Texas Instruments

DSP/BIOS PREVIEWER Device Driver

User's Manual

January 2007 *Platform Software Group*

Document Version Post-silicon Release 0.5.0

IMPORTANT NOTICE

Texas Instruments Incorporated and its subsidiaries (TI) reserve the right to make corrections, modifications, enhancements, improvements, and other changes to 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 TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its hardware products to the specifications applicable at the time of sale in accordance with TI's standard warranty. Testing and other quality control techniques are used to the extent TI deems necessary to support this warranty. Except where mandated by government requirements, testing of all parameters of each product is not necessarily performed.

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

TI does not warrant or represent that any license, either express or implied, is granted under any TI patent right, copyright, mask work right, or other TI intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information published by TI regarding third-party products or services does not constitute a license from TI 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 TI under the patents or other intellectual property of TI.

Reproduction of information in TI data books or data sheets is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. Reproduction of this information with alteration is an unfair and deceptive business practice. TI is not responsible or liable for such altered documentation.

Resale of TI products or services with statements different from or beyond the parameters stated by TI for that product or service voids all express and any implied warranties for the associated TI product or service and is an unfair and deceptive business practice. TI is not responsible or liable for any such statements.

Following are URLs where you can obtain information on other Texas Instruments products and application solutions:

Products		Applications	
Amplifiers	amplifier.ti.com	Audio	www.ti.com/audio
Data Converters	dataconverter.ti.com	Automotive	www.ti.com/automotive
DSP	dsp.ti.com	Broadband	www.ti.com/broadband
Interface	interface.ti.com	Digital Control	www.ti.com/digitalcontrol
Logic	logic.ti.com	Military	www.ti.com/military
Power Mgmt	power.ti.com	Optical Networking	www.ti.com/opticalnetwork
Microcontrollers	microcontroller.ti.com	Security	www.ti.com/security
		Telephony	www.ti.com/telephony
		Video & Imaging	www.ti.com/video
		Wireless	www.ti.com/wireless

Mailing Address:
Texas Instruments
Post Office Box 655303, Dallas, Texas 75265

Read This First

About This Manual

The API reference guide serves as a software programmer's handbook for working with the PREVIEWER device driver modules. This reference guide provides necessary information regarding how to use these modules in user systems and applications.

Abbreviations

Table of Abbreviations

Abbreviation	Description
API	Application Programming Interface
DDC	Device Driver Core
IOM	Device Driver Adapter
ISR	Interrupt Service Routine
OS	Operating System
ROM	Read Only Memory
SOC	System On Chip
CSL	Chip Support Library
PAL	Platform Abstraction Layer

Revision History

Date	Author	Comments	Version
November 15,2006	EI2	Created the document	1.0.0
November 20,2006	EI2	Release to TI	Pre-silicon Release 0.3.0
November 29,2006	EI2	Release to TI	Post-silicon Release 0.3.0
December 25, 2006	EI2	Description added for APIs and data structures	Post-silicon Release 0.3.0
December 29, 2006	EI2	On-the-fly applications are added.	Post-silicon Release 0.3.0
January 2, 2007	EI2	Section 1.6 added.	Post-silicon Release 0.4.0
January 16, 2007	EI2	Bios version changed in Section 1.5	Post-silicon Release 0.4.1
January 25, 2007	EI2	Abbreviations and Section 3.1.3 has updated and Bios Version Changed	Post-silicon Release 0.5.0

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

This document is an API reference guide on DSP/BIOS PREVIEWER Device Driver for DM6437 SOC.

1.1 H/W S/W Support

This PREVIEWER Device driver has been developed for the DSP/BIOS operating system using the TI supplied Chip Support Library. For more details on the version numbers refer to the release notes in the root of the installation.

1.2 Driver Components

The driver is constituted of following sub components:

PREVIEWER IOM – Application facing, OS Specific Adaptation of Previewer Device Driver

PREVIEWER DDC –OS Independent part of Previewer Driver Core

PREVIEWER CSL –The low-level Previewer h/w abstraction module

System components:

PALOS – DSP/BIOS Abstraction

CSL– Non functional h/w abstraction.

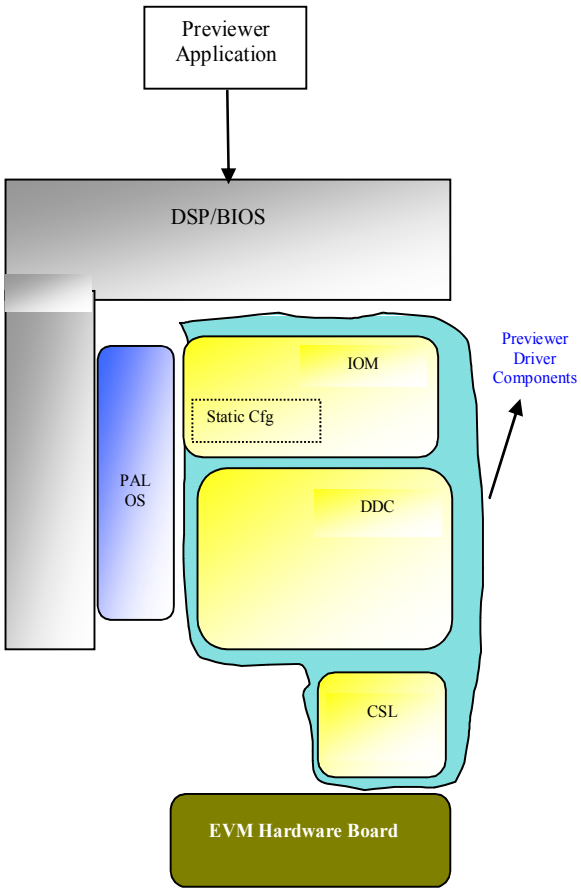

Figure 1 Device Driver Functional Decomposition

1.3 Default Driver Configuration

Previewer driver does not have any default configuration support. Before using the driver application should configure the driver with valid configurations. In case the driver recognizes invalid input parameters, it will return the corresponding error code.

1.4 Driver Capabilities

The significant driver features are:

- Previewer driver is a Loadable Module.
- Previewer driver supports input image in Bayer pattern
- Previewer driver works synchronously.
- Previewer driver supports input from SDRAM or DDRAM or from CCDC.
- Previewer driver converts input image in Bayer pattern to image in YCbCr 4:2:2 format

1.5 System Requirements

The ***DSP/BIOS PREVIEWER Device Driver*** is supported on platforms characterized by the following requirements.

Hardware:

- Target Board: DM6437 EVM Board
- Emulation Setup: XDS 510 USB Emulator

Software:

- CCS 3.3.38
- Code generation tools: V6.0.8
- Operating System: DSP/BIOS 5.31.02

1.6 Constraints

The Following is a list of driver and register configuration constraints:

- Memory output/input line offset and address should be on 32 byte boundaries.
- Output Width should be less than or equal to 1280.
- Output Width should be even.
- Input Width should be a multiple of the down sampling rate.
- When Source is CCDC, Input height should be smaller than CCD Controller output height.
- When Source is CCDC, Input width should be at least 4 pixels smaller than CCD Controller output width. SPH at least 2, EPH at least 2 pixels before last pixel from CCDC.
- PSP_PREV_IOCTL_SET_PARAMS Ioctl can be called only when Previewer is disabled.

Chapter 2

INSTALLATION GUIDE

2.1 Component Folder

Upon installing the Previewer driver the following directory structure is found in the driver's directory.

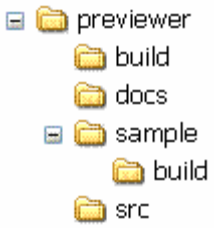

Figure 2 Previewer Driver Directory Structure

- **Build :** This folder contains the driver library.This library can be made dependent to projects for executing applications.
- **docs :** This folder contains API document, Design document ,Release notes document User Guide Document, Doxygen document and Sample Application Usage document.

  API Document contains the Driver details which can be helpful for the Application developers as to understand the driver interface.

  Design Document contains the Driver details which can be helpful for the developers as well as consumers to understand the driver designs.

  Release Note Document gives the details about system requirements, steps to Install /Uninstall the Package.This document list the known issues present in the driver.

  User Guide Document provides an information about how to use the driver .It provides sample applications which guide the end user in making their applications run along with this driver.

  Application usage note provides the information about how to run sample application and view the output.
- **sample :** This folder contains the tci file which in included in the tcf files.

  build : This subfolder in sample folder contains the read me file which specifies the location of sample appliaction.User can run specific sample application at the path provided in the text file.
- **src :**This folder contains driver source files.

2.2 Build

This section describes for each supported target environment, the applicable build options, supported configurations and how selected, the featured capabilities and how enabled, the allowed user customizations for the software to be installed and how the same can be realized.

The component might be delivered to user in different formats:

- Source-less ie., binary executables and object libraries only.
- Source–inclusive.,The entire source code, used to implement the driver is included in the delivered product.

Source-selective ie., Only a part of the overall source is included. This delivery mechanism might be required either because ;certain parts of the driver require soruce level extensions and/or customization at the user's end or because,specific parts of the driver is exposed to user at the source level to insure user's software development.

When source is included as part of the product delivery, the CCS project file is provided as part of the package. When object format is distributed, the driver header files are part of the "src" folder and the driver library is provided in /drivers/previewer/build folder.

2.3 Build Options

This driver does not have any specific build option at the time of writing of this manual.

The build folder contains a CCS project file that builds the driver into a library for debug and release mode.

Chapter 3

DSP/BIOS Previewer

This chapter describes the functions, data structures, enumerations and macros for the List module.

3.1 Functions

This section lists the functions available in the PSP module.

3.1.1 GIO_create

GIO_Handle GIO_create	**(String name, Int mode, Int* status, Ptr chanParams, GIO_Attrs* attrs)**

This function is called by the application to create the PREVIEWER channel. GIO_create () populates static settings in driver object, formally creates/registers driver entry point with DSP/BIOS,

Parameters:

name	[INOUT] Name of the device to open
mode	[INOUT] Mode in which device is to be opened
Status	[OUT] Address to which driver returns status
chanParams	[IN] optional
attrs	[IN] pointer to GIO_attrs structure

Returns:
GIO_Handle – if the operation is successful
NULL – if the operation is failed

Example:

```
GIO_Handle prevHandle;

prevHandle = GIO_create(deviceName, IOM_INOUT, NULL, &prevChannel,
    NULL);
```

deviceName is passed as Previewer. Instance is declared in the configuration file.

3.1.2 GIO_delete

Int GIO_delete	**(GIO_Handle giodChan)**

This function deletes the driver channel and undoes the memory allocation made by the GIO_create

Parameters:

gioChan	[IN] GIO Handle for the channel

Returns:
IOM_EBADARGS – If the parameters passed are not correct or the channel has never been created

Example:

```
GIO_Handle prevHandle;

GIO_delete(prevHandle);
```

3.1.3 GIO_control

PSP_Result GIO_control	**(GIO_Handle *gioChan*, Int *cmd*, Ptr *cmdArg*)**

This function handles the IOCTLs for the Previewer driver.

Parameters:

gioChan	[IN] GIO Handle for the channel
cmd	[IN] IOCTL Command
cmdArg	[INOUT] Argument for the IOCTL

Returns:
IOM_COMPLETED if successful or else suitable error code is given.
IOM_EBADARGS– *gioChan* is not valid or if the state is not appropriate or the argument passed is not appropriate
IOM_EBADMODE– if the cmdArg is not appropriate or if the fifo is not enabled or if the mode is not supported

Example:

```
GIO_Handle prevHandle;
PSP_previewerParams params;
params.sizeParam.sph = 1; /**< Start Pixel Horizontal */
params.sizeParam.eph = WIDTH; /**< End Pixel Horizontal */
params.sizeParam.slv = 1;          /**< Start Line Vertical */
params.sizeParam.elv = HEIGHT; /**< End Line Vertical */
params.sizeParam.pixelSize = PSP_PREVIEWER_INWIDTH_8BIT;
/**< Pixel size value can be PSP_PREVIEWER_INWIDTH_8BIT or
 PSP_PREVIEWER_INWIDTH_10BIT - used when source is SDRAM */
```

```
params.sizeParam.inPitch = WIDTH; /**< Line Offset of Input Image - used when source is SDRAM */
params.sizeParam.outPitch = (WIDTH*2);  /**< Line Offset of Output Image */

status = GIO_control(prevfd, PSP_PREVIEWER_IOCTL_SET_PARAMS, &params);
```

Note : PSP_PREVIEWER_IOCTL_SET_PARAMS like other IOCTL command names can be get from Section 3.3.

3.2 Data Structures Configuration defines

The file **psp_previewer.h** has the **PSP_previewerChannelSource** data structure that is used as a member element of PSP_previewerChannelCreate mode. The parameters of the structure are explained below:

1) Table Configuration Data Structure

Parameter	Description
source	Represents Input Source: value can be PSP_PREVIEWER_CHANNEL_CCDC (On-the-fly mode) or PSP_PREVIEWER_CHANNEL_SDRAM (One shot mode). When Source in CCDC Video Port Path Should be enable.

The file **psp_previewer.h** has the **PSP_previewerChannelCreateMode** is used at the time of channel creation. That will be passed to GIO_CREATE. The parameters of the structure are explained below:

2) Table Configuration Data Structure

Parameter	Description
source	channel source
segId	segment id passed by application, will be used to allocate memory

The file **psp_previewer.h** has the **PSP_previewerSize** data structure that used to configure Previewer size parameters. The instance of this structure will be part of PSP_previewerParams structure. The parameters of the structure are explained below:

3) Table Configuration Data Structure

Parameter	Description
inPitch	Represents line offset of input image - used when source is SDRAM .It should be 32 byte align
outPitch	Represents line offset of output image. It should be 32 byte align
sph	For specifying start pixel horizontal. If Source is CCDC then sph must be >=2
eph	For specifying end pixel horizontal. If Source is CCDC then sph must be >=2
slv	For specifying start line vertical
elv	For specifying end line vertical
pixelSize	pixel size of the image in terms of bits - used when source is SDRAM

The file **psp_previewer.h** has the **PSP_previewerWhiteBalance** data structure that is used for white balancing parameters. The instance of this structure will be part of PSP_previewerParams structure. The parameters of the structure are explained below:

4) Table Configuration Data Structure

Parameter	Description
position	Represents 16 position, out of 4 values
wbDgain	Represents white balance common(digital) gain
wbGain[PSP_PREVIEWER_WB_GAIN_MAX]	Individual color gains

The file **psp_previewer.h has** the **PSP_previewerBlackAdj** data structure that is used for black adjustment for colors. The instance of this structure will be part of PSP_previewerParams structure. The parameters of the structure are explained below:

5) Table Configuration Data Structure

Parameter	Description
redAdj	Represents black adjustment offset for red color
greenAdj	Represents black adjustment offset for green color
blueAdj	Represents black adjustment offset for blue color

The file **psp_previewer.h** has the **PSP_previewerRgbBlending** data structure that is used for for RBG2RGB blending. The instance of this structure will be part of PSP_previewerParams structure. The parameters of the structure are explained below:

6) Table Configuration Data Structure

Parameter	Description
blending[PSP_PREVIEWER_ RGB_MAX][PSP_PREVIEWE R_RGB_MAX]	Represents color correlation 3x3 matrix
offset[PSP_PREVIEWER_R GB_MAX]	Represents color correlation offsets

The file **psp_previewer.h has** the **PSP_previewerRgb2ycbcrCoeffs** data structure that is used for Rgb2Ycbcr parameters. The instance of this structure will be part of PSP_previewerParams structure. The parameters of the structure are explained below:

7) Table Configuration Data Structure

Parameter	Description
coeff[PSP_PREVIEWER_RG	Represents color conversion gains in 3x3 matrix

B_MAX][PSP_PREVIEWER_RGB_MAX];	
yOffset	Represents y color conversion offsets
cbOffset	Represents Cb color conversion offsets
crOffset	Represents Cr color conversion offsets

The file **psp_previewer.h has** the **PSP_previewerCfaCoeffs** data. The instance of this structure will be part of PSP_previewerParams structure. The parameters of the structure are explained below:

8) Table Configuration Data Structure

Parameter	Description
hThreshold	Represents horizontal threshold
vThreshold	Represents vertical threshold
coeffs[PSP_PREVIEWER_CFA_COEFF_TABLE_SIZE]	Represents cfa coefficients

The file **psp_previewer.h** has the **PSP_previewerGammaCoeffs** data. The instance of this structure will be part of PSP_previewerParams structure. The parameters of the structure are explained below:

9) Table Configuration Data Structure

Parameter	Description
red[PSP_PREVIEWER_GAMMA_TABLE_SIZE]	Represents table of gamma correction values for red
green[PSP_PREVIEWER_GAMMA_TABLE_SIZE]	Represents table of gamma correction values for green
blue[PSP_PREVIEWER_GAMMA_TABLE_SIZE]	Represents table of gamma correction values for blue

The file **psp_previewer.h** has the **PSP_previewerNoiseFilterCoeffs** data. The instance of this structure will be part of PSP_previewerParams structure. The parameters of the structure are explained below:

10) Table Configuration Data Structure

Parameter	Description
noise[PSP_PREVIEWER_NOISE_FILTER_TABLE_SIZE]	Represents noise filter table
strength	Represents to find out weighted average

The file **psp_previewer.h** has the **PSP_previewerChromaSuppression** data. The instance of this structure will be part of PSP_previewerParams structure. The parameters of the structure are explained below:

11) Table Configuration Data Structure

Parameter	Description
hpfy	Represents whether to use high passed version of Y or normal Y
threshold	Represents threshold for chroma suppress
gain	Represents chroma suppression gain

The file **psp_previewer.h** has the **PSP_previewerParams** data. That will be passed to GIO_CONTROL .when IOCTL is PSP_PREVIEWER_IOCTL_SET_PARAMS The parameters of the structure are explained below:

12) Table Configuration Data Structure

Parameter	Description
Features	Set of features enabled
sizeParam	size parameters
whiteBalanceParam	white balancing parameters
blackAdjParam	black adjustment parameters
rgbBlendingParam	rgb blending parameters
rgb2ycbcrParam	rgb to ycbcr parameters
cfaCoeffsParam	CFA coefficients
gammaCoeffsParam	gamma coefficients
noiseFilterCoeffsParam	noise filter coefficients
chromaSuppressionParam	chroma suppression coefficients
outPixelOrderParam	output pixel format
lumaEnhance[PSP_PREVIEWER_LUMA_TABLE_SIZE]	luma enhancement coeffs
darkFrameAddr	dark frame address
darkFrameOffset	Dark frame offset
hmfThreshold	hmfThreshold
contrast	constrast
brightness	brightness
downSampleRate	down sampling rate for averager
lensShadingShift	number of bits to be shifted for lens shading

The file **psp_previewer.h** has **PSP_previewerStatus** data. That will be passed to GIO_CONTROL when IOCTL is PSP_PREVIEWER_IOCTL_GET_STATUS,. The params are explained below:

13) Table Configuration Data Structure

Parameter	Description
channelStatus	value can be PSP_PREVIEWER_CHANNEL_BUSY or PSP_PREVIEWER_CHANNEL_FREE

The file **psp_previewer.h** has **PSP_previewerDarkFrameStatus** data. That will be passed to GIO_CONTROL when IOCTL is PSP_PREVIEWER_IOCTL_GET_DARK_FRAME_STATUS,.The parameters of the structure are explained below Table Configuration Data Structure

Parameter	Description
status	value can be PSP_PREVIEWER_DARK_FRAME_FAILED or PSP_PREVIEWER_DARK_FRAME_WORKING

The file **psp_previewer.h** has **PSP_preview** data. That will be passed to GIO_CONTROL when IOCTL is PSP_PREVIEWER_IOCTL_PREVIEW,.The parameters of the structure are explained below

14) Table Configuration Data Structure

Parameter	Description
inBuf	address of the input buffer. It should 32-byte aligned
outBuf	address of the output buffer It should 32-byte aligned
inBufSize	input buffer size
outBufSize	output buffer size

The file **psp_previewer.h** has the **PSP_previewerCropSize** data. That will be passed to GIO_CONTROL when IOCTL is PSP_PREVIEWER_IOCTL_GET_CROPSIZE,. The parameters of the structure are explained below:

15) Table Configuration Data Structure

Parameter	Description
hCrop	Represents number of pixels per line cropped in output image
vCrop	Represents number of lines cropped in output image

The file **psp_previewer.h** has the **PSP_previewerReadReqExp** data. Is used to pass a preview read request expand The parameters of the structure are explained below

16) Table Configuration Data Structure

Parameter	Description
prevExp	a preview read request expand

The file **psp_previewer.h** has the **PSP_previewerDarkFrameCapture** data. That will be passed to GIO_CONTROL when IOCTL is PSP_PREVIEWER_SET_DARK_FRAME_CAPTURE or PSP_PREVIEWER_GET_DARK_FRAME_CAPTURE,. The parameters of the structure are explained below:

17) Table Configuration Data Structure

Parameter	Description
darkFrameState	Indicates whether to enable dark frame capture or not. Value can be PSP_PREVIEWER_DARK_FRAME_CAPTURE_DISABLE or PSP_PREVIEWER_DARK_FRAME_CAPTURE_ENABLE
outPitch	offset in dark image for each row. Value is only relevant when flag is PSP_PREVIEWER_DARK_FRAME_CAPTURE_ENABLE

3.3 Symbolic Constants and Enumerated Data types

This section summarizes all the symbolic constants specified as either #define macros and/or enumerated C data types. Described alongside the macro or enumeration is the semantics or interpretation of the same in terms of what value it stands for and what it means.

It is typical to classify the data types into logical groups and list them in alphabetical order for ease of use.

Symbolic Constants and Enumerated Data Types

Group or Enumeration Class	Symbolic Constant Name	Description or Evaluation
Macro	PSP_PREVIEWER_MD_VERSION_1	Previewer mini driver version
Macro	PSP_PREVIEWER_MD_DEVPARAMS_DEFAULT	It describes the default parameter of the previewer mini driver.
Macro	PSP_PREVIEWER_INVERSE_ALAW	Enables support of Inverse A-Law. The inverse A-law block

		decompresses the 8-bit non-linear data to 10-bit linear data if enabled. This macro is used to assign value to features field of _ PSP_previewerParams
Macro	PSP_PREVIEWER_HMF	Enabled Support of Horizontal Median Filter The preview engine contains a horizontal median filter that is useful for reducing temperature induced noise effects. This macro is used to assign value to features field of _ PSP_previewerParams
Macro	PSP_PREVIEWER_NOISE_FILTER	Enabled Support of Noise Filter. This macro is used to assign value to features field of _ PSP_previewerParams
Macro	PSP_PREVIEWER_CFA	Enabled Support of CFA Interpolation Filter. This macro is used to assign value to features field of _ PSP_previewerParams
Macro	PSP_PREVIEWER_GAMMA	Enabled Support of Gamma Correction This macro is used to assign value to features field of _ PSP_previewerParams
Macro	PSP_PREVIEWER_LUMA_ENHANCE	Enabled Support of Luminance Enhance This macro is used to assign value to features field of _

		PSP_previewerParams
Macro	PSP_PREVIEWER_CHROMA_SUPPRESS	Enabled Support of Chrominance Suppression This macro is used to assign value to features field of _ PSP_previewerParams
Macro	PSP_PREVIEWER_DARK_FRAME_SUBTRACT	Enabled Support of Dark Frame Subtract. This macro is used to assign value to features field of _ PSP_previewerParams
Macro	PSP_PREVIEWER_LENS_SHADING	Enabled Support of Lens shading. This macro is used to assign value to features field of _ PSP_previewerParams
Macro	PSP_PREVIEWER_DOWN_SAMPLE_RATE1	Down Sample Rate will be 1. This macro is used to assign value to downSampleRate field of _ PSP_previewerParams
Macro	PSP_PREVIEWER_DOWN_SAMPLE_RATE2	Down Sample Rate will be 2. This macro is used to assign value to downSampleRate field of _ PSP_previewerParams
Macro	PSP_PREVIEWER_DOWN_SAMPLE_RATE4	Down Sample Rate will be 4. This macro is used to assign value to downSampleRate field of _ PSP_previewerParams
Macro	PSP_PREVIEWER_DOWN_SAMPLE_RATE8	Down Sample Rate will be 8. This macro is used to assign value to

		downSampleRate field of _ PSP_previewerParams
Macro	PSP_PREVIEWER_LUMA_TABLE_SIZE	Size of the Luminance Enhancement table. Its value is 128. This macro is used to assign value to lumaEnhance field of _ PSP_previewerParams
Macro	PSP_PREVIEWER_GAMMA_TABLE_SIZE	Size of the Gamma Correction Coefficient's table. Its value is 1024. This macro is used to assign value to gammaCoeffsParam field of _ PSP_previewerParams
Macro	PSP_PREVIEWER_CFA_COEFF_TABLE_SIZE	Size of the CFA Interpolation Coefficient's table. Its value is 576. This macro is used to assign value to cfaCoeffsParam field of _ PSP_previewerParams
Macro	PSP_PREVIEWER_NOISE_FILTER_TABLE_SIZE	Size of the Noise Filter Coefficients table. Its value is 256. This macro is used to assign value to noiseFilterCoeffsParam field of PSP_previewerParams
Macro	PSP_PREVIEWER_WB_GAIN_MAX	Describe the number of coefficient for the white balace gain that 4 This macro is used to assign value to position field of PSP_previewerWhiteBalance
Macro	PSP_PREVIEWER_RGB_MAX	Describes the matrix size of the RGB2RGB Blending that is 3. This macro is used to assign value to blending field of PSP_previewerRgbBlending

Macro	PSP_PREVIEWER_MAX_IMAGE_WIDTH	Maximum image width supported by the driver. Its value is 1280.
Macro	PSP_PREVIEWER_CHANNEL_CCDC	Enabling Source is CCDC
Macro	PSP_PREVIEWER_CHANNEL_SDRAM	Enabling Source is SDRAM This macro is used to assign value to source field of PSP_previewerChannelSource
Macro	PSP_PREVIEWER_INWIDTH_8BIT	Indicates that the input image's pixel width is 8 bits. This macro is used to assign value to pixelSize field of PSP_previewerSize
Macro	PSP_PREVIEWER_INWIDTH_10BIT	Indicates that the input image's pixel width is 10 bits. This macro is used to assign value to pixelSize field of PSP_previewerSize
Macro	PSP_PREVIEWER_CHANNEL_FREE	Indicates previewer channel free This macro is used to assign value to channelStatus field of PSP_previewerStatus
Macro	PSP_PREVIEWER_CHANNEL_BUSY	Indicates previewer channel busy. This macro is used to assign value to channelStatus field of PSP_previewerStatus
Macro	PSP_PREVIEWER_DARK_FRAME_FAILED	Indicates Dark Frame Subtract Failed. This macro is used to assign value to status field of PSP_previewerDarkFrameStatus
Macro	PSP_PREVIEWER_DARK_FRAME_WORKING	Indicates Dark Frame Subtract working.

		This macro is used to assign value to status field of PSP_previewerDarkFrameStatus
enum _PSP_previewerControlCmd	PSP_PREVIEWER_IOCTL_SET_PARAMS	to set PSP_previewerParams parameters. Control Command to set Previewer channel configuration parameters. This command will not write to hardware it will just store the channel configuration. _PSP_previewerParams structure is passed in GIO_control for this control command. This IOCTL can be called only when previewer is disable.
enum _PSP_previewerControlCmd	PSP_PREVIEWER_IOCTL_GET_PARAMS	to get PSP_previewerParams parameters. PSP_previewerParams structure is passed in GIO_control for this control command;
enum _PSP_previewerControlCmd	PSP_PREVIEWER_IOCTL_GET_STATUS	to get the status of channel. PSP_previewerStatus structure is passed in GIO_control for this control command;
enum _PSP_previewerControlCmd	PSP_PREVIEWER_IOCTL_GET_DARK_FRAME_STATUS	to get dark frame subtract fail status PSP_previewerDarkFrameStatus structure is passed in GIO_control for this control command
enum _PSP_previewerControlCmd	PSP_PREVIEWER_IOCTL_PREVIEW	to trigger previewer when source is SDRAM. PSP_preview structure is passed in GIO_control for this control command
enum _PSP_previewerContro	PSP_PREVIEWER_IOCTL_GET_CROPSIZE	to get crop size. PSP_previewerCropSize

lCmd		structure is passed in GIO_control for this control command
enum _PSP_previewerControlCmd	PSP_PREVIEWER_IOCTL_SET_EXP	to set read request expand. PSP_previewerReadReqExp structure is passed in GIO_control for this control command
enum _PSP_previewerControlCmd	PSP_PREVIEWER_GET_INFO_FOR_CCDC	To get the information required for CCDC continuous mode PSP_previewerGetInfoForCCDC structure is passed in GIO_control for this control command
enum _PSP_previewerControlCmd	PSP_PREVIEWER_SET_DARK_FRAME_CAPTURE	To enable/disable dark frame capture feature PSP_previewerDarkFrameCapture structure is passed in GIO_control for this control command
enum _PSP_previewerControlCmd	PSP_PREVIEWER_GET_DARK_FRAME_CAPTURE	To get current dark frame capture feature state PSP_previewerDarkFrameCapture structure is passed in GIO_control for this control command
enum _PSP_previewerOutPixelOrder	PSP_PREVIEWER_PIXELORDER_YCBYCR	Represents output pixel order Values are: LSB Y0 Cb0 Y1 Cr0 MSB
enum _PSP_previewerOutPixelOrder	PSP_PREVIEWER_PIXELORDER_YCRYCB	Represents output pixel order Values are LSB Y0 Cr0 Y1 Cb0 MSB
enum _PSP_previewerOutPixelOrder	PSP_PREVIEWER_PIXELORDER_CBYCRY	Represents output pixel order Values are LSB Cb0 Y0 Cr0 Y1 MSB
enum _PSP_previewerOutPixelOrder	PSP_PREVIEWER_PIXELORDER_CRYCBY	Represents output pixel order Values are LSB Cr0 Y0 Cb0 Y1 MSB

Chapter 4

EXAMPLE APPLICATIONS

This section describes the example applications that are included in the package. These sample applications can be run as is for quick demonstration, but the user will benefit most by using these samples as reference source code in developing new applications.

4.1 The Previewer Sample Application

4.1.1 Introduction

psp_bios_prev_st_basic_example.pjt: This application demonstrates basic previewing functionality. This application converts raw image in Bayer pattern to YUV4:2:2 image.

4.1.2 Building the Application

The sample application is located in the \pspdrivers\system\DM6437\bios\dm6437_evm\src\video\sample\previewer\ folder. The sample can be built directly from its project file using Code Composer studio.

4.1.3 Loading the Application

The sample application is loaded and executed via Code composed studio. It is recomended to reset the board before loading Code Composer

4.2 The Previewer Multipass Application

4.2.1 Introduction

psp_bios_prev_st_multipass_example.pjt: This application demonstrates multipass functionality to preview image of horizontal size greater than 1280 pixels. This application converts raw image in Bayer pattern to YUV4:2:2 image.

4.2.2 Building the Application

The sample application is located in the \pspdrivers\system\DM6437\bios\dm6437_evm\src\video\sample\previewer\ folder. The sample can be built directly from its project file using Code Composer studio.

4.2.3 Loading the Application

The sample application is loaded and executed via Code composed studio. It is recommended to reset the board before loading Code Composer.

4.3 The Previewer On-The-Fly Sample Application

4.3.1 Introduction

psp_bios_prev_st_on_the_fly_example.pjt: This application demonstrates previewer on the fly functionality. It is loopback application and captured image will be displayed on TV.

4.3.2 Building the Application

The sample application is located in the \pspdrivers\system\DM6437\bios\dm6437_evm\src\video\sample\previewer_on_the_fly\ folder. The sample can be built directly from its project file using Code Composer studio.

PSP_VIDEO_PATH_ENABLE macro must be defined in psp_vpfe.h.

4.3.3 Loading the Application

The sample application is loaded and executed via Code composed studio. It is recomended to reset the board before loading Code Composer.

4.4 The Previewer On-The-Fly Dark Frame Capture Sample Application

4.4.1 Introduction

psp_bios_prev_st_on_the_fly_dfc_example.pjt: This application demonstrates previewer dark frame capture functionality in respect to on-the-fly mode. It is loopback application and captured image will be displayed on TV.

4.4.2 Building the Application

The sample application is located in the \pspdrivers\system\DM6437\bios\dm6437_evm\src\video\sample\previewer_on_the_fly\ folder. The sample can be built directly from its project file using Code Composer studio.

PSP_VIDEO_PATH_ENABLE macro must be defined in psp_vpfe.h.

4.4.3 Loading the Application

The sample application is loaded and executed via Code composed studio. It is recomended to reset the board before loading Code Composer.