN u8Cr);

Description	Set Background color
Argument	u8Y u8Cb u8Cr
Return value	
Example	

2.39.PPDRV_SVM_OUT_SetSection

Prototype	PP_RESULT_E PPDRV_SVM_OUT_SetSection(PP_SVMDRV_SECTION_NUMBER_E IN enSectionNumber, PP_RECT_S IN stSectionRect);
Description	Set section (see. Fig 6)
Argument	enSectionNumber : Section Number typedef enum ppSVMDRV_SECTION_NUMBER_E { eSVMDRV_SECTION_NUM_0 = 0, eSVMDRV_SECTION_NUM_1, eSVMDRV_SECTION_NUM_2, eSVMDRV_SECTION_NUM_3, eSVMDRV_SECTION_NUM_MAX, } PP_SVMDRV_SECTION_NUMBER_E; stSectionRect : Section are (see. 2.11)
Return value	eSUCCESS eERROR_INVALID_ARGUMENT : Argument value error eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialized
Example	

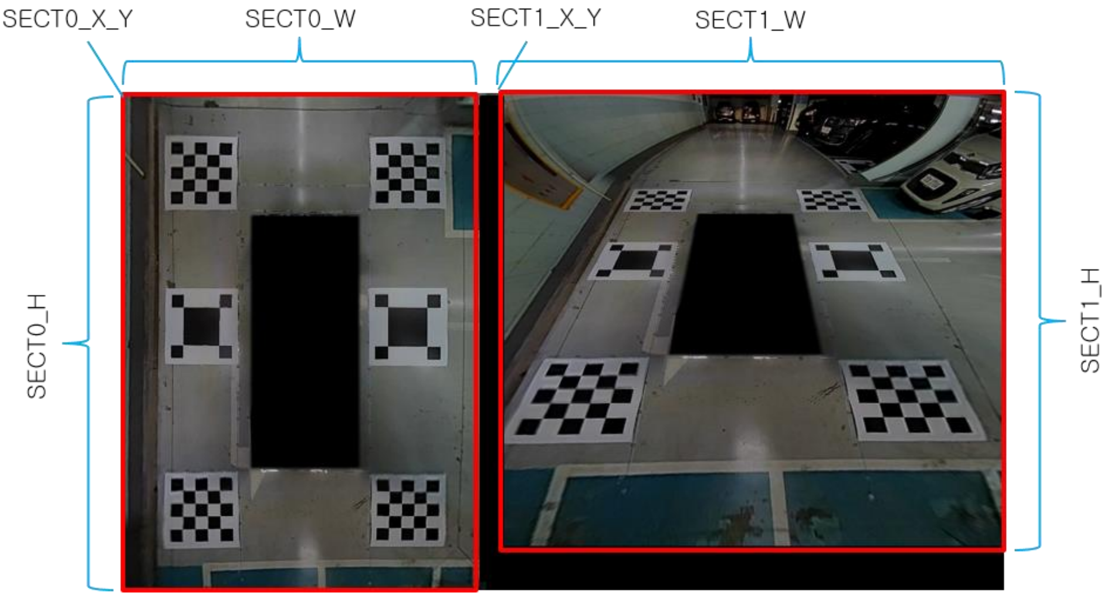

Figure 6 Section Area

2.40.PPDRV_SVM_OUT_SetSectionEnable

Prototype	PP_RESULT_E PPDRV_SVM_OUT_SetSectionEnable(PP_SVMDRV_SECTION_NUMBER_E IN enSectionNumber, PP_BOOL IN bEnable);
Description	Decide whether to use section
Argument	enSectionNumber : (see. 2.39) bEnable : PP_TRUE or PP_FALSE
Return value	eSUCCESS eERROR_INVALID_ARGUMENT : Argument value error eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialized
Example	

2.41.PPDRV_SVM_OUT_SetTilt

Prototype	PP_RESULT_E PPDRV_SVM_OUT_SetTilt(PP_S8 IN s8X, PP_S8 IN s8Y);
Description	Change start position of output screen (see. Fig 7)
Argument	s8X : -63 ~ 63 s8Y : -63 ~ 63

Return value	eSUCCESS eERROR_INVALID_ARGUMENT : Argument value error eERROR_SVM_NOT_INITIALIZE : SVM Block is not intialized
Example	

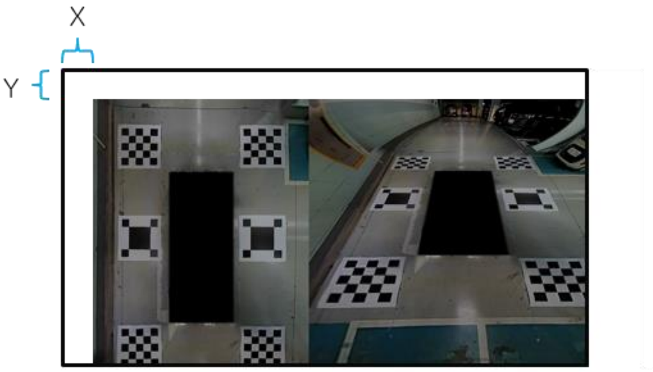

Figure 7 Tilt

2.42. PPDRV_SVM_OUT_SetHold

Prototype	PP_RESULT_E PPDRV_SVM_OUT_SetHold(PP_BOOL IN bEnable);
Description	Pause output screen
Argument	bEnable : PP_TRUE or PP_FALSE
Return value	eSUCCESS eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialized
Example	

2.43. PPDRV_SVM_OUT_GetHoldFrameBufferAddress

Prototype	PP_U32* PPDRV_SVM_OUT_GetHoldFrameBufferAddress(PP_FIELD_E IN enField);
Description	Obtain memory base address of output screen which is paused
Argument	enField : (see. 2.14)
Return value	eSUCCESS eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialized
Example	

2.44.PPDRV_SVM_OUT_SetEnable

Prototype	PP_RESULT_E PPDRV_SVM_OUT_SetEnable(PP_BOOL IN bFBEnable, PP_BOOL IN bLREnable);
Description	Decide whether to write data to output buffer
Argument	bFBEnable : Front, Rear output bLRenable : Left, Right output
Return value	eSUCCESS eERROR_SVM_NOT_INITIALIZE : SVM Block is not initialized
Example	

2.45.PPDRV_SVM_ISR_SetVsyncHandler

Prototype	PP_VOID PPDRV_SVM_ISR_SetVsyncHandler(PPDRV_SVM_CALLBACK_VSYNC IN callback);
Description	Register Vsync handler. Regstered handler will be called every Vsync
Argument	callback : handler function pointer typedef PP_VOID (*PPDRV_SVM_CALLBACK_VSYNC) (PP_VOID);
Return value	
Example	

2.46.PPDRV_SVM_ISR_SetErrorHandler

Prototype	PP_VOID PPDRV_SVM_ISR_SetErrorHandler(PPDRV_SVM_CALLBACK_ERROR IN callback);
Description	Register Error handler. Registered handler will be called when a problem is happned in AXI BUS
Argument	callback : handler function pointer typedef PP_VOID (*PPDRV_SVM_CALLBACK_ERROR) (PP_VOID);
Return value	
Example	

evision History

Version	Date	Description
V0.1	20180608	