

ADV8005 evaluation board External OSD Software Demo User Manual

Preliminary 0.1

August 2015

Revision History

Revision	Data	Remarks
0.1	Aug,2015	Initial version

Table of Contents

External OSD Software Demo User Manual.....	1
1.0 Introduction.....	5
1.1 Purpose	5
1.2 Scope.....	5
1.3 Limitation	5
1.3.1 Video Formats.....	5
1.3.2 Color Space on Analog Outputs.....	5
1.3.3 Main Video in 4K Resolution	5
2.0 Hardware Overview.....	6
2.1 OSD TTL.....	6
2.2 HDMI RX	6
2.3 Main TTL.....	7
3.0 Software Configuration	7
3.1 Input Selection	8
3.2 External OSD Source.....	8
3.2.1 Main Video Input Selection	8
3.2.2 Secondary Video Input Selection.....	9
3.3 Data Format	9
3.3.1 ADV800X.....	9
3.4 Color Space.....	9
3.4.1 ADV800X.....	9
3.4.2 SPI Configuration	9
3.5 Clock.....	10
3.6 Saving External OSD to Memory	10
4.0 Display Modes	11
4.1 Main Video with OSD Menu	11
4.2 OSD Menu Only	11
4.3 External OSD Only	11
4.4 Menu over External OSD.....	12
4.5 PiP with Menu.....	12
4.6 PoP with Menu.....	13
5.0 API List	14
5.1 Default Settings.....	14
5.1.1 Initialize.....	14
5.1.2 Buffer Size.....	14
5.2 User Control.....	15
5.2.1 Initialization.....	15
5.2.2 Set HDMI Port for External OSD.....	15
5.2.3 Get External OSD HDMI Port.....	15
5.2.4 Set External OSD VIC	16
5.2.5 Get External OSD VIC.....	16
5.2.6 Get External OSD Source.....	16

5.3 Display Modes	17
5.3.1 Default Mode.....	17
5.3.2 Menu Only	17
5.3.3 External OSD Only.....	17
5.3.4 Menu on External OSD	18
5.3.5 PiP	18
5.3.6 PoP.....	19
5.3.7 Setting Display Mode.....	19
5.4 External OSD Driver Configuration	20
5.4.1 Set Source.....	20
5.4.2 Set Format.....	20
5.4.3 Set Color Space.....	21
5.4.4 Get Color Space	21
5.4.5 Set Size.....	21
5.4.6 Set Location	22
5.4.7 Set Crop Location.....	22
5.4.8 Turn External OSD ON/OFF.....	23
5.4.9 Turn Main Menu ON/OFF	23
5.4.10 Set Main Menu Location	23
5.4.11 Set Main Menu Location.....	24
5.4.12 Get Main Menu Location.....	24
Annex A. Console Commands for External OSD	25
A.1 Initialization	25
A.2 Control	26
A.3 Display Mode.....	26
A.3.1 pipmode.....	26
A.3.2 when to run pipmode.....	26
A.3.3 Test results	27
Appendix A. BLIMP for External OSD	28

1.0 Introduction

VSP (ADV8003 and ADV8005) provides several choices for inputs, like main TTL, HDMI RX, OSD TTL. These inputs can work at the same time. For instance, main video is from main TTL, while 2nd video from OSD TTL or HDMI RX.

The 2nd video input is usually used for OSD, hereafter we take the 2nd input as external OSD. When the 2nd video is displayed over the main video, we get the result of Picture In Picture (PiP).

1.1 Purpose

This document describes hardware connection, VSP registers and API functions for VSP to control external OSD.

1.2 Scope

This document is intended to be used in conjunction with the below documents referenced at <https://ez.analog.com/docs/DOC-10635>

- Blimp_User_Manual
- Blimp_Framework_User_Manual

1.3 Limitation

When using main video and 2nd video together, here are some limitations on current release.

1.3.1 Video Formats

When to have pip, it is not recommended for main video and 2nd video to be interlaced format (480i, 576i, 1080i)

1.3.2 Color Space on Analog Outputs

All development is on HDMI TX1 and TX2. Color space conversion for analog outputs (Component, CVBS/S-video) is not implemented in current release.

1.3.3 Main Video in 4K Resolution

When main video is 4Kx2K resolution, here are some limitations:

Mode	Output		
	4K resolution	non-4K resolution	Note
OSD menu only	OK (1)	not available (2)	(1) Output 4K is using the same resolution of input 4K. (2) SVSP is used for OSD only mode, it cannot do 4K downscaling at the same time.
2nd video only	OK	OK	VSP is working on 2nd video, it does not matter if main video is 4K or not.
2nd video + menu	OK	OK	
PiP	OK (3)	not available (4)	(3) Output 4K is using the same resolution of input 4K. (4) SVSP is used to save 2nd video to OSD buffer, it cannot do 4K downscaling at the same time.
PoP	Not available(5)	OK	(5) POP cannot support 4K main video. Because SVSP is required for 4K downscaling, while

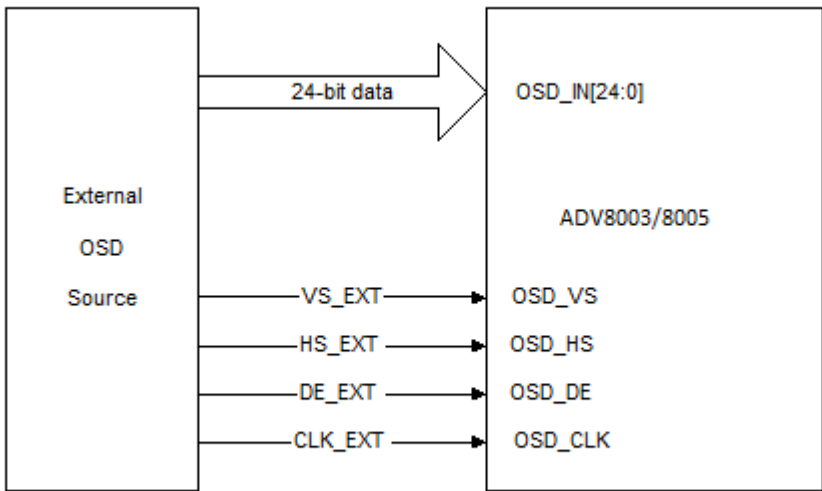
			SVSP is used for 2nd video in POP. There is no resource to support 4K.
--	--	--	--

2.0 Hardware Overview

There are two VSP platforms, EVAL-ADV8003 EBZ for 8003 and EVAL-ADV8005 EBZ for 8005. EVAL-ADV800 and ADV8005 EBZ platform has all inputs, main TTL, HDMI Rx and OSD TTL. HDMI Rx and main TTL for external OSD are developed on this platform.

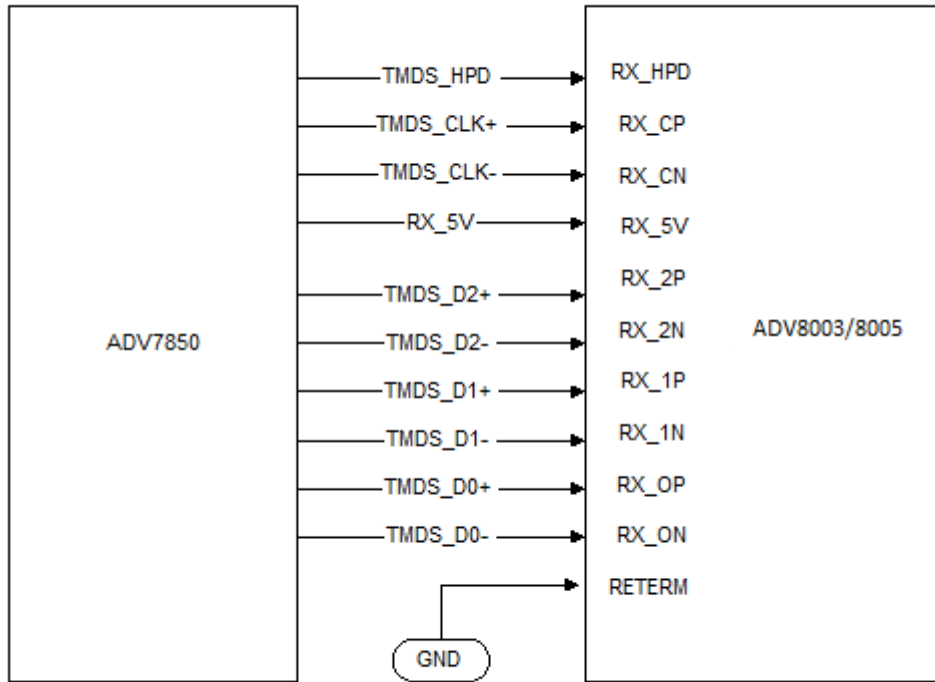
2.1 OSD TTL

Here are the signals connections on using OSD TTL as the source for external OSD.



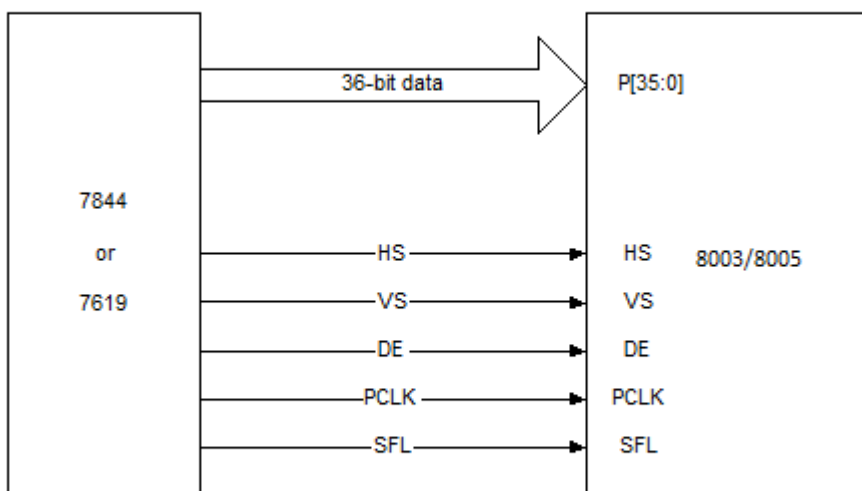
2.2 HDMI RX

HDMI RX connection is using TMDS interface for external OSD.



2.3 Main TTL

Main TTL takes 36-bit data bus for external OSD.



3.0 Software Configuration

There are some major ADV8003/8005 registers to control the external OSD input, format, color space, etc.

3.1 Input Selection

Each module, OSD, or VSP, or TX, or Encoder has its own registers to select input channel.

Register	Bits	Function	Values and Description	Meaning
[0X1A05]	Bits3:0	PVSP_INP_SEL[3:0]	0x00 From Primary Input Channel 0x01 From Internal OSD Blend 1 0x02 From Secondary Input Channel 0x03 From RX input) 0x04 From Secondary VSP 0x05 From Horizontal Prescaler	This signal is used to select the video Source for the Main VSP
	Bits7:4	SVSP_INP_SEL[3:0]	0x00 From Primary Input Channel 0x01 From Internal OSD Blend 1 0x02 From Primary VSP 0x03 From Internal OSD(OSD only, No Blend) 0x04 From Secondary Input Channel 0x05 From RX Input 0x06 From Horizontal Prescaler	This Signal is used to select the video Source for the Secondary VSP

ADV8003 and ADV8005 take OSD TTL as a normal input as main TTL or HDMI RX. Each module (OSD, VSP, Encoder and TX) could select main TTL or HDMI RX or OSD TTL freely.

3.2 External OSD Source

ADV800X (ADV8003 and ADV8005) use different registers to select the hardware inputs for external OSD.

Be noted that section "Input Selection" provides the capability for each module to use OSD TTL channel, they are controlled by 8003/8005 IIC registers [0x1A02]..[0x1A06]. The OSD TTL channel is a logic map of hardware pins controlled by "External OSD Source".

The External OSD Source must be connected to either main TTL input pins, or the OSD TTL input pins, or TMDS receiver input pins using s_inp_sel[1:0] ([0x1A07] bit3:2).

External OSD Source controls the action like saving to DDR memory. These actions are run in BLIMP and OSD driver.

3.2.1 Main Video Input Selection

Main Video Source is selected by p_inp_sel, defined at [0x1A07] bit1:0 in ADV8003 and ADV8005

p_inp_sel = 00: TTL Video (from main TTL)
 = 01: TTL OSD Video (from OSD TTL)
 = 10: 48 bit TTL input for 3GHz Interleaved TTL
 = 11: reserved

3.2.2 Secondary Video Input Selection

External OSD source is selected by s_inp_sel, defined at [0x1A07] bit3:2 in ADV8003 and ADV8005

- s_inp_sel = 00: TTL Video (from main TTL)
- = 01: TTL OSD Video (from OSD TTL)
- = 10: Rx Video (from HDMI RX)
- = 11: reserved

3.3 Data Format

Correct data format on the data bus must be configured for external OSD

3.3.1 ADV800X

Different formats could be configured in ADV8003 or ADV8005 for different external OSD source. Default settings are:

extOSD Source	Format Register	Default Value	Meaning
main TTL	[1B48]	0x08	vid_swap_bus_ctrl = 0: D[35:24] D[23:12] D[11:0] vid_format_sel = 8: 3x12 bit 4:4:4
OSD TTL	[1B68]	0x0C	extosd_swap_bus_ctrl = 0: D[23:16] D[15:8] D[7:0] extosd_format_sel = 0x0C: 3x8 bit 4:4:4
HDMI RX	[1B88]	0x08	rx_swap_bus_ctrl = 0: D[35:24] D[23:12] D[11:0] rx_format_sel = 8: 3x12 bit 4:4:4

3.4 Color Space

Both RGB and YCbCr 4:4:4 input are supported by external OSD. A color space converter (CSC) is required when input is YCbCr.

3.4.1 ADV800X

Each input, main TTL, HDMI RX, and OSD TTL, has its own IIC registers for color space converter in ADV8003 and ADV8005.

extOSD Source	Color Register	Enable	Default Setting
main TTL	[1B30]..[1B47]	[1B30] bit7	vid_csc_enabled = 0: csc disabled
OSD TTL	[1B50]..[1B67]	[1B50] bit7	extosd_csc_enabled = 0: csc disabled
HDMI RX	[1B70]..[1B87]	[1B70] bit7	rx_csc_enabled = 0: csc disabled

IIC CSC registers are usually used when the input is for main video.

When using 7850 or 7619 to receive the external OSD, 7850 or 7619 auto Color Space Conversion is enabled, and RGB is the default output. CSC in 8003 is not required.

3.4.2 SPI Configuration

There is an OSD color space converter by SPI interface:

ADIAPI_OsdCsAdjust (BOOL CsRgb, UCHAR Range)

Usually this SPI command is used to adjust OSD menu color space.

3.5 Clock

ADV8003 and ADV8005 have the same registers to select clock for external OSD.

Register	Bits	Definition	Bits Value	Meaning	Register Value
[0x1A15]	bit2	vsp_3d_in_clk_sel_man_en	1	Manual mode selection for PVSP clk in	0x02
	bit3	vsp_2d_in_clk_sel_man_en	1	Manual mode selection for SVSP clk in	0x08
[0x1A19]	bit7:4	vsp_2d_in_clk_sel_main[3:0]	1001	Clock from External OSD to SVSP	0x90
[0x1A24]	bit4	osd_hs_vs_mode	1	Use external HS/VS/DE Syncs	0x10

3.6 Saving External OSD to Memory

By default, external OSD is saved to OSD memory at 480P (720x480@60), taking part of internal OSD memory.

When external OSD is other resolution than 480P, external OSD could goto SVSP at first for downscaling or upscaling to 480P. SVSP has a feature to save its output to the OSD memory (taking part of internal OSD memory). This is done by bit svsp_osd_mode_en, it is [0xE611] bit2.

ADV8003 gives definition:

svsp_osd_mode_en = 1: VIM works in EXOSD mode

When [0xE611]=0x04, hardware controls by not saving external OSD to memory, and saving SVSP output to memory.

Resolutions other than 480P could be used to save external OSD, refer to Annex B.1 for configurations.

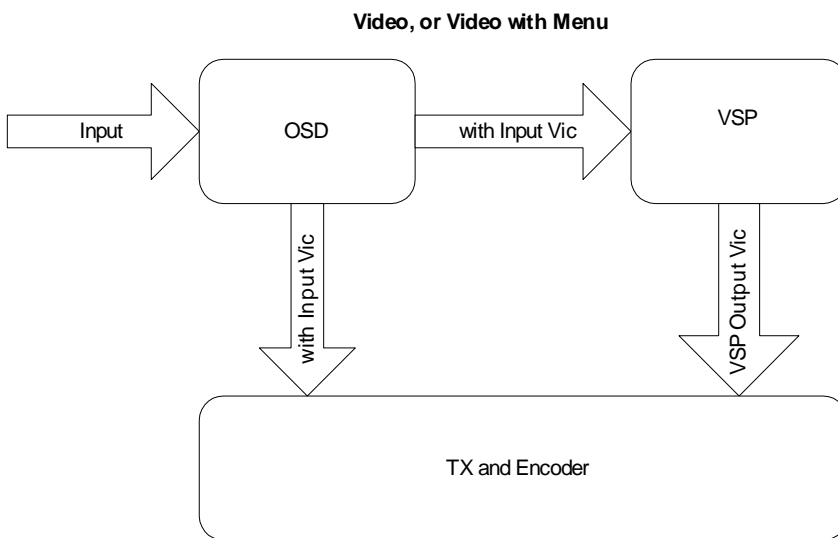
4.0 Display Modes

When getting two video inputs at the same time, there are several display mode:

1. Main video or main video with OSD menu (default mode)
2. OSD menu only
3. 2nd video only (external OSD only)
4. 2nd video with menu (external OSD and OSD menu, no main video)
5. Main video, 2nd video and OSD menu (all inputs), 2nd video inside main video (PiP)
6. Main video, 2nd video and OSD menu (all inputs), 2nd video outside main video (PoP)

4.1 Main Video with OSD Menu

This is the default display mode. Video goes like:

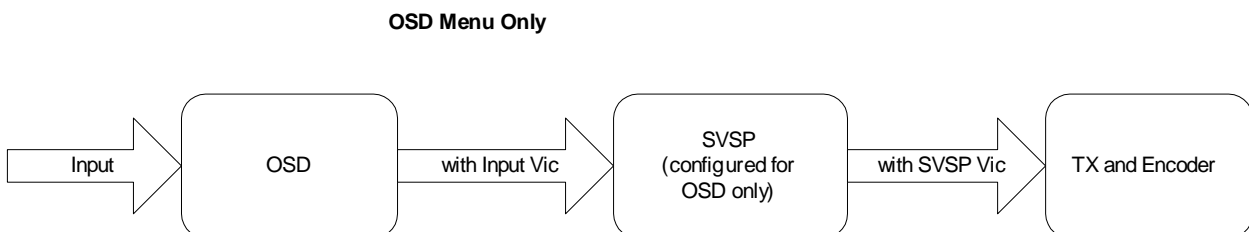


- Note: 1) OSD timing is on main video.
2) External OSD is not supported in this mode.

4.2 OSD Menu Only

SVSP has a feature to get OSD only, it is done by:

`vsp2d_inp_sel[3:0] = 0011` ([0x1A05] bit7:4 = 0011) for SVSP from OSD (OSD only, no blend).

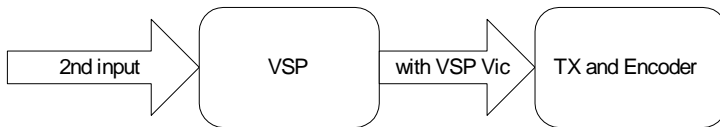


Note: VSP generate the timing 1080P for external OSD. The main video could be any format or free run.

4.3 External OSD Only

Suppose external OSD comes from the 2nd input, it could be handled like the main input.

external OSD only

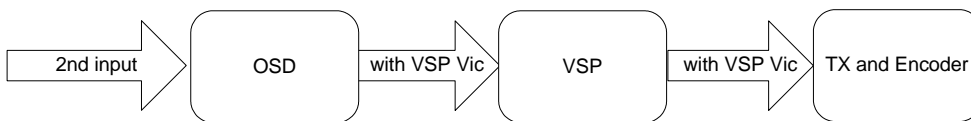


Note: Main input and OSD menu are hidden in this mode, no need to set OSD resolution

4.4 Menu over External OSD

When to display menu over extOSD, extOSD should be used as video input.

OSD menu over External OSD



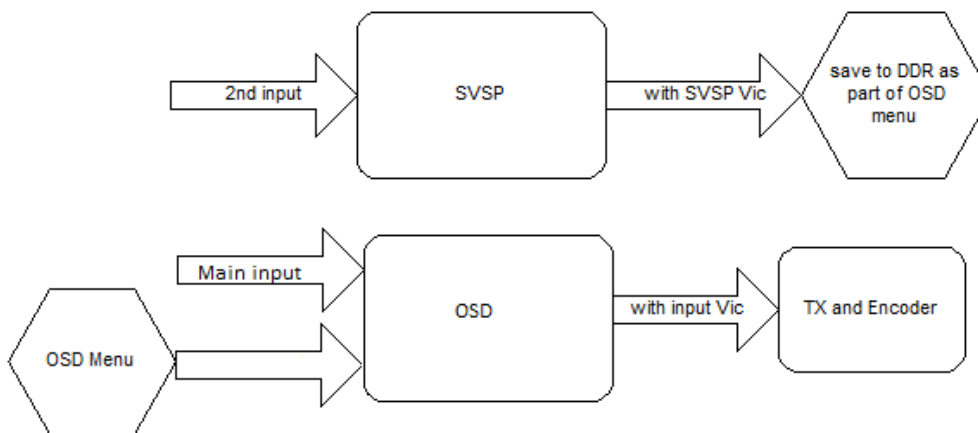
Note: 1) Main input is not used in this mode.

2) OSD timing is depending on 2nd video

4.5 PiP with Menu

It is possible to display main input, 2nd input and OSD menu together.

Picture in Picture



Note: OSD timing is on main video input

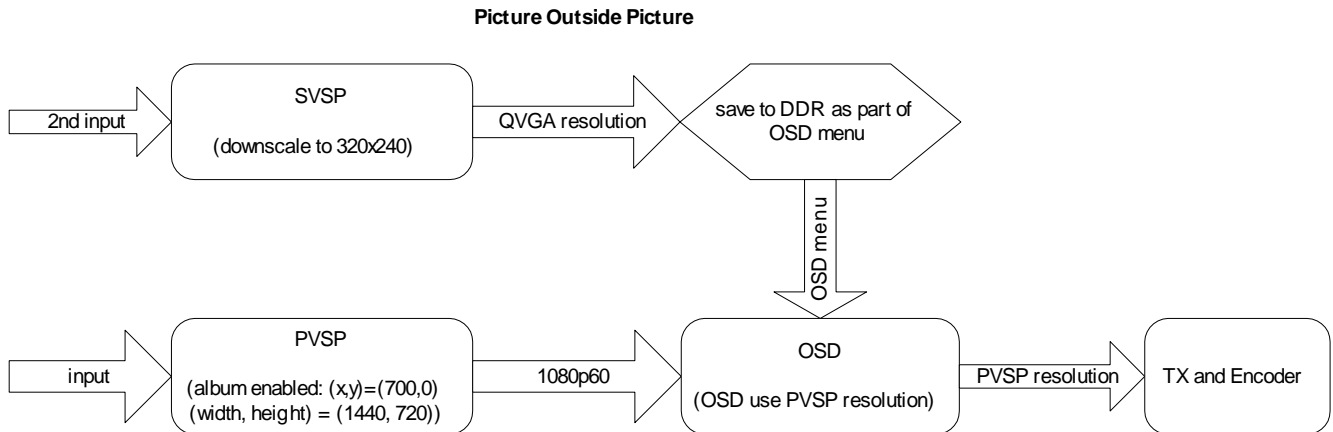
Main and 2nd Input generates PiP. OSD menu is placed over the PiP.

The 2nd input and OSD menu could be turned on/off separately to get combinations:

- * Menu and ExtOSD over Main Video
- * Menu over Main Video
- * ExtOSD over Main Video

4.6 PoP with Menu

It is possible to display the 2nd input outside the main input with OSD menu together. Example below is using 1080P for main picture (PVSP) resolution and QVGA for small picture (SVSP) resolution.



Note: OSD timing is on PVSP output

Here SVSP, PVSP and OSD module are configured by using fixed resolution for demo purpose:

- SVSP output QVGA (320x240) for 2nd video not to take much space in the screen;
- QVGA is for small picture at bottom-left corner;
- As OSD takes PVSP as source, OSD should use PVSP output VIC to set its resolution;
- PVSP album configuration set the main video at the top-right corner

All these configuration could be adjusted to get the expected PoP result.

The 2nd input and OSD menu could be turned on/off separately to get combinations:

- * Menu over Main Video, and ExtOSD outside Main Video
- * Menu over Main Video
- * ExtOSD outside Main Video

Be noted that PoP demo is based on 60Hz frame rate. PVSP is configured to generate 60Hz output, while 2nd input must be 60Hz. It is possible to support 50Hz PoP, it need some software work on preparing SVSP timing.

5.0 API List

This section lists API functions for external OSD.

5.1 Default Settings

5.1.1 Initialize

Description

External OSD is defined as one page of OSD menu by BLIMP, besides normal menu initialization, like size and location, external OSD need setup on its source, data format, and color space.

Synopsis

```
void Initialize(POSD_EXTERNALOSD page, HObj hObj)
```

Parameters

page: structure pointer to the page for external OSD
hObj: object handle pointer for external OSD

Return Value

None

Remarks

Called from menu initialization function OSDEG_customizeOSD(void)

5.1.2 Buffer Size

Description

720x576x4 buffer size is allocated to external OSD at bootup. It means that external OSD is saved at resolution of 576P at most as a page to the OSD menu.

Buffer size could be enlarged to 720P or 1080P, it will take more memory, increasing the hardware cost.

Synopsis

```
ATV_ERR HAL_SecondaryVSPSetFrameBuffer(UINT32 FrameBufferSize)
```

Parameters

FrameBufferSize: buffer size allocated to external OSD, used by SVSP to save its output to memory

Return Value

ATVERR_OK

Remarks

Called from SVSP initialization function HAL_SecondaryVSPInitialize(void).

5.2 User Control

5.2.1 Initialization

Description

To initialize external OSD based on hardware connections.

Synopsis

```
ATV_ERR APP_PipInit (void)
```

Parameters

None

Return Value

ATVERR_OK

5.2.2 Set HDMI Port for External OSD

Description

To select HDMI port for external OSD

Synopsis

```
void APP_PipSet2ndVidInputPort(UCHAR Hdmiport)
```

Parameters

Hdmiport = 0:	HDMI port A
= 1:	HDMI port B
= 2:	HDMI port C
= 3:	HDMI port D

Return Value

None

Remarks

pipvic should be run to set VIC for selected port if different resolutions are used in HDMI ports.

5.2.3 Get External OSD HDMI Port

Description

To read back the HDMI Port external OSD.

Synopsis

```
ATV_ERR APP_PipGet2ndVidInputPort(UCHAR* Hdmiport)
```

Parameters

Hdmiport: pointer to save the HDMI port number of external OSD.

Return Value

ATVERR_OK

5.2.4 Set External OSD VIC**Description**

To set the VIC for external OSD

Synopsis

```
ATV_ERR APP_PipSet2ndVidVic(UINT16 ExtVic)
```

Parameters

ExtVic: VIC of external OSD

Return Value

ATVERR_OK

Remarks

Software can not automatically detect the external OSD VIC (or resolution). The VIC must be manually set. A correct VIC is required for VSP to work properly.

5.2.5 Get External OSD VIC**Description**

To get the VIC of external OSD

Synopsis

```
ATV_ERR APP_PipGet2ndVidVic(UINT16 *ExtVic)
```

Parameters

*ExtVic: pointer to address to save external OSD VIC

Return Value

ATVERR_OK

5.2.6 Get External OSD Source**Description**

to read back the source of external OSD.

Synopsis

```
ATV_ERR APP_PipGet2ndVidSrc(EXT_OSD_SOURCE *src)
```

Parameters

src: pointer to save the source of external OSD.

Return Value

ATVERR_OK

5.3 Display Modes

5.3.1 Default Mode

Description

By default, video and OSD menu are enabled, external OSD is disabled.

Synopsis

```
ATV_ERR APP_PiPMenuMainVideo(UCHAR MainVidVic)
```

Parameters

MainVidVic: main video VIC

Return Value

ATVERR_OK

Remarks

This mode is called when to switch among different display modes, or when main video resolution is changed.

5.3.2 Menu Only

Description

This mode displays OSD menu only, no main video, no external OSD.

Synopsis

```
ATV_ERR APP_PiPMenuOnly(UCHAR MainVidVic, UCHAR MenuVic)
```

Parameters

MainVidVic: main video VIC, used for the SVSP input resolution

MenuVic: SVSP output VIC, it is for the display resolution.

Return Value

ATVERR_OK

Remarks

Only SVSP could support the feature of OSD only.

5.3.3 External OSD Only

Description

This mode displays External OSD video only, no main video, no OSD menu

Synopsis

```
ATV_ERR APP_PiP2ndVideoOnly(UCHAR ExtOsdVic, UCHAR PvspVic, UCHAR SvspVic)
```

Parameters

ExtOsdVic: external OSD VIC, used for the VSP input resolution
PvspVic: PVSP output VIC
SvspVic: SVSP output VIC

Return Value

ATVERR_OK

Remarks

This function uses PVSP and SVSP at the same time for demo purpose.

5.3.4 Menu on External OSD**Description**

This mode displays OSD menu over the external OSD video, external OSD video is in full screen. There is no main video.

Synopsis

ATV_ERR APP_PiPMenu2ndVideo(UCHAR ExtOsdVic, UCHAR SpicVic)

Parameters

ExtOsdVic: external OSD VIC, used for the VSP input resolution
SpicVic: VSP output VIC, used for the display resolution

Return Value

ATVERR_OK

Remarks

This function uses PVSP or SVSP.

5.3.5 PiP**Description**

This mode displays OSD menu and external OSD video over the main video, main video is in full screen.

Synopsis

ATV_ERR APP_PiPMenuMain2ndVideo(UCHAR ExtOsdVic, UCHAR SpicVic, UCHAR OsdVic)

Parameters

ExtOsdVic: external OSD VIC, used for the SVSP input resolution
SpicVic: SVSP output VIC for external OSD, 480P or 576P because of buffer limitation
OsdVic: main video VIC, used for OSD resolution

Return Value

ATVERR_OK

Remarks

PiP(Picture in Picture) is external-OSD-video in main-video, OSD menu is over main-video.

5.3.6 PoP

Description

This mode displays OSD menu over main video, and external OSD video outside the main video. Main video is at top-right corner, external OSD at bottom-left corner, OSD menu at the left side.

Synopsis

```
ATV_ERR APP_PoPMenuMain2ndVideo(EXT_OSD_POP PoPConfig)
```

Parameters

```
typedef struct
{
    UINT8 VideoVic;           /* main video VIC */
    UINT8 PVspVic;           /* Primary VSP output VIC */
    UINT8 ExtOsdVic;         /* external OSD VIC */
    UINT8 SVspVic;           /* Secondary VSP output VIC */
    UINT8 OsdVic;            /* OSD menu VIC */
    UINT16 VideoX;           /* Primary VSP album: (x, y, width, height) */
    UINT16 VideoY;
    UINT16 VideoWidth;
    UINT16 VideoHeight;
    UINT16 ExtOsdX;          /* external OSD location (x,y) */
    UINT16 ExtOsdY;
    UINT16 ExtOsdWidth;     /* external OSD display size (width, height) */
    UINT16 ExtOsdHeight;
} EXT_OSD_POP;
```

Return Value

```
ATVERR_OK
```

Remarks

60Hz frame rate is used in this demo.

5.3.7 Setting Display Mode

Description

This function is the entry to set display modes, which are described in 5.3.1 ~ 5.3.6.

Synopsis

```
ATV_ERR ATV_ERR APP_ExtOsdDisplayMode (UCHAR Mode)
```

Parameters

Mode: display mode,
1 = default

2 = menu only
3 = extOSD only
4 = extOSD+menu
5 = PiP
6 = PoP

Return Value

ATVERR_OK

5.4 External OSD Driver Configuration

This section is for APIs in application layer used to control or to provide information to the external OSD driver.

External OSD driver is in code:

OSD_BMP\ES2\source\vsp_osdeg_extosd.c

Refer to OSD firmware manual for detail information.

5.4.1 Set Source

Description

To select the source for external OSD, main TTL, or OSD TTL, or HDMI Rx

Synopsis

```
void APP_ExtOSDSetSource(EXT_OSD_SOURCE src)
```

Parameters

src: external OSD source, main TTL, or OSD TTL, or HDMI Rx

Return Value

None

5.4.2 Set Format

Description

To set the data format of external OSD.

Usually main TTL and HDMI Rx provides 3x12-bit 4:4:4 at D[35:24]D[23:12]D[11:0],

OSD TTL provides 3x8-bit 4:4:4 at D[23:16]D[15:8]D[7:0],

Synopsis

```
void APP_APP_ExtOSDSetFormat (UINT8 fmt)
```

Parameters

fmt: external OSD data format

Return Value

None

5.4.3 Set Color Space

Description

To set the color space, RGB or YCbCr 4:4:4, of external OSD.

Synopsis

```
void APP_ExtOSDSetColorSpace(UINT8 cs)
```

Parameters

cs: external OSD color space, 0 for RGB, 1 for YCbCr 4:4:4

Return Value

None

5.4.4 Get Color Space

Description

To get the color space, RGB or YCbCr 4:4:4, of external OSD.

Synopsis

```
void APP_ExtOSDGetColorSpace (UINT8* cs)
```

Parameters

cs: address pointer to save external OSD color space

Return Value

None

5.4.5 Set Size

Description

To set the size (or display area) of external OSD

Synopsis

```
void APP_ExtOSDSetDisplayArea(UINT16 Osdx, UINT16 Osdy)
```

Parameters

Osdx: width

Osdy: height

Return Value

None

Remarks

As external OSD default buffer size is 576P, the size cannot go over Osdx=720 and Osdy=576. When (Osdx, Osdy) is less than 720x576, only part of the external OSD is being displayed. There is no downscaling or upscaling.

As normal OSD menu is initialized to 720P resolution by commands:

```
OSD_setOSDSize (1280, 720);
```

```
OSD_setVideoResolution (1280, 720, 0, 0, 0);
```

480P/576P external OSD occupies part of OSD menu (720P).

When external OSD need higher resolution than 720P, OSD menu resolution should be increased at the same time.

5.4.6 Set Location

Description

To set the external OSD display location.

Synopsis

```
void APP_ExtOSDSetLocation(UINT16 Osdx, UINT16 Osdy)
```

Parameters

Osdx: place in horizontal

Osdy: place in vertical

Return Value

None

Remarks

The upper-left corner is (0, 0).

5.4.7 Set Crop Location

Description

To set the external OSD crop location to display.

Synopsis

```
void APP_ExtOSDSetDisplayAreaLoc(UINT16 Osdx, UINT16 Osdy)
```

Parameters

Osdx: place to crop in horizontal

Osdy: place to crop in vertical

Return Value

None

Remarks

The upper-left corner is (0, 0).

5.4.8 Turn External OSD ON/OFF

Description

To display or hide the external OSD.

Synopsis

```
void APP_ShowExternalOSD(BOOL show)
```

Parameters

show = TRUE: to display external OSD
= FALSE: to hide external OSD

Return Value

None

Remarks

External OSD could be hidden when to display the OSD menu.

5.4.9 Turn Main Menu ON/OFF

Description

To display or hide the main OSD menu.

Synopsis

```
void APP_ShowMainPage(BOOL show)
```

Parameters

show = TRUE: to display OSD menu
= FALSE: to hide OSD menu

Return Value

None

Remarks

It is used to hide main menu when to see external OSD only.

5.4.10 Set Main Menu Location

Description

To set the main video display location.

Synopsis

```
void APP_ExtOSDSetMainVideoLocation(UINT16 mainx, UINT16 mainy)
```

Parameters

(mainx, mainy): (x,y) for PVSP album feature

Return Value

None

Remarks

This function is for PiP only. PiP uses album feature to display main video.

5.4.11 Set Main Menu Location**Description**

To set the display mode by OSD menu

Synopsis

```
void APP_ExtOSDSetMode(UINT8 pip_mode)
```

Parameters

pip_mode: PiP or PoP or external OSD only

Return Value

None

Remarks

There are total 6 display modes, OSD menu supports only 3 of them.

5.4.12 Get Main Menu Location**Description**

To get the display mode for OSD menu to use

Synopsis

```
void void APP_ExtOSDGetMode(UINT8* pip_mode)
```

Parameters

pip_mode: address pointer to save mode

Return Value

None

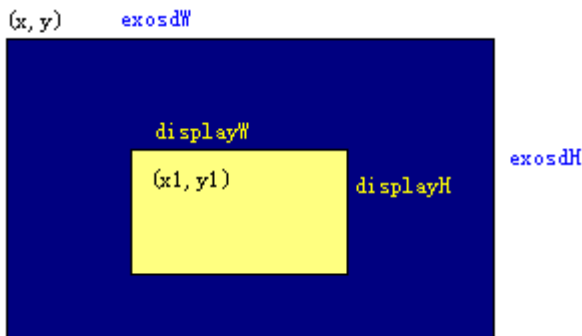
Annex A. Console Commands for External OSD

Here are all console commands to control external OSD.

pipinit	Initializes the source of external OSD and set the default data format and color space
pipdisp	Enable/Disable small picture size <on, off>
pipsize	Configure small picture size <width> <height>
piploc	Configure small picture location <x> <y>
pipcs	Configure small picture Color Space
pipfmt	Configure small picture Data Format
pipvic	manually set VIC of small picture
pipmode	Configure video display mode <index>

A.1 Initialization

External OSD components are initialized at bootup.



Assume blue area is external OSD video full size, yellow area defined the pickup area to display.

(x, y) is the position that external OSD component will be displayed. $(exosdW, exosdH)$ is the external OSD maximum size. We use this value to allocate DDR2 buffer.

$(x1, y1)$ is the offset from (x, y) , $(displayW, displayH)$ is the area to be picked up and displayed.

Initialization code uses 480P by default for external OSD:

```
void Initialize(POSD_EXTERNALOSD page, HOBJ hObj)
{
    page->externalOSD1 = OSDEG_createObj(CTRL_EXTOSD, "externalOSD1", 1, 0, 0, 720, 480,
(HOBJ)hObj, NULL); //exosdW=720, exosdH=480, cannot change by user command
    EXTOSD_FUNC->suspendLayout((HOBJ)page->externalOSD1);
    EXTOSD_FUNC->setZ((HOBJ)page->externalOSD1, 10);
    EXTOSD_FUNC->setLocationX((HOBJ)page->externalOSD1, 10); //x=10, set by "piploc"
    EXTOSD_FUNC->setLocationY((HOBJ)page->externalOSD1, 10); //y=10, set by "piploc"
    EXTOSD_FUNC->setEnabled((HOBJ)page->externalOSD1, TRUE);
    EXTOSD_FUNC->setSource((HOBJ)page->externalOSD1, EXTOSD_SRC_OSD_TTL); //
EXTOSD_SRC_OSD_TTL=0, set by "pipinit"
    EXTOSD_FUNC->setDisplayArea((HOBJ)page->externalOSD1, 0, 0, 500, 300); // x1=0, y1=0,
displayW=500, displayH=300; set by "pipsize"
```

```

EXTOSD_FUNC->setFormat((HOBJ)page->externalOSD1, 12); //24bit SDR 444 for OSD TTL, set by
“pipfmt”
EXTOSD_FUNC->setColorSpace((HOBJ)page->externalOSD1, 0); // RGB, set by “pipcs”
EXTOSD_FUNC->resumeLayout((HOBJ)page->externalOSD1);
Load(page);
_OSDEG_init_ready = TRUE;
}

```

When using QVGA (320x240@60Hz) as default external OSD size, the configuration will be like:

```

void Initialize(POSD_EXTERNALOSD page, HOBJ hObj)
{
    page->externalOSD1 = OSDEG_createObj(CTRL_EXTOSD, "externalOSD1", 1, 0, 0, 320, 240,
(HOBJ)hObj, NULL); //exosdW=320, exosdH=240 for external OSD size in OSD memory
    .....
    EXTOSD_FUNC->setDisplayArea((HOBJ)page->externalOSD1, 0, 0, 320, 240); // x1=0, y1=0,
displayW=320, displayH=240; set by “pipsize”
    .....
}

```

A.2 Control

There are several commands to control the external OSD mode. Please refer the document BF_800X_system commands document available at <https://ez.analog.com/docs/DOC-11128> for external OSD commands added.

A.3 Display Mode

When there are two inputs (main video and external OSD) with OSD menu, there are several combinations on display modes.

A.3.1 pipmode

“pipmode” set the display mode, as

- “pipmode 1” for video and OSD menu, it is the default mode
- “pipmode 2” for OSD menu only
- “pipmode 3” for external OSD only
- “pipmode 4” for OSD menu on external OSD, no main video
- “pipmode 5” for OSD menu and external OSD over main video
- “pipmode 6” for OSD menu and external OSD outside main video

A.3.2 when to run pipmode

Be sure to run “pipmode 1” at first when to switch among each mode. For instance, to switch from mode 2 to mode 5, commands are run as:

```

“pipmode 2”
“pipmode 1”
“pipmode 5”

```

When main video resolution is changed, “pipmode 1” should be run again.

A.3.3 Test results

“pipmode” are tested on resolution 480i/576i/480P/576P/720P/1080i/1080P for main video and external OSD, Here is the table for all combinations and the result. Each combination is tested on “pipmode 1/2/3/4/5”.

main video 2nd video	480i 60	480p 60	720p 60	1080i 60	1080p 60	576i 50	576p 50	720p 50	1080i 50	1080p 50	command (2)
480i60	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	pipvic 6
480p60	ok(1)	ok	ok	ok(1)	ok	ok(1)	ok	ok	ok(1)	ok	pipvic 2
720p60	ok(1)	ok	ok	ok(1)	ok	ok(1)	ok	ok	ok(1)	ok	pipvic 4
1080i60	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	pipvic 5
1080p60	ok(1)	ok	ok	ok(1)	ok	ok(1)	ok	ok	ok(1)	ok	pipvic 16
576i50	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	pipvic 21
576p50	ok(1)	ok	ok	ok(1)	ok	ok(1)	ok	ok	ok(1)	ok	pipvic 17
720p50	ok(1)	ok	ok	ok(1)	ok	ok(1)	ok	ok	ok(1)	ok	pipvic 19
1080i50	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	ok(1)	pipvic 20
1080p50	ok(1)	ok	ok	ok(1)	ok	ok(1)	ok	ok	ok(1)	ok	pipvic 31

Note:

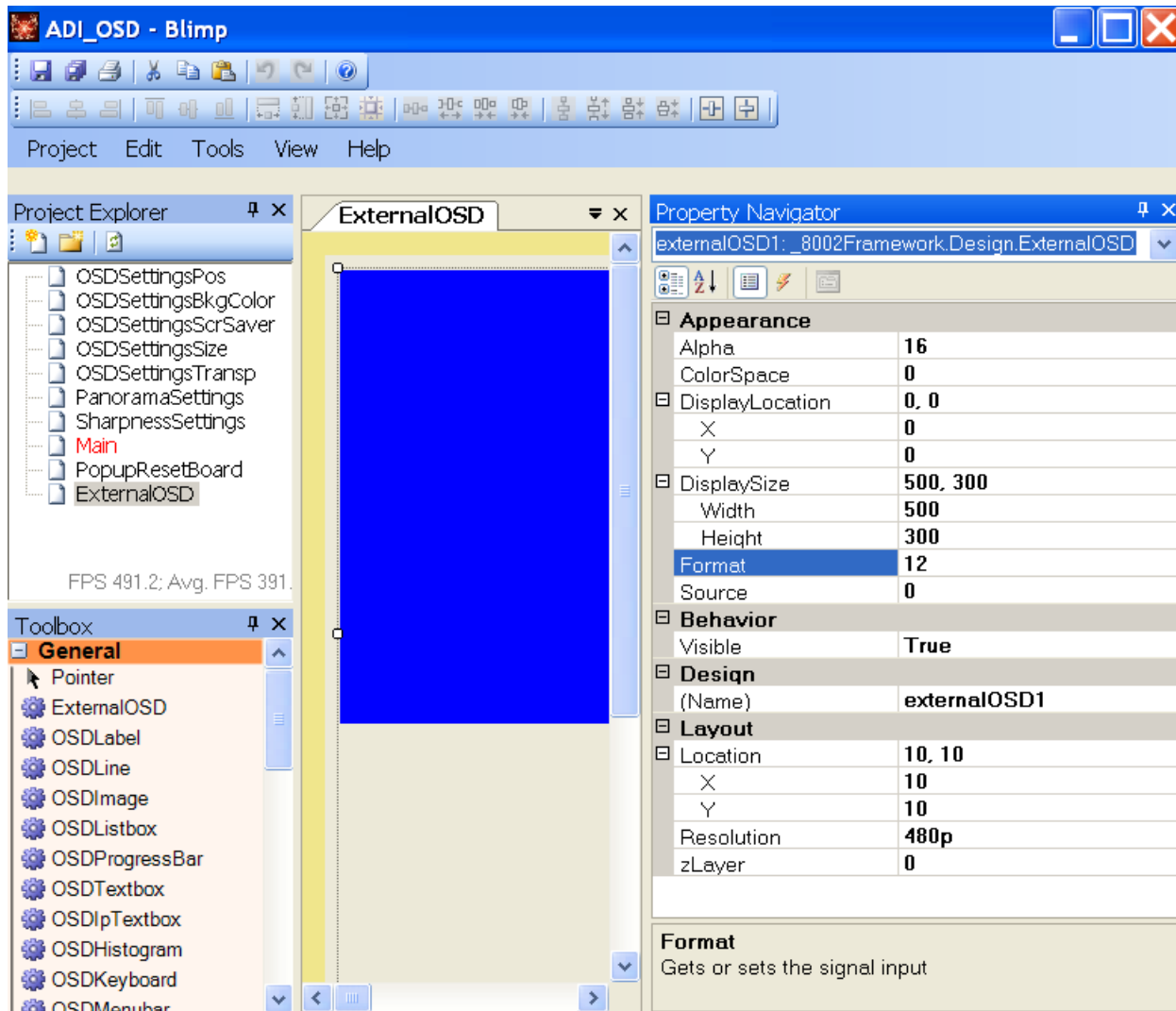
- 1) SVSP cannot support interlaced input. When main video or 2nd video is interlaced, some of pipmode are not displayed properly; they are split in the upper half screen.
- 2) “pipvic” must be run at first to set the right VIC for the 2nd video before running “pipmode”.

According to pipmode 6:

- 1) main video goes through PVSP to get 1080P60 (or any resolution), main video could be any formats except 1080p100/120 or formats whose pixel clock is over 162MHz, which is the VSP limitation;
- 2) external OSD goes through SVSP which can not take interlace input, SVSP could convert 480p60, 576p50, 720p120/100/60/50/30/25/24, 1080p60/50/30/25/24 to QVGA properly.

Appendix A. BLIMP for External OSD

Here is the BLIMP setup for external OSD configuration.



These Blimp settings are generated automatically to files:

APP\8002_AVR\OSD\ADI_DEMO\blimp_externalosd.c and blimp_externalosd.h