TI81XX PSP VIDEO CAPTURE Driver User Guide

TI81xx AM38xx PSP Video Capture Driver User Guide

Hardik Shah

Video Capture Driver

Read This First

This section show the major updates since last release, please review this section before any SW activities.

- V4L2 capture is not supported for TI813X(TI385 called in other places)family.
- The build procedures are changed as below:

```
$make ARCH=arm CROSS_COMPILE=PATH_TO_TOOLCHAIN/bin/arm-arago-linux-gnueabi- ti8168_evm_defconfig
   $make ARCH=arm CROSS_COMPILE=PATH_TO_TOOLCHAIN/bin/arm-arago-linux-gnueabi- uImage
   $make ARCH=arm CROSS_COMPILE=PATH_TO_TOOLCHAIN/bin/arm-arago-linux-gnueabi- modules
```

- Bootargs parameter is required to use notify driver in kernel. This address is valid for TI816X/AM389X platforms.

```
$ notifyk.vpssm3_sva=0xBF900000
```

- Do not use 0xA0000000.
- M3 firmware loader program is changed from procmgrapp to slaveloader

```
$ ./slaveloader startup VPSS-M3 ti816x_hdvpss.xem3
```

- TI816x/AM389x platforms share the same M3 firmware ti816x_hdvpss.xem3
- TI814x/AM387x platforms share the same M3 firmware ti814x_hdvpss_1G_debug.xem3.
- V4L2 capture nodes are /dev/video0, /dev/video4, /dev/video5 and /dev/video6.
- Above nodes are created only when ti81xxvo.ko is loaded followed by ti81xxvin.ko
- V4L2 capture streaming is supported only on /dev/video0 node with TVP7002 as decoder.

Introduction

Video Processing Sub-System hardware integrates theee graphics pipeline, five video pipelines, two capture ports and 4 video compositors. Video compositors are connected to four VENCs to support various output format and mode.

The primary functionality of the VPSS driver is to provide interfaces to user level applications and management of Video Processing Sub-System hardware. Interfaces and features of the fbdev and v4l2 display driver are covered in video driver user guide. This document covers the software architecture, interfaces and features provided by the V4L2 capture driver.

References

1. Video for Linux Two Home Page [http://linux.bytesex.org/v4l2/ [1]]
2. Video for Linux Two API Specification [[http://v4l2spec.bytesex.org/v4l2spec/v4l2.pdf [2]]

Acronyms & Definitions

Acronym	Definition
V4L2	Video for Linux Two
VPSS	Vidoe Processing SubSystem
HDMI	Hight Definition Multimedia Interface
NTSC	National Television System Committee
PAL	Phase Alternating Line
SD	Standard Definition
SDK	Software Development Kit
VIP	Video Input Port

Hardware Overview

The video processing subsystem provides the functions to display a video frame from the memory frame buffer to external display device, such as LCD,TV; or capture a video frame from external camera to memory. The video processing subsystem integrates the following main elements

- Graphics processing modules
- Video Capture processing modules
- Video display processing modules
- Video Compositor modules

Software overview

V4L2 capture driver provides the application interface for the video capture processing modules of the VPSS. It allows the external decoders to be interfaced to VIP capture port of the VPSS through standard V4L2 sub device driver model. Current driver supports VIP 0 instance with TVP7002 decoder on I/O expansion cards of TI814x and TI816x. It is possible to support other decoders on VIP 0 instance by adding the sub-device driver for required decoder and passing the board specific data through board file. Please refer to the application note on how to add different decoders to the V4L2 capture driver.

Installation Guide

This is the TI81XX Linux VPSS(VPSS capture) Driver user installation guide. This documents the steps to get the VPSS Driver worked up on TI81XX based EVM(Simulator is not tested). This driver has the following dependencies outside of Kernel.

- TI81XX PSP 04.04.00.02 [3] release for TI814x, TI837x, TI816x, TI389x, TI385x and TI813x
- Syslink 2.20.00.14 [4] release (slaveloader)
- HDVPSS 01.00.01.44 [5] release

Note: Only the above mentioned version of the components are verified and tested.

Preliminary Work

- Build the Linux Uboot and Kernel following the installation guide of TI816x PSP 04.04.00.02 release based on platform
- M3 BIOS VPSS Firmware(ti816x_hdvpss.xem3/ti814x_hdvpss_1G.xem3) is loaded by the slaveloader, which is the user space program from syslink and is part of PSP release package.

Build V4L2 capture driver

Though build dependency over syslink has been removed, VPSS driver still relies on the slaveloader to load M3 firmware, V4L2 Capture Driver supports dynamic build only. The pre-build uImage inside of PSP release package supports the pre-built V4L2 capture module.

NOTE:

1. Applications are not required to rebuild the uImage or V4L2 capture module unless V4L2 drivers are changed.

```
$ make ARCH=arm CROSS_COMPILE=PATH_TO_TOOLCHAIN/bin/arm-arago-linux-gnueabi- ti8168_evm_defconfig
$ make ARCH=arm CROSS_COMPILE=PATH_TO_TOOLCHAIN/bin/arm-arago-linux-gnueabi- uImage
$ make ARCH=arm CROSS_COMPILE=PATH_TO_TOOLCHAIN/bin/arm-arago-linux-gnueabi- menuconfig
```

- Enable V4L2 Capture Driver in Menuconfig

Select Device Drivers from the main menu.

```
...
    ...
    Kernel Features  --->
    Boot options  --->
    CPU Power Management  --->
    Floating point emulation  --->
    Userspace binary formats  --->
    Power management options  --->
[*] Networking support  --->
"   Device Drivers  --->"
    ...
    ...
```

Select Multimedia support from the menu.

```
    ...
    ...
    Sonics Silicon Backplane  --->
    Multifunction device drivers  --->
[*] Voltage and Current Regulator Support  --->
"<*> Multimedia support  --->"
    Graphics support  --->
<*> Sound card support  --->
[*] HID Devices  --->
[*] USB support  --->
    ...
    ...
```

Select Video For Linux from the menu.

```
    ...
    ...
    *** Multimedia core support ***
[ ]   Media Controller API (EXPERIMENTAL)
"<*>   Video For Linux"
[*]     Enable Video For Linux API 1 (DEPRECATED)
< >   DVB for Linux
    ...
    ...
```

Select Video capture adapters from the same menu. Press <ENTER> to enter the corresponding sub-menu.

```
    ...
    ...
[ ]   Customize analog and hybrid tuner modules to build  --->
"[*]   Video capture adapters  --->"
[ ]   Memory-to-memory multimedia devices  --->
[ ]   Radio Adapters  --->
    ...
    ...
```

Select TI81XX V4L2-Capture driver

```
[ ]   Autoselect pertinent encoders/decoders and other helper chi
      Encoders/decoders and other helper chips  --->
<>   TI81XX V4L2-Capture driver
"<M>   TI81XX V4L2-Capture driver"
< >   CPiA2 Video For Linux
< >   Philips SAA7134 support
```

- Build module for the drivers

```
$ make ARCH=arm CROSS_COMPILE=PATH_TO_TOOLCHAIN/bin/arm-arago-linux-gnueabi- modules
```

- ti81xxvin.ko(VPSS V4L2 capture driver) is generated under drivers/media/video/ti81xx

Prerequisites

V4L2 capture driver requires following modules to be loaded for its functioning.

1. syslink kernel driver module, referred as "syslink.ko"
2. loader user space program, referred as "slaveloader"
3. M3 BIOS Firmware binary, referred as "ti816x_hdvpss.xem3/ti814x_hdvpss_1G.xem3"
4. VPSS kernel driver module, referred as "vpss.ko"
5. TVP7002 decoder kernel module referred to as "tvp7002.ko"
6. V4L2 capture kernel module, referred as "ti81xxvin.ko"

Load V4L2 capture and dependent modules

- **Load Syslink Module**

```
$ insmod syslink.ko
```

- **Load VPSS M3 Firmware**

```
$ ./slaveloader startup VPSS-M3 ti816x_hdvpss.xem3
```

- **Load VPSS Module**

```
 $ insmod vpss.ko
```

- **Load TVP7002 kernel module**

```
$ insmod tvp7002.ko
```

- **Load V4L2 Capture Module**

```
$ insmod ti81xxvin.ko
```

Note: Some of the above modules are common between V4L2 capture, V4L2 display and FBDEV modules. These modules need to be loaded only once. tvp7002.ko must be loaded before ti81xxvin.ko.

Features

V4L2 capture driver supports following features.

- Supports VIP0 instance of the VPSS.
- Supports TVP7002 standard Linux sub device driver.
- Supports YUV422 interleaved, YUV422 semiplanar, YUV420 semiplanar and RGB888 color formats to memory.
- Supports scaling and cropping for YUV formats (downscaling only)
- Supports 1080P60, 1080P30, 1080I60 and 720P60 input resolutions.
- Supports user pointer buffer mechanism of the V4L2 driver interface.

Architecture

This chapter describes the Driver Architecture and Design concepts

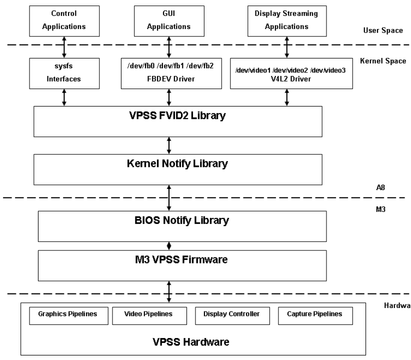
Driver Architecture

TI81XX Video Processing hardware integrates three graphics pipeline, five video pipelines, 2 capture ports and 4 video compositors. Video compositors are connected to four VENCs to support various output format and mode.

The primary functionality of the VPSS driver is to provide interfaces to user level applications and management to video processing subsystem hardware.

This includes, but is not limited to:

- GUI rendering through graphics pipelines.
- Video rendering through video pipelines.
- Connecting each of three graphics pipelines to four compositors so the display layer is presented on the selected output path.
- Connecting each of three video pipelines to four compositors so the display layer is presented on the selected output path.
- Image/Video processing(alpha blending, colorkeying, cropping)



Video Processing Subsystem Architecture

Software Design Interfaces

Above figure (Video Processing Subsystem Architecture) shows the major components that makes up the VPSS software sub-system

- **M3 VPSS Firmware**

This is a firmware running over processor Cotex M3 controlling the Video Processing subsystem hardware.

- **Syslink Library**

This is a funcional layer controlling the inter-processor communction between Cotex A8 and Cotex M3.

- **VPSS FVID2 Library**

This is a HAL/functional layer controlling the Firmware running over the M3. It exposes the number of APIs controlling the video compositors, VENCs, graphics/video pipelines to the user interface drivers like V4L2 and FBDEV. It translates the V4L2/Fbdev data structures and ioctl to FVID2 data structures and command, and call the Linux Syslink IPC notify function to pass FVID2 data structures and commands to Firmware running over the M3 processor.

- **Video Applications & V4L2 subsystem**

Video applications (camera, camcorder, image viewer, etc.) use the standard V4L2 APIs to render static images and video to the video layers, or capture/preview camera images.

Usage

Command Line arguments

V4L2 capture driver supports enabling debug option. Argument are only specified at the time of inserting the driver since V4L2 capture driver only supports dynamic build.

Below is the list of arguments which V4L2 driver supports -

V4L2 Driver Command Line Arguments

Argument	Description
debug	Enable debug messaging

Insert the dynamically built module with following parameters: Following example shows how to enable debug of V4L2:

```
$ insmod ti81xxvin.ko debug=1
```

Opening and Closing of Driver

Driver exposes four nodes corresponding to one VIP port. There are 2 instances of the VIP capture in VPSS. Each of the VIP instance can act as a single 16/24 bit port or two 8-bit ports. So overall 4 8-bit capture, or two 16 bit capture or one 16 bit and one 24 bit capture can be supported by VIP capture instances. As maximum of 4 videos can be captured, total four driver nodes are exposed to application. Based on the interface, all 4 nodes or less than that can be opened by application.

```
/* Open a video capture logical channel in blocking mode */
fd = open ("/dev/video0", O_RDWR);
if (fd == -1) {
    perror("Failed to open display device\n");
    return -1;
}
/* Closing of channel */
close (fd);
```

Setting input

V4L2 architecture supports number of decoders through standard V4L2 subdevice model. Number of different decoders can be supported for same V4L2 capture driver based on different boards. Different decoder supports different type of inputs like S-Video, Composite, Component, DVI, HDMI etc. Application can select one of the input from number of supported inputs. Following ioctl allows application to select specific input from supported inputs. V4L2 capture driver sets the input in the connected decoder as V4L2 capture driver is independent of the inputs.

```
int input_index = 0;
ioctl(fd, VIDIOC_S_INPUT, &input_index);
if (ret) {
```

```
         printf("%s: Setting Input failed\n", APP_NAME);
         printf("Return Value = %d\n", ret);
         exit(2);
}
```

Querying DV preset

DV preset is Digital video presets. It represents the current video resolution detected by the decoder. Once the input is connected to decoder, it can detect the resolution like 1080P60, 720P60 etc based on clock and control signals. This IOCTL allows application to detect the incoming video resolution. Again querying of the video resolution is a decoder feature and not of the V4L2 capture driver.

```
struct v4l2_dv_preset dv_preset;
dv_preset.preset = -1;
if (ioctl(fd, VIDIOC_QUERY_DV_PRESET, &dv_preset)) {
        printf("Querying DV Preset failed\n");
        exit(2);
}
switch (dv_preset.preset) {
case V4L2_DV_720P60:
        printf("%s:\n Mode set is 720P60\n", APP_NAME);
        break;
case V4L2_DV_1080I60:
        printf("%s:\n Mode set is 1080I60\n", APP_NAME);
        break;
case V4L2_DV_1080P60:
        printf("%s:\n Mode set is 1080P60\n", APP_NAME);
        break;
case V4L2_DV_1080P30:
        printf("%s:\n Mode set is 1080P30\n", APP_NAME);
        break;
default:
       printf("%s:\n Failed to query preset\n", APP_NAME);
}
```

Setting DV Preset

After querying for preset application to set the same preset. This is to ensure that all the registers of the decoder are set properly according the preset queried. This is a feature of decoder and not V4L2 capture driver. V4L2 capture driver simply passes the call to the respective V4L2 subdevices (decoders)driver. Following is the example code for setting up the dv preset.

```
struct v4l2_dv_preset dv_preset;
dv_preset.preset = -1;
if (ioctl(fd, VIDIOC_QUERY_DV_PRESET, &dv_preset)) {
        printf("Querying DV Preset failed\n");
        exit(2);
}
if (ioctl(capt.fd, VIDIOC_S_DV_PRESET, &capt.dv_preset)) {
        printf("Setting DV Preset failed\n");
        exit(2);
}
```

Cropping

The V4L2 Driver allows an application to define a rectangular portion of the image to be cropped via the VIDIOC_S_CROP Ioctl with the V4L2_BUF_TYPE_VIDEO_CAPTURE buffer type. When application calls VIDIOC_S_FMT ioctl, driver sets default cropping rectangle that is the largest rectangle no larger than the image size. All cropping dimensions are rounded down to even numbers. Cropping is done by the scalar module included with VIP capture instance.

Following example shows how to change crop size.

```
struct v4l2_crop crop;
crop.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
crop.c.left = 0;
crop.c.top = 0;
crop.c.width = 320;
crop.c.height = 320;
ret = ioctl(fd, VIDIOC_S_CROP, &crop);
if (ret < 0) {
    perror("VIDIOC_S_CROP\n");
    close(fd);
    exit(0);
}
/* Image cropping rectangle is now changed */
```

Scaling

V4L2 capture driver supports scaling of the captured image. Only downscaling is supported on the captured image. Scaling dimensions are rounded to even numbers. Image can be scaled upto 1/8x in horizontal direction and vertical directions.

Following example show how to scale the input image.

```
capt.fmt_win.type = V4L2_BUF_TYPE_VIDEO_OVERLAY;
capt.fmt_win.fmt.win.w.left = 0;
capt.fmt_win.fmt.win.w.top = 0;
capt.fmt_win.fmt.win.w.width = 720;
capt.fmt_win.fmt.win.w.height = 480;

if (ioctl(capt.fd, VIDIOC_S_FMT, &capt.fmt_win)) {
        printf("Setting window failed\n");
        exit(2);
}
```

Buffer Format

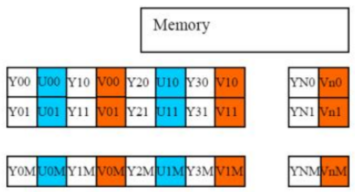
Buffer format describes various buffer parameters like pixel format, memory organization for each color component and for interlaced capture memory organization for each field. It also describes the size and pitch of the buffer. V4L2 capture driver and hardware supports following buffer formats: V4L2_PIX_FMT_YUYV (YUV422 interleaved), V4L2_PIX_FMT_NV12 (YUV420 semi-planar), V4L2_PIX_FMT_NV16 (YUV422 semi-planar)and V4L2_PIX_FMT_RGB24 (RGB888 packed). Buffer format can be changed using VIDIOC_S_FMT ioctl with type as V4L2_BUF_TYPE_VIDEO_CAPTURE.

```
/* Get the default format */
fmt.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
if (ioctl(fd, VIDIOC_G_FMT, &fmt)) {
        printf("Getting format failed");
        exit(2);
}
/* Change required parameters and set the format */
fmt.fmt.pix.width = 1920;
fmt.fmt.pix.height = 1080;
fmt.fmt.pix.bytesperline = fmt.fmt.pix.width * 2u;
fmt.fmt.pix.sizeimage = fmt.fmt.pix.bytesperline * fmt.fmt.pix.height;
fmt.fmt.pix.colorspace = V4L2_COLORSPACE_REC709;

fmt.fmt.pix.field = V4L2_FIELD_ANY;
fmt.fmt.pix.pixelformat = V4L2_PIX_FMT_YUYV;

fmt.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
if (ioctl(fd, VIDIOC_S_FMT, &fmt)) {
        printf("Setting format failed\n");
        exit(2);
}
```

Buffer Formats

- YUYV422 Interleaved Format

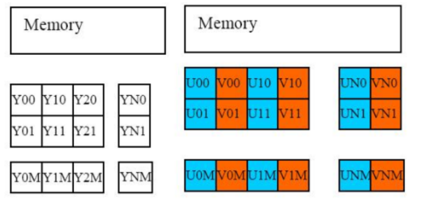
YUYV422 is expected to be packed with Y0 in the lowest byte followed by U(Cb) followed by Y1 finally V(Cr) in the upper most byte.



YUYV422 Interleaved Data Memory Organization

- YUV422 Semi-planar Format

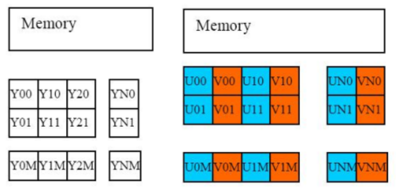
Y and UV are stored separately. UV is expected to be packed with U(Cb) in the lower byte and V(Cr) in the upper byte.



YUV422 semi-planar Data Memory Organization

- YUV420 Semi-planar Format

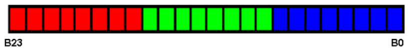
Y and UV are stored separately. UV is expected to be packed with U(Cb) in the lower byte and V(Cr) in the upper byte.



YUV420 semi-planar Data Memory Organization

- RGB24 Format

Red, Blue and Green components needs to be packed in memory.



RGB888 Data Memory Organization

Buffer Management

User pointer buffer are supported by driver. Memory mapped buffers are not supported by driver because its almost impossible to get HD size buffer from the kernel. Minimum number of buffers application should allocate and prime for no frame drops is 4. Frame drops will be observed if application primes less than 4 buffers.

The main steps that the application must perform for buffer allocation are:

- **Allocating V4L2 buffer descriptor Memory**

VIDIOC_REQBUFS

This is a necessary ioctl for streaming IO. It has to be called for both drivers buffer mode and user buffer mode. Using this ioctl, driver will identify whether driver buffer mode or user buffer mode will be used.

It takes a pointer to instance of the v4l2_requestbuffers structure as an argument.

User can specify the buffer type (V4L2_BUF_TYPE_VIDEO_OUTPUT), number of buffers, and memory type (V4L2_MEMORY_MMAP, V4L2_MEMORY_USERPTR)at the time of buffer allocation. In case of driver buffer mode, this ioctl also returns the actual number of buffers allocated in count member of v4l2_requestbuffer structure.

```
/* structure to store buffer request parameters */
struct v4l2_requestbuffers reqbuf;
reqbuf.count = numbuffers;
reqbuf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
reqbuf.memory = V4L2_MEMORY_USERPTR;
ret = ioctl(fd , VIDIOC_REQBUFS, &reqbuf);
```

```
if(ret < 0) {
    printf("cannot allocate memory\n");
    close(fd);
    return -1;
}
```

Streaming

V4L2 driver supports the streaming of the buffer. To do streaming minimum of fourbuffers should be requested by the application by using VIDIOC_REQBUFS ioctl. V4L2 capture driver support user pointer buffers only. Mmapped buffers are not supported.

```
/* Request buffers. We are operating in userPtr mode */
        capt.reqbuf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
        capt.reqbuf.count = MAX_BUFFER;
        capt.reqbuf.memory = V4L2_MEMORY_USERPTR;
        ret = ioctl(capt.fd, VIDIOC_REQBUFS, &capt.reqbuf);
        if (ret<0) {
                printf("Could not allocate the buffers\n");
                return -1;
        }

        /* Prime buffers */
        int ret, i;
        for (i = 0; i < (MAX_BUFFER; i++) {
                capt.buf.type = V4L2_BUF_TYPE_VIDEO_CAPTURE;
                capt.buf.memory = V4L2_MEMORY_USERPTR;
                capt.buf.index = i;
                capt.buf.m.userptr = (unsigned long)buffer_addr[capt.buf.index];
                capt.buf.length = capt.fmt.fmt.pix.sizeimage;
                ret = ioctl(capt.fd, VIDIOC_QBUF, &capt.buf);
                if (ret < 0) {
                        perror("VIDIOC_QBUF\n");
                        return -1;
                }
        }
        /* Start streaming */
        int a = V4L2_BUF_TYPE_VIDEO_CAPTURE, ret;
        ret = ioctl(capt.fd, VIDIOC_STREAMON, &a);
        if (ret < 0) {
                perror("VIDIOC_STREAMON\n");
                return -1;
        }
        /* Start the loop of capture */
        for (i = 0; i < MAXLOOPCOUNT; i++) {
        /* Dequeue the captured buffer */
                ret = ioctl(capt.fd, VIDIOC_DQBUF, &capt.buf);
                if (ret < 0) {
                        perror("VIDIOC_DQBUF\n");
                        return -1;
                }
                /* Process the buffer here */


                /* Queue the buffer after processing is done */
                ret = ioctl(capt.fd, VIDIOC_QBUF, &capt.buf);
                if (ret < 0) {
                         perror("VIDIOC_QBUF\n");
                         return -1;
                }
        }
        /* Stop streaming */
        int a = V4L2_BUF_TYPE_VIDEO_CAPTURE, ret;
        ret = ioctl(capt.fd, VIDIOC_STREAMOFF, &a);
        if (ret < 0) {
                perror("VIDIOC_STREAMOFF\n");
                return -1;
        }
```

V4L2 Capture Driver Interface

Application Interface

open()

To open a video device

close()

To close a video device

ioctl()

To send ioctl commands to the display driver.

mmap()

To memory map a driver allocated buffer to user space

Supported Standard IOCTLs

This section describes the standard V4L2 IOCTLs supported by the Capture Driver.

NOTE: Standard IOCTLs that are not listed here are not supported. The Display Driver handles the unsupported ones by returning EINVALerror code.

VIDIOC_QUERYCAP

This is used to query the driver's capability. The video driver fills a v4l2_capability struct indicating the driver is capable of output and streaming.

VIDIOC_ENUM_FMT

This is used to enumerate the image formats that are supported by the driver. The driver fills a v4l2_fmtdesc struct.

VIDIOC_G_FMT

This is used to get the current image format or display window depending on the buffer type. The driver fills the information to a *v4l2_format* struct. It also used to get the scaling ratio with v4l2_buf_type as

V4L2_BUF_TYPE_VIDEO_OVERLAY

VIDIOC_TRY_FMT

This is used to validate a new image format or a new display window depending on the buffer type. The driver may change the passed values if they are not supported. Application should check what is granted.

VIDIOC_S_FMT

This is used to set a new image format or a new display window depending on the buffer type. The driver may change the passed values if they are not supported. Application should check what is granted if *VIDIOC_TRY_FMT* is not used first. It is also used to set the scaling parameters with v4l2_buf_type as V4L2_BUF_TYPE_VIDEO_OVERLAY

VIDIOC_CROPCAP

This is used to get the default cropping rectangle based on the current image size and the current display panel size. The driver fills a *v4l2_cropcap* struct.

VIDIOC_G_CROP

This is used to get the current cropping rectangle. The driver fills a *v4l2_crop* struct.

VIDIOC_S_CROP

This is used to set a new cropping rectangle. The driver fills a *v4l2_crop* struct. Application should check what is granted.

VIDIOC_REQBUFS

This is used to request a number of buffers that can later be memory mapped. The driver fills a *v4l2_request* buffers struct. Application should check how many buffers are granted.

VIDIOC_QUERYBUF

This is used to get a buffer's information so mmap can be called for that buffer. The driver fills a *v4l2_buffer* struct.

VIDIOC_QBUF

This is used to queue a buffer by passing a *v4l2_buffer* struct associated to that buffer.

VIDIOC_DQBUF

This is used to dequeue a buffer by passing a *v4l2_buffer* struct associated to that buffer.

VIDIOC_STREAMON

This is used to turn on streaming. After that, any *VIDIOC_QBUF* results in an image being rendered.

VIDIOC_STREAMOFF

This is used to turn off streaming.

VIDIOC_ENUMINPUT

This is used to enumerate number of inputs supported by various decoders.

VIDIOC_S_INPUT

This is used to set one of inputs, from the number of inputs supported by decoders

VIDIOC_G_INPUT

This is used to get the currently set input

VIDIOC_QUERY_DV_PRESET

This is used to detect the preset. Decoders can generally detect the incoming digital video preset. This is used to detect it and configure the master driver according to detected preset.

VIDIOC_G_DV_PRESET

This is used to get the current preset.

VIDIOC_S_DV_PRESET

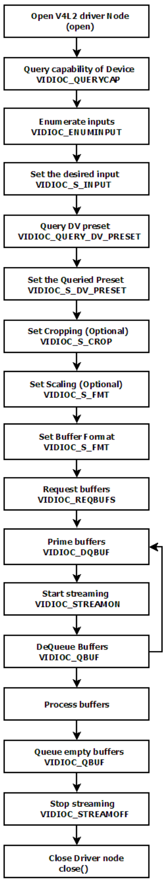
This is used to set the preset. After querying the preset, application needs to set the queried preset

VIDIOC_ENUM_DV_PRESETS

This is used to enumerate all the DV presets supported by decoder

Sample Application Flow

This chapter describes the application flow using the V4l2 driver.



Application for v4l2 driver

Sample Applications

Sample applications are provided with the PSP package to demonstrate various feature of the driver like scaling, cropping, different formats etc.

Read This First

All the sample applications of V4L2 capture driver uses FBDEV buffer as the user pointer buffers. FBDEV needs to be loaded with following arguments to make all sample applications work successfully ***insmod ti81xxfb.ko vram=0:40M,1:1M,2:1M***. This is to ensure that fbdev has enough memory for using it with V4L2 capture driver. ti81xxvo.ko must loaded before ti81xxvin.ko, which is to ensure that /dev/video1-3 are created for display function. Capture source must be connected to the component input of IO EXPANSION board before run the sample application, otherwise the sample application reports error.

Following is the list of applications for the V4L2 capture driver.

- saLoopBack

- saLoopBackFbdev
- saLoopBackScale

saLoopBack

This application demonstrates simple loop back from capture to display. Application takes input through TVP7002 decoder on VPSS VIP port. Capture buffers are displayed using V4L2 display driver. Sample application uses user pointer buffer mechanism for both capture and display driver. User pointer buffers are taken from FBDEV driver. V4L2 capture driver outputs YUV422 (y and cbcr interleaved) data to memory and display driver displays the same format from memory. Capture driver detects the incoming resolution, configures capture and display driver for same resolution. For e.g. For 1080P60 input resolution, Application sets the buffer size to 1920*1080*2 for both display and capture. Application changes the display resolution to 1080P60 on display using the sysfs entry.

saLoopBackFbdev

This application demonstrates simple loop back from capture to fbdev display driver. Application takes input through TVP7002 decoder on VPSS VIP port. Captured buffers are displayed using FBDEV driver. Sample application uses user pointer buffer mechanism for capture driver, and mmapped buffers for fbdev. FBDEV buffers are used as userpointer buffer for V4L2 capture. Capture driver takes YUV422 as input, converts it to RGB888 using color space converter of VIP port. Capture driver detects incoming resolution and configures capture and display driver for same resolution. For e.g. for 720P60 input resolution, Application sets the buffer size to 1280*720*3 for capture, and set the fbdev driver for RGB888 display. Application changes the display resolution to 720P60 using the sysfs entry.

saLoopBackScale

This application demonstrates the scaling and cropping feature of the V4L2 capture driver. Application takes the input through TVP7002 decoder on VPSS VIP port. Capture buffers are displayed using V4L2 display driver. Sample application uses user pointer buffer mechanism for both capture and display drivers. Capture driver crops the image to the same original input resolution. It then scales it to VGA(640X480) resolution and displays it through display driver.

Technical Support and Product Updates =

For further information or to report any problems, contact http://community.ti.com or http://support.ti.com.

OMAP Linux Mailing List: http://vger.kernel.org/vger-lists.html#linux-omap

References

[1] http://linux.bytesex.org/v4l2/

[2] http://v4l2spec.bytesex.org/v4l2spec/v4l2.pdf

[3] http://software-dl.ti.com/dsps/dsps_public_sw/psp/LinuxPSP/TI81XX_04_04/04_04_00_02//exports/TI81XX-LINUX-PSP-04.04.00.02.tgz

[4] http://software-dl.ti.com/dsps/dsps_public_sw/sdo_sb/targetcontent/syslink/2_20_00_14/exports/syslink_2_20_00_14.tar.gz

[5] http://software-dl.ti.com/dsps/dsps_public_sw/psp/LinuxPSP/TI81XX_04_04/04_04_00_02/exports/TI81xx-HDVPSS-01.00.01.44.zip

Article Sources and Contributors

TI81XX PSP VIDEO CAPTURE Driver User Guide *Source*: http://processors.wiki.ti.com/index.php?oldid=122567 *Contributors*: A0131716, ChrisRing, HardikShah, Kevinsc, Parth.saxena, RK, Ralph, X0186257, Yihe

Image Sources, Licenses and Contributors

Image:TIBanner.png *Source*: http://processors.wiki.ti.com/index.php?title=File:TIBanner.png *License*: unknown *Contributors*: Nsnehaprabha

Image:vpss_linux_sw_arch_3.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Vpss_linux_sw_arch_3.png *License*: unknown *Contributors*: Yihe

Image:yuyv.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Yuyv.png *License*: unknown *Contributors*: Yihe

Image:yuv422sp.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Yuv422sp.png *License*: unknown *Contributors*: Yihe

Image:yuv420sp.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Yuv420sp.png *License*: unknown *Contributors*: Yihe

Image:rgb888.png *Source*: http://processors.wiki.ti.com/index.php?title=File:Rgb888.png *License*: unknown *Contributors*: Yihe

Image:V4L2_Capture_flowchart.png *Source*: http://processors.wiki.ti.com/index.php?title=File:V4L2_Capture_flowchart.png *License*: unknown *Contributors*: HardikShah

License

THE WORK (AS DEFINED BELOW) IS PROVIDED UNDER THE TERMS OF THIS CREATIVE COMMONS PUBLIC LICENSE ("CCPL" OR "LICENSE"). THE WORK IS PROTECTED BY COPYRIGHT AND/OR OTHER APPLICABLE LAW. ANY USE OF THE WORK OTHER THAN AS AUTHORIZED UNDER THIS LICENSE OR COPYRIGHT LAW IS PROHIBITED.

BY EXERCISING ANY RIGHTS TO THE WORK PROVIDED HERE, YOU ACCEPT AND AGREE TO BE BOUND BY THE TERMS OF THIS LICENSE. TO THE EXTENT THIS LICENSE MAY BE CONSIDERED TO BE A CONTRACT, THE LICENSOR GRANTS YOU THE RIGHTS CONTAINED HERE IN CONSIDERATION OF YOUR ACCEPTANCE OF SUCH TERMS AND CONDITIONS.

License

1. Definitions

a. "**Adaptation**" means a work based upon the Work, or upon the Work and other pre-existing works, such as a translation, adaptation, derivative work, arrangement of music or other alterations of a literary or artistic work, or phonogram or performance and includes cinematographic adaptations or any other form in which the Work may be recast, transformed, or adapted including in any form recognizably derived from the original, except that a work that constitutes a Collection will not be considered an Adaptation for the purpose of this License. For the avoidance of doubt, where the Work is a musical work, performance or phonogram, the synchronization of the Work in timed-relation with a moving image ("synching") will be considered an Adaptation for the purpose of this License.

b. "**Collection**" means a collection of literary or artistic works, such as encyclopedias and anthologies, or performances, phonograms or broadcasts, or other works or subject matter other than works listed in Section 1(f) below, which, by reason of the selection and arrangement of their contents, constitute intellectual creations, in which the Work is included in its entirety in unmodified form along with one or more other contributions, each constituting separate and independent works in themselves, which together are assembled into a collective whole. A work that constitutes a Collection will not be considered an Adaptation (as defined below) for the purposes of this License.

c. "**Creative Commons Compatible License**" means a license that is listed at http://creativecommons.org/compatiblelicenses that has been approved by Creative Commons as being essentially equivalent to this License, including, at a minimum, because that license: (i) contains terms that have the same purpose, meaning and effect as the License Elements of this License; and, (ii) explicitly permits the relicensing of adaptations of works made available under that license under this License or a Creative Commons jurisdiction license with the same License Elements as this License.

d. "**Distribute**" means to make available to the public the original and copies of the Work or Adaptation, as appropriate, through sale or other transfer of ownership.

e. "**License Elements**" means the following high-level license attributes as selected by Licensor and indicated in the title of this License: Attribution, ShareAlike.

f. "**Licensor**" means the individual, individuals, entity or entities that offer(s) the Work under the terms of this License.

g. "**Original Author**" means, in the case of a literary or artistic work, the individual, individuals, entity or entities who created the Work or if no individual or entity can be identified, the publisher; and in addition (i) in the case of a performance the actors, singers, musicians, dancers, and other persons who act, sing, deliver, declaim, play in, interpret or otherwise perform literary or artistic works or expressions of folklore; (ii) in the case of a phonogram the producer being the person or legal entity who first fixes the sounds of a performance or other sounds; and, (iii) in the case of broadcasts, the organization that transmits the broadcast.

h. "**Work**" means the literary and/or artistic work offered under the terms of this License including without limitation any production in the literary, scientific and artistic domain, whatever may be the mode or form of its expression including digital form, such as a book, pamphlet and other writing; a lecture, address, sermon or other work of the same nature; a dramatic or dramatico-musical work; a choreographic work or entertainment in dumb show; a musical composition with or without words; a cinematographic work to which are assimilated works expressed by a process analogous to cinematography; a work of drawing, painting, architecture, sculpture, engraving or lithography; a photographic work to which are assimilated works expressed by a process analogous to photography; a work of applied art; an illustration, map, plan, sketch or three-dimensional work relative to geography, topography, architecture or science; a performance; a broadcast; a phonogram; a compilation of data to the extent it is protected as a copyrightable work; or a work performed by a variety or circus performer to the extent it is not otherwise considered a literary or artistic work.

i. "**You**" means an individual or entity exercising rights under this License who has not previously violated the terms of this License with respect to the Work, or who has received express permission from the Licensor to exercise rights under this License despite a previous violation.

j. "**Publicly Perform**" means to perform public recitations of the Work and to communicate to the public those public recitations, by any means or process, including by wire or wireless means or public digital performances; to make available to the public Works in such a way that members of the public may access these Works from a place and at a place individually chosen by them; to perform the Work to the public by any means or process and the communication to the public of the performances of the Work, including by public digital performance; to broadcast and rebroadcast the Work by any means including signs, sounds or images.

k. "**Reproduce**" means to make copies of the Work by any means including without limitation by sound or visual recordings and the right of fixation and reproducing fixations of the Work, including storage of a protected performance or phonogram in digital form or other electronic medium.

2. Fair Dealing Rights

Nothing in this License is intended to reduce, limit, or restrict any uses free from copyright or rights arising from limitations or exceptions that are provided for in connection with the copyright protection under copyright law or other applicable laws.

3. License Grant

Subject to the terms and conditions of this License, Licensor hereby grants You a worldwide, royalty-free, non-exclusive, perpetual (for the duration of the applicable copyright) license to exercise the rights in the Work as stated below:

a. to Reproduce the Work, to incorporate the Work into one or more Collections, and to Reproduce the Work as incorporated in the Collections;

b. to create and Reproduce Adaptations provided that any such Adaptation, including any translation in any medium, takes reasonable steps to clearly label, demarcate or otherwise identify that changes were made to the original Work. For example, a translation could be marked "The original work was translated from English to Spanish," or a modification could indicate "The original work has been modified.";

c. to Distribute and Publicly Perform the Work including as incorporated in Collections; and,

d. to Distribute and Publicly Perform Adaptations.

e. For the avoidance of doubt:

   i. **Non-waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme cannot be waived, the Licensor reserves the exclusive right to collect such royalties for any exercise by You of the rights granted under this License;

   ii. **Waivable Compulsory License Schemes.** In those jurisdictions in which the right to collect royalties through any statutory or compulsory licensing scheme can be waived, the Licensor waives the exclusive right to collect such royalties for any exercise by You of the rights granted under this License; and,

   iii. **Voluntary License Schemes.** The Licensor waives the right to collect royalties, whether individually or, in the event that the Licensor is a member of a collecting society that administers voluntary licensing schemes, via that society, from any exercise by You of the rights granted under this License.

The above rights may be exercised in all media and formats whether now known or hereafter devised. The above rights include the right to make such modifications as are technically necessary to exercise the rights in other media and formats. Subject to Section 8(f), all rights not expressly granted by Licensor are hereby reserved.

4. Restrictions

The license granted in Section 3 above is expressly made subject to and limited by the following restrictions:

a. You may Distribute or Publicly Perform the Work only under the terms of this License. You must include a copy of, or the Uniform Resource Identifier (URI) for, this License with every copy of the Work You Distribute or Publicly Perform. You may not offer or impose any terms on the Work that restrict the terms of this License or the ability of the recipient of the Work to exercise the rights granted to that recipient under the terms of the License. You may not sublicense the Work. You must keep intact all notices that refer to this License and to the disclaimer of warranties with every copy of the Work You Distribute or Publicly Perform. When You Distribute or Publicly Perform the Work, You may not impose any effective technological measures on the Work that restrict the ability of a recipient of the Work from You to exercise the rights granted to that recipient under the terms of the License. This Section 4(a) applies to the Work as incorporated in a Collection, but this does not require the Collection apart from the Work itself to be made subject to the terms of this License. If You create a Collection, upon notice from any Licensor You must, to the extent practicable, remove from the Collection any credit as required by Section 4(c), as requested. If You create an Adaptation, upon notice from any Licensor You must, to the extent practicable, remove from the Adaptation any credit as required by Section 4(c), as requested.

b. You may Distribute or Publicly Perform an Adaptation only under the terms of: (i) this License; (ii) a later version of this License with the same License Elements as this License; (iii) a Creative Commons jurisdiction license (either this or a later license version) that contains the same License Elements as this License (e.g., Attribution-ShareAlike 3.0 US)); (iv) a Creative Commons Compatible License. If you license the Adaptation under one of the licenses mentioned in (iv), you must comply with the terms of that license. If you license the Adaptation under the terms of any of the licenses mentioned in (i), (ii) or (iii) (the "Applicable License"), you must comply with the terms of the Applicable License generally and the following provisions: (I) You must include a copy of, or the URI for, the Applicable License with every copy of each Adaptation You Distribute or Publicly Perform; (II) You may not offer or impose any terms on the Adaptation that restrict the terms of the Applicable License or the ability of the recipient of the Adaptation to exercise the rights granted to that recipient under the terms of the Applicable License; (III) You must keep intact all notices that refer to the Applicable License and to the disclaimer of warranties with every copy of the Work as included in the Adaptation You Distribute or Publicly Perform; (IV) when You Distribute or Publicly Perform the Adaptation, You may not impose any effective technological measures on the Adaptation that restrict the ability of a recipient of the Adaptation from You to exercise the rights granted to that recipient under the terms of the Applicable License. This Section 4(b) applies to the Adaptation as incorporated in a Collection, but this does not require the Collection apart from the Adaptation itself to be made subject to the terms of the Applicable License.

c. If You Distribute, or Publicly Perform the Work or any Adaptations or Collections, You must, unless a request has been made pursuant to Section 4(a), keep intact all copyright notices for the Work and provide, reasonable to the medium or means You are utilizing: (i) the name of the Original Author (or pseudonym, if applicable) if supplied, and/or if the Original Author and/or Licensor designate another party or parties (e.g., a sponsor institute, publishing entity, journal) for attribution ("Attribution Parties") in Licensor's copyright notice, terms of service or by other reasonable means, the name of such party or parties; (ii) the title of the Work if supplied; (iii) to the extent reasonably practicable, the URI, if any, that Licensor specifies to be associated with the Work, unless such URI does not refer to the copyright notice or licensing information for the Work; and (iv) , consistent with Ssection 3(b), in the case of an Adaptation, a credit identifying the use of the Work in the Adaptation (e.g., "French translation of the Work by Original Author," or "Screenplay based on original Work by Original Author"). The credit required by this Section 4(c) may be implemented in any reasonable manner; provided, however, that in the case of a Adaptation or Collection, at a minimum such credit will appear, if a credit for all contributing authors of the Adaptation or Collection appears, then as part of these credits and in a manner at least as prominent as the credits for the other contributing authors. For the avoidance of doubt, You may only use the credit required by this Section for the purpose of attribution in the manner set out above and, by exercising Your rights under this License, You may not implicitly or explicitly assert or imply any connection with, sponsorship or endorsement by the Original Author, Licensor and/or Attribution Parties, as appropriate, of You or Your use of the Work, without the separate, express prior written permission of the Original Author, Licensor and/or Attribution Parties.

d. Except as otherwise agreed in writing by the Licensor or as may be otherwise permitted by applicable law, if You Reproduce, Distribute or Publicly Perform the Work either by itself or as part of any Adaptations or Collections, You must not distort, mutilate, modify or take other derogatory action in relation to the Work which would be prejudicial to the Original Author's honor or reputation. Licensor agrees that in those jurisdictions (e.g. Japan), in which any exercise of the right granted in Section 3(b) of this License (the right to make Adaptations) would be deemed to be a distortion, mutilation, modification or other derogatory action prejudicial to the Original Author's honor and reputation, the Licensor will waive or not assert, as appropriate, this Section, to the fullest extent permitted by the applicable national law, to enable You to reasonably exercise Your right under Section 3(b) of this License (right to make Adaptations) but not otherwise.

5. Representations, Warranties and Disclaimer

UNLESS OTHERWISE MUTUALLY AGREED TO BY THE PARTIES IN WRITING, LICENSOR OFFERS THE WORK AS-IS AND MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND CONCERNING THE WORK, EXPRESS, IMPLIED, STATUTORY OR OTHERWISE, INCLUDING, WITHOUT LIMITATION, WARRANTIES OF TITLE, MERCHANTIBILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT, OR THE ABSENCE OF LATENT OR OTHER DEFECTS, ACCURACY, OR THE PRESENCE OF ABSENCE OF ERRORS, WHETHER OR NOT DISCOVERABLE. SOME JURISDICTIONS DO NOT ALLOW THE EXCLUSION OF IMPLIED WARRANTIES, SO SUCH EXCLUSION MAY NOT APPLY TO YOU.

6. Limitation on Liability

EXCEPT TO THE EXTENT REQUIRED BY APPLICABLE LAW, IN NO EVENT WILL LICENSOR BE LIABLE TO YOU ON ANY LEGAL THEORY FOR ANY SPECIAL, INCIDENTAL, CONSEQUENTIAL, PUNITIVE OR EXEMPLARY DAMAGES ARISING OUT OF THIS LICENSE OR THE USE OF THE WORK, EVEN IF LICENSOR HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

7. Termination

a. This License and the rights granted hereunder will terminate automatically upon any breach by You of the terms of this License. Individuals or entities who have received Adaptations or Collections from You under this License, however, will not have their licenses terminated provided such individuals or entities remain in full compliance with those licenses. Sections 1, 2, 5, 6, 7, and 8 will survive any termination of this License.
b. Subject to the above terms and conditions, the license granted here is perpetual (for the duration of the applicable copyright in the Work). Notwithstanding the above, Licensor reserves the right to release the Work under different license terms or to stop distributing the Work at any time; provided, however that any such election will not serve to withdraw this License (or any other license that has been, or is required to be, granted under the terms of this License), and this License will continue in full force and effect unless terminated as stated above.

8. Miscellaneous

a. Each time You Distribute or Publicly Perform the Work or a Collection, the Licensor offers to the recipient a license to the Work on the same terms and conditions as the license granted to You under this License.
b. Each time You Distribute or Publicly Perform an Adaptation, Licensor offers to the recipient a license to the original Work on the same terms and conditions as the license granted to You under this License.
c. If any provision of this License is invalid or unenforceable under applicable law, it shall not affect the validity or enforceability of the remainder of the terms of this License, and without further action by the parties to this agreement, such provision shall be reformed to the minimum extent necessary to make such provision valid and enforceable.
d. No term or provision of this License shall be deemed waived and no breach consented to unless such waiver or consent shall be in writing and signed by the party to be charged with such waiver or consent.
e. This License constitutes the entire agreement between the parties with respect to the Work licensed here. There are no understandings, agreements or representations with respect to the Work not specified here. Licensor shall not be bound by any additional provisions that may appear in any communication from You. This License may not be modified without the mutual written agreement of the Licensor and You.
f. The rights granted under, and the subject matter referenced, in this License were drafted utilizing the terminology of the Berne Convention for the Protection of Literary and Artistic Works (as amended on September 28, 1979), the Rome Convention of 1961, the WIPO Copyright Treaty of 1996, the WIPO Performances and Phonograms Treaty of 1996 and the Universal Copyright Convention (as revised on July 24, 1971). These rights and subject matter take effect in the relevant jurisdiction in which the License terms are sought to be enforced according to the corresponding provisions of the implementation of those treaty provisions in the applicable national law. If the standard suite of rights granted under applicable copyright law includes additional rights not granted under this License, such additional rights are deemed to be included in the License; this License is not intended to restrict the license of any rights under applicable law.