

LSP 1.20 DaVinci Linux Video Sysfs

User's Guide



Literature Number: SPRUEL6
March 2008

1	Overview	5
2	Supported Encoders	6
	2.1 VPBE Encoder.....	6
	2.2 Logic PD Encoder.....	6
	2.3 THS8200 Encoder.....	6
3	Boot Arguments/Module Parameters for Encoder Manager	6
4	Example Code	7
	4.1 From The Console.....	7
	4.2 From a C Application.....	8

List of Tables

1	VPBE Encoder: Supported Output and Mode Names	6
2	Logic PD Encoder: Supported Output and Mode Names	6
3	THS8200 Encoder: Supported Output and Mode Names	6

LSP 1.20 DaVinci Linux Video Sysfs

This guide introduces the DaVinci Linux Video Sysfs interface. For LSP 1.20, the Sysfs interface is supported on the following EVMs: DM644x, DM355.

1 Overview

In the DaVinci Video Driver, Sysfs driver attributes are used for configuring the *encoder* in the driver to work for a specific display. The *Encoder Manager* is shared by both the FBDev and V4L2 drivers. Encoders register with the manager and implement a set of API calls to configure the display hardware for a specific output and mode. In addition to this, they also provide control functions like enable/disable display, control brightness, hue, and contrast. Sysfs attributes are defined to allow you to control these operations from your applications.

The following are the Sysfs attributes defined in the DaVinci Video Driver for DM355 and DM6446:

- *Output*

`/sys/class/davinci_display/ch0/output`

You use this attribute to select a display output. Each of the encoder modules defines a set of outputs; applications can use this attribute to set the output to one of the supported outputs. At any time only one encoder is active in the driver. This encoder is selected when an output that is defined by the encoder is selected as the current output of the driver.

- *Mode*

`/sys/class/davinci_display/ch0/mode`

You use this attribute to select a display mode. Each of the encoder modules defines a set of modes; the application can use this attribute to set the mode to one of the supported modes. When an encoder is selected as the current encoder (selected based on the current output), application can set the current display mode to one of the supported modes of the current encoder.

- *Enable*

`/sys/class/davinci_display/ch0/enable`

You use this attribute to enable or disable the display. Write “on” to this file to enable the display and “off” to disable the display. Use the output names and mode names listed in next section for writing to output and mode attributes.

In addition the following Sysfs attributes are defined for fine tuning the system. You may have to fine tune these values for the end system being developed.

- *reg*

`/sys/class/davinci_system/system/reg`

You use this for fine tuning the PBBPR register in EMIF. This is a system level knob you can adjust if noise appears on the displayed image.

- *vpbe_osd_basepx*

`/sys/class/davinci_system/system/vpbe_osd_basepx`

You use this for adjusting the OSD BASEPX value. This would need to be adjusted if the video displayed is not properly aligned to the display area.

- *vpbe_osd_basepy*

`/sys/class/davinci_system/system/vpbe_osd_basepy`

You use this for adjusting the OSD BASEPY value. This would need to be adjusted if the video displayed is not properly aligned to the display area.

2 Supported Encoders

2.1 VPBE Encoder

This encoder is used for displaying Standard Definition (SD) video outputs and modes using built-in DAC and VENC in the VPBE. The following table lists the supported output and mode names that can be used for writing into the Sysfs attribute when using this encoder.

Table 1. VPBE Encoder: Supported Output and Mode Names

Output Names	Mode Names	Supported Platforms
COMPOSITE	NTSC, PAL	DM355 and DM6446
SVIDEO	NTSC, PAL	DM6446 only
COMPONENT	NTSC, PAL, 480P-60, 576P-50	DM6446 only

2.2 Logic PD Encoder

This encoder is used for displaying to a Logic PD LCD monitor. The following table lists the supported output and mode names that can be used for writing into the Sysfs attribute when using this encoder.

Table 2. Logic PD Encoder: Supported Output and Mode Names

Output Names	Mode Names	Supported Platforms
LCD	640 x 480 640 x 400 640 x 350	DM355 and DM6446

Note: The Enable attribute is not implemented for the Logic PD Encoder.

2.3 THS8200 Encoder

This encoder is used for displaying High Definition (HD) video outputs and modes using the THS8200 daughter card. The component output connector available on the THS8200 daughter card is used for this display. The following table lists the supported output and mode names that can be used for writing into the Sysfs attribute when using this encoder.

Table 3. THS8200 Encoder: Supported Output and Mode Names

Output names	Mode names	Supported platform/s
COMPONENT1	720P-60 1080I-30	DM355 and DM6446

3 Boot Arguments/Module Parameters for Encoder Manager

The desired output and mode selected by the Encoder Manager at initialization time can also be controlled using boot arguments. The following boot arguments are used for setting the default output and mode at boot up.

The Encoder Manager will start up using these defaults:

```
davinci_enc_mgr.ch0_output=<output_name>
davinci_enc_mgr.ch0_mode=<mode_name>
```

If the Encoder Manager is built as a module, then the following applies:

```
>insmod davinci_enc_mgr.ko ch0_output=<output_name>  
ch0_mode=<mode_name>  
>rmmmod davinci_enc_mgr.ko
```

where `output_name` and `mode_name` are given in [Section 2](#).

4 Example Code

This section shows how to set/get Sysfs attributes from the console as well as from a C application.

4.1 From The Console

From the console, you can set an attribute using the echo command. For example to set output to LCD, type the following command:

```
>echo "LCD" >/sys/class/davinci_display/ch0/output
```

To display the current output, you can use the cat command as follows:

```
>cat /sys/class/davinci_display/ch0/output
```

To disable display from the console, you can enter the command

```
echo "off" >/sys/class/davinci_display/ch0/enable
```

To enable display from the console, you can enter the command

```
echo "on" >/sys/class/davinci_display/ch0/enable
```

To write a value into the PBBPR register in EMIF, you can enter the command

```
echo "20000020 30" >/sys/class/davinci_system/system/reg
```

To read a value from the PBBPR register you can enter the command

```
echo "20000020" >/sys/class/davinci_system/system/reg
```

To write a value into the OSD BASEPX and BASEPY registers you can do the following:

```
echo "120" >/sys/class/davinci_system/system/vpbe_osd_basepx  
echo "120" >/sys/class/davinci_system/system/vpbe_osd_basepy
```

To read the above values you can do the following:

```
cat >/sys/class/davinci_system/system/vpbe_osd_basepx  
cat >/sys/class/davinci_system/system/vpbe_osd_basepy
```

4.2 From a C Application

The following sample code shows how to set the output and mode of the driver from a C application.

```
int file_fd;
file_fd = open("/sys/class/davinci_display/ch0/output",O_RDWR);
if (file_fd < 0)
{
    perror("Error in opening /sys/class/davinci_display/ch0/output\n");
    exit(1);
}
if (write(file_fd,"COMPOSITE",10) <= 0)
{
    perror("Error switching output to COMPOSITE\n");
    exit(1);
}
lseek(file_fd,0,SEEK_SET);

bzero(output_str,15);
if (read(file_fd,output_str,10) < 0 )
{
    perror("Error reading output\n");
    exit(1);}
printf("output : %s\n",output_str);
if (!strcmp(output_str,"COMPOSITE\n"))
{
    printf("Successfully changed output to %s\n",output_str);
}
else
{
    printf("Failed to switch output to Composite\n");
    exit(1);
}
if (close(file_fd) <0)
{
    perror("error closing \n");
    exit(1);
}
file_fd = open("/sys/class/davinci_display/ch0/mode",O_RDWR);
if (file_fd < 0)
{
    perror("Error in opening /sys/class/davinci_display/ch0/mode\n");
    exit(1);
}
if (write(file_fd,"NTSC",5) <= 0)
{
    perror("Error switching mode to NTSC\n");
    exit(1);
}
lseek(file_fd,0,SEEK_SET);

bzero(output_str,15);
if (read(file_fd,output_str,5) < 0 )
{
    perror("Error reading mode\n");
    exit(1);
}
printf("mode: %s\n",output_str);
if (!strcmp(output_str,"NTSC\n"))
    printf("Successfully changed mode to %s\n",output_str);
else
{
    printf("Failed to switch mode to NTSC\n");
    exit(1);
}
if (close(file_fd) <0)
{
    perror("error closing \n");
    exit(1);
}
```

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 TI 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. Information of third parties may be subject to additional restrictions.

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.

TI products are not authorized for use in safety-critical applications (such as life support) where a failure of the TI 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. Buyers represent that they have all necessary expertise in the safety and regulatory ramifications of their applications, and acknowledge and agree that they are solely responsible for all legal, regulatory and safety-related requirements concerning their products and any use of TI products in such safety-critical applications, notwithstanding any applications-related information or support that may be provided by TI. Further, Buyers must fully indemnify TI and its representatives against any damages arising out of the use of TI products in such safety-critical applications.

TI products are neither designed nor intended for use in military/aerospace applications or environments unless the TI products are specifically designated by TI as military-grade or "enhanced plastic." Only products designated by TI as military-grade meet military specifications. Buyers acknowledge and agree that any such use of TI products which TI has not designated as military-grade is solely at the Buyer's risk, and that they are solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI products are neither designed nor intended for use in automotive applications or environments unless the specific TI products are designated by TI as compliant with ISO/TS 16949 requirements. Buyers acknowledge and agree that, if they use any non-designated products in automotive applications, TI will not be responsible for any failure to meet such requirements.

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

Products

Amplifiers	amplifier.ti.com
Data Converters	dataconverter.ti.com
DSP	dsp.ti.com
Clocks and Timers	www.ti.com/clocks
Interface	interface.ti.com
Logic	logic.ti.com
Power Mgmt	power.ti.com
Microcontrollers	microcontroller.ti.com
RFID	www.ti-rfid.com
RF/IF and ZigBee® Solutions	www.ti.com/lprf

Applications

Audio	www.ti.com/audio
Automotive	www.ti.com/automotive
Broadband	www.ti.com/broadband
Digital Control	www.ti.com/digitalcontrol
Medical	www.ti.com/medical
Military	www.ti.com/military
Optical Networking	www.ti.com/opticalnetwork
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
Copyright © 2008, Texas Instruments Incorporated