Category:V4L2

Contents

Stable: 17.06.2020 - 15:27 / Revision: 16.01.2020 - 08:05

Category:V4L2

Stable: 17.06.2020 - 10:27 | Revision: 16.01.2020 - 08:08

A quality version of this page, approved on *17 June 2020*, was based off this revision.

This category groups together all articles related to the Linux® **V4L2** software framework.

It is recommended to first read the V4L2 camera overview article.

Video 4 Linux version 2

Category:V4L2

Pages in category "V4L2"

The following 6 pages are in this category, out of 6 total.

- DCMI device tree configuration
- How to make a camera preview
- How to stream camera over network
- V4L2 camera overview
- V4l2-ctl
- Yavta

Stable: 07.01.2021 - 10:03 / Revision: 10.11.2020 - 10:57

A quality version of this page, approved on *7 January 2021*, was based off this revision.

Contents

1 Article purpose

This article explains how to configure the **DCMI** internal peripheral when assigned to the Linux®OS. In that case, it is controlled by the V4L2 camera framework.

The configuration is performed using the device tree mechanism that provides a hardware description of the DCMI peripheral, used by the STM32 DCMI Linux driver or by the V4L2 camera framework.

If the peripheral is assigned to another execution context, refer to How to assign an internal peripheral to a runtime context article for guidelines on peripheral assignment and configuration.

2 DT bindings documentation

The DCMI internal peripheral is documented through the STM32 DCMI device tree bindings file[1].

3 DT configuration

This hardware description is a combination of the **STM32 microprocessor** device tree files (*.dtsi* extension) and **board** device tree files (*.dts* extension). See the device tree article for an explanation of the device tree file split.

STM32CubeMX can be used to generate the board device tree. Refer to How to configure the DT using STM32CubeMX for more details.

3.1 DT configuration (STM32 level)

The DCMI device tree node is declared in stm32mp151.dtsi [2].

The declaration (shown below) provides the hardware registers base address, interrupts, reset line, clocks and dma channel used.

```
dcmi: dcmi@4c006000 {
        compatible = "st,stm32-dcmi";
        reg = <0x4c006000 0x400>;
        interrupts = <GIC_SPI 78 IRQ_TYPE_LEVEL_HIGH>;
        resets = <&rcc CAMITF_R>;
        clocks = <&rcc DCMI>;
        clock-names = "mclk";
        dmas = <&dmamux1 75 0x400 0xe0000001>;
        dma-names = "tx";
        status = "disabled";
};
```

Warning

This device tree part is related to STM32 microprocessors. It must be kept as is, without being modified by the end-user.

When using a different sensor camera device, only the sensor-related configuration part must be adapted in the associated board devicetree file (see #DT configuration (board level)).

Refer to stm32-dcmi bindings[1] for more details.

3.2 DT configuration (board level)

```
&dcmi {
        status = "okay";
        pinctrl-names = "default", "sleep";
        pinctrl-0 = <&dcmi_pins_a>;
        pinctrl-1 = <&dcmi_sleep_pins_a>;

        port {
                dcmi_0: endpoint {
                        remote-endpoint = <&ov5640_0>;
                        bus-width = <8>;
                        hsync-active = <0>;
```

```
                        vsync-active = <0>;
                        pclk-sample = <1>;
                        pclk-max-frequency = <77000000>;
                };
        };
};
```

This section, part of the STM32MP15 evaluation board device tree file[3], shows how is configured the DCMI hardware block to interconnect with the sensor camera device. The configurable settings are the following:

- Camera sensor endpoint: in this case, the Omnivision OV5640 model[4].
- Bus width: 8, 10, 12 or 14 bits
- Horizontal synchronization line level: active low (0) or active high (1)
- Vertical synchronization line level: active low (0) or active high (1)
- Pixel clock polarity line level: active low (0) or active high (1)
- Pixel clock maximum frequency in Hertz

This section also defines what is the DCMI pins multiplexing used for this board (*<&dcmi_pins_a>, <&dcmi_sleep_pins_a>*), exact pins details being defined in the STM32MP15 evaluation board pinctrl device tree file[5]:

```
                        dcmi_pins_a: dcmi-0 {
                                pins {
                                        pinmux = <STM32_PINMUX('H', 8,  AF13)>,/* DCMI_HSY
NC */
                                                 <STM32_PINMUX('B', 7,  AF13)>,/* DCMI_VSY
NC */
                                                 <STM32_PINMUX('A', 6,  AF13)>,/* DCMI_PIX
CLK */
                                                 <STM32_PINMUX('H', 9,  AF13)>,/* DCMI_D0
*/
                                                 <STM32_PINMUX('H', 10, AF13)>,/* DCMI_D1
*/
                                                 <STM32_PINMUX('H', 11, AF13)>,/* DCMI_D2
*/
                                                 <STM32_PINMUX('H', 12, AF13)>,/* DCMI_D3
*/
                                                 <STM32_PINMUX('H', 14, AF13)>,/* DCMI_D4
*/
                                                 <STM32_PINMUX('I', 4,  AF13)>,/* DCMI_D5
*/
                                                 <STM32_PINMUX('B', 8,  AF13)>,/* DCMI_D6
*/
                                                 <STM32_PINMUX('E', 6,  AF13)>,/* DCMI_D7
*/
                                                 <STM32_PINMUX('I', 1,  AF13)>,/* DCMI_D8
*/
                                                 <STM32_PINMUX('H', 7,  AF13)>,/* DCMI_D9
*/
                                                 <STM32_PINMUX('I', 3,  AF13)>,/* DCMI_D10
*/
                                                 <STM32_PINMUX('H', 15, AF13)>;/* DCMI_D11
*/
                                        bias-disable;
                                };
                        };

                        dcmi_sleep_pins_a: dcmi-sleep-0 {
                                pins {
                                        pinmux = <STM32_PINMUX('H', 8,  ANALOG)>,/* DCMI_H
SYNC */
                                                 <STM32_PINMUX('B', 7,  ANALOG)>,/* DCMI_V
SYNC */
```

```
                                                        <STM32_PINMUX('A', 6,  ANALOG)>,/* DCMI_P
IXCLK */
                                                        <STM32_PINMUX('H', 9,  ANALOG)>,/* DCMI_D
0 */
                                                        <STM32_PINMUX('H', 10, ANALOG)>,/* DCMI_D
1 */
                                                        <STM32_PINMUX('H', 11, ANALOG)>,/* DCMI_D
2 */
                                                        <STM32_PINMUX('H', 12, ANALOG)>,/* DCMI_D
3 */
                                                        <STM32_PINMUX('H', 14, ANALOG)>,/* DCMI_D
4 */
                                                        <STM32_PINMUX('I', 4,  ANALOG)>,/* DCMI_D
5 */
                                                        <STM32_PINMUX('B', 8,  ANALOG)>,/* DCMI_D
6 */
                                                        <STM32_PINMUX('E', 6,  ANALOG)>,/* DCMI_D
7 */
                                                        <STM32_PINMUX('I', 1,  ANALOG)>,/* DCMI_D
8 */
                                                        <STM32_PINMUX('H', 7,  ANALOG)>,/* DCMI_D
9 */
                                                        <STM32_PINMUX('I', 3,  ANALOG)>,/* DCMI_D
10 */
                                                        <STM32_PINMUX('H', 15, ANALOG)>;/* DCMI_D
11 */
                                };
                        };
```

An alternate pin multiplexing could be defined (for example to fit a new board design) by modifying the STM32MP15 evaluation board pinctrl device tree file[5] following the possible pins assignment defined in the MPU reference manual[6].

STM32CubeMX [7] pins configurator is of great help to find valid alternatives thanks to its visual GUI.

Refer to STM32 DCMI bindings[1] for more details.

3.3 DT configuration examples

```
        ov5640: camera@3c {
                compatible = "ovti,ov5640";
                reg = <0x3c>;
                clocks = <&clk_ext_camera>;
                clock-names = "xclk";
                DOVDD-supply = <&v2v8>;
                powerdown-gpios = <&stmfx_pinctrl 18 (GPIO_ACTIVE_HIGH | GPIO_PUSH_PULL)>;
                reset-gpios = <&stmfx_pinctrl 19 (GPIO_ACTIVE_LOW | GPIO_PUSH_PULL)>;
                rotation = <180>;
                status = "okay";

                port {
                        ov5640_0: endpoint {
                                remote-endpoint = <&dcmi_0>;
                                bus-width = <8>;
                                data-shift = <2>; /* lines 9:2 are used */
                                hsync-active = <0>;
                                vsync-active = <0>;
                                pclk-sample = <1>;
                                pclk-max-frequency = <77000000>;
                        };
                };
        };
```

This section, part of the STM32MP15 evaluation board device tree file[3], enables the support of the OV5640 Omnivision camera sensor[4] located on the MB1379 camera daughter board[8] connected to the CN7 camera connector[9] of the STM32MP15 evaluation board[10].

Refer to the OV5640 bindings [4] for more details.

Documentation on various V4L2 camera sensors can be found inside I2C media bindings folder[11]. Refer to the dedicated sensor binding documentation to adapt your board devicetree file to this dedicated sensor.

4 How to configure the DT using STM32CubeMX

The STM32CubeMX tool can be used to configure the STM32MPU device and get the corresponding platform configuration device tree files.
The STM32CubeMX may not support all the properties described in the above DT bindings documentation paragraph. If so, the tool inserts **user sections** in the generated device tree. These sections can then be edited to add some properties and they are preserved from one generation to another. Refer to STM32CubeMX user manual for further information.

5 References

Please refer to the following links for additional information:

- 1.01.11.2 Linux kernel STM32 DCMI bindings (st,stm32-dcmi.txt)
- Linux kernel STM32MP157C device tree (stm32mp157c.dtsi
- 3.03.1 Linux kernel STM32MP157 evaluation board device tree (stm32mp157c-ev1.dts)
- 4.04.14.2 Linux kernel OV5640 bindings (ov5640.txt)
- 5.05.1 Linux kernel STM32MP15 pinctrl device tree (stm32mp15-pinctrl.dtsi)
- STM32MP15 reference manuals
- STM32CubeMX
- MB1379 camera daughter board
- STM32MP157x-EV1 Evaluation board CN7 Camera sensor connector
- STM32MP15 evaluation board
- Linux kernel I2C media devices bindings (bindings/media/i2c)

Linux® is a registered trademark of Linus Torvalds.

Operating System

Digital Camera Memory Interface

Video 4 Linux version 2

Device Tree

Generic Interrupt Controller

Serial Peripheral Interface

Microprocessor Unit

Graphical User Interface

General-Purpose Input/Output (A realization of open ended transmission between devices on an embedded level. These pins available on a processor can be programmed to be used to either accept input or provide output to external devices depending on user desires and applications requirements.)

Inter-Integrated Circuit (Bi-directional 2-wire bus standard for efficient inter-IC control.)

Stable: 07.10.2019 - 07:50 / Revision: 07.10.2019 - 07:49

A quality version of this page, approved on *7 October 2019*, was based off this revision.

Refer to V4L2_camera_overview#Fullscreen_preview for a basic example of camera preview thanks to GStreamer application on top of V4L2 Linux® kernel framework

Stable: 10.04.2020 - 15:50 / Revision: 10.04.2020 - 15:48

A quality version of this page, approved on *10 April 2020*, was based off this revision.

1 Overview

This article will explain how to stream camera content over network thanks to GStreamer application on top of V4L2 Linux® kernel framework.

Capturing compressed JPEG pictures is an efficient way to send camera images to any local or remote player; JPEG pictures require a limited bandwidth while being fully interoperable.

Find below some examples of command lines allowing to capture a continuous JPEG stream while playing it using various multimedia players, either local or remote.

2 Local streaming

Here is an example of a local preview involving V4l2-ctl for JPEG pictures capture and gst-play GStreamer player for JPEG decoding and display. JPEG pictures are sent over a standard Linux pipe.

```
Board $> v4l2-ctl --set-parm=30;v4l2-ctl --set-fmt-video=width=640,height=480,
pixelformat=JPEG --stream-mmap --stream-count=-1 --stream-to=- 2>/dev/null | gst-play-1.0
"fd://0"
```

--stream-to=- tells V4l2-ctl to output binary captured content to standard output, which is then sent to pipe **|**.

Special URI **fd://0** tells gst-play GStreamer player to read data from the pipe.

Note the **2>/dev/null** right after the V4l2-ctl command to remove the logs from console output.

3 UDP streaming

Information

An internet connection is required, for example by plugging an ethernet cable on the STM32MP157x-EV1 Evaluation board CN3 ethernet connector

Get first the IP address **aa.bb.cc.dd** of the host PC using ifconfig command:

```
PC $> ifconfig | grep "inet addr"
   inet addr:aa.bb.cc.dd  Bcast:10.201.23.255  Mask:255.255.252.0
   inet addr:127.0.0.1  Mask:255.0.0.0
```

Then fill the host= udpsink property with this IP address on the remote side:

```
Board $> v4l2-ctl --set-parm=30;v4l2-ctl --set-fmt-video=width=640,height=480,
pixelformat=JPEG --stream-mmap --stream-count=-1 --stream-to=- 2>/dev/null | gst-launch-
1.0 fdsrc ! jpegparse ! rtpjpegpay ! udpsink host=aa.bb.cc.dd port=5000
```

Then play the UDP stream on host PC:

```
PC $> gst-launch-1.0 udpsrc port=5000 ! application/x-rtp, encoding-name=JPEG !
rtpjpegdepay ! jpegparse ! decodebin ! autovideosink
```

A new window will popup on host PC displaying the camera content.

Information

Due to SDP protocol signaling, this solution is not fully interoperable because it needs a dedicated GStreamer command line to be played on host side

Stable: 26.02.2021 - 17:14 / Revision: 26.02.2021 - 17:13

A quality version of this page, approved on *26 February 2021*, was based off this revision.

This article gives information about the Linux®V4L2 camera framework.

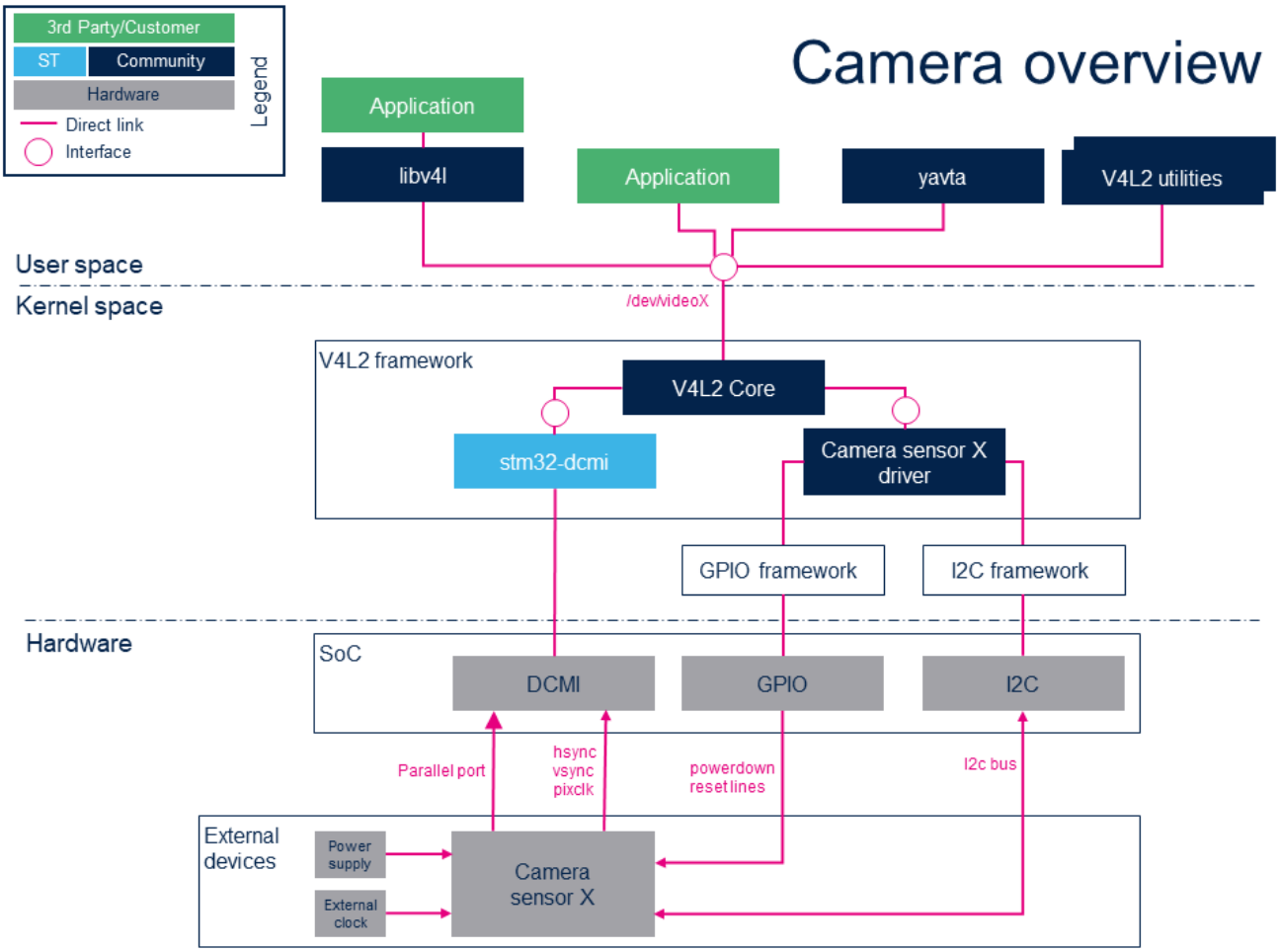
Contents

1 Framework purpose

The V4L2 Linux kernel framework[1] allows to control both an external camera sensor and the camera interface in order to capture raw frames in various pixel formats or encoded stream data such as JPEG.

This could be typically used, with the help of other Linux multimedia frameworks and applications, to take snapshot, to make preview, to make a video recording or even remotely stream images from the camera sensor.

2 System overview



2.1 Component description

- **Application** (User space)

Any application relying on V4L2 Linux kernel interface or libv4l abstraction layer. GStreamer framework provides such application.

- **V4L2 utilities** (User space)

A set of tools to test, configure and use the whole camera subsystem, including the external camera sensor and the camera interface. V4l2-ctl is one of the most usefull utility.

- **V4L2 libraries (libv4l)** (User space)

A set of libraries on top of the V4L2 Linux kernel interface which abstract the kernel interface in order to simplify, keep compatibility or add some hooks between V4L-based applications and the V4L2 kernel interface.

- **yavta** (User space)

Yavta is a test tool which relies on the V4L2 Linux kernel interface.

- **V4L2 core** (Kernel space)

This layer represents the standard Linux kernel V4L2 Framework.

- **stm32-dcmi** (Kernel space)

This V4L2 DCMI Linux device driver handles the DCMI hardware block.

- **Camera sensor X driver** (Kernel space)

This V4L2 Linux device driver handles the camera sensor X external peripheral, handles some GPIOs lines and potentially power supplies to power-up/down the camera sensor. The communication with camera sensor is done through the i2c bus.

- **DCMI** (Hardware)

The Digital Camera Memory Interface hardware block.

- **Camera sensor X** (Hardware)

The camera sensor external peripheral.

2.2 APIs description

The V4L2 userland API is documented in the Linux Media subsystem documentation[2]

The V4L2 kernel framework internal API is documented in the V4L2 Kernel Support section of the Linux Kernel documentation[3]

3 Configuration

3.1 Kernel configuration

The STM32 camera interface and OV5640 camera sensor are enabled by default in STMicroelectronics deliveries.

Nevertheless this is not the case when using upstream kernel version. In this case, the DCMI V4L2 driver can be enabled using Linux kernel menuconfig tool:

```
[*] Device Drivers --->
    [*] Multimedia support --->
        [*] V4L platform devices  --->
            [*] STM32 Digital Camera Memory Interface (DCMI) support
```

The external camera sensor connected to the camera interface must also be enabled, here is an example with the OV5640 Omnivision camera sensor located on the MB1379 camera daughter board[4] connected to the CN7 camera connector[5] of the STM32MP15 evaluation board[6]:

```
[*] Device Drivers --->
    [*] Multimedia support --->
        I2C Encoders, decoders, sensors and other helper chips  --->
            [*] OmniVision OV5640 sensor support
```

3.2 Device tree configuration

Refer to DCMI device tree configuration article for a complete view of DCMI & sensor configuration thanks to Linux kernel device tree mechanism.

4 How to use the framework

The use cases described here are enabled using V4l2-ctl, gst-launch or gst-play command line utilities.

4.1 List the video devices and their capabilities

List all the available video devices using **--list-devices** option:

```
Board $> v4l2-ctl --list-devices
```

```
STM32 Camera Memory Interface (platform:dcmi):
        /dev/video0
```

If several devices are available, use **-d** option after any V4l2-ctl commands to target a specific device. If **-d** option is not specified, /dev/video0 is targeted by default.

In order to have information on a specific device, use **-D** option:

```
Board $> v4l2-ctl -d /dev/video0 -D
```

```
Driver Info (not using libv4l2):
        Driver name   : stm32-dcmi
        Card type     : STM32 Camera Memory Interface
        Bus info      : platform:dcmi
        Driver version: X.Y.Z
        Capabilities  : 0x85200001
                Video Capture
                Read/Write
                Streaming
                Extended Pix Format
                Device Capabilities
        Device Caps   : 0x05200001
                Video Capture
                Read/Write
                Streaming
                Extended Pix Format
```

4.2 Controlling camera

Use V4l2-ctl with **-L** option to get the list of supported controls:

```
Board $> v4l2-ctl -L
```

```
User Controls

                     contrast (int)    : min=0 max=255 step=1 default=0 value=0
flags=slider
                   saturation (int)    : min=0 max=255 step=1 default=64 value=64
flags=slider
                          hue (int)    : min=0 max=359 step=1 default=0 value=0
flags=slider
        white_balance_automatic (bool)   : default=1 value=1 flags=update
                   red_balance (int)    : min=0 max=4095 step=1 default=0 value=128
flags=inactive, slider
                  blue_balance (int)    : min=0 max=4095 step=1 default=0 value=128
flags=inactive, slider
                     exposure (int)    : min=0 max=65535 step=1 default=0 value=885
flags=inactive, volatile
                gain_automatic (bool)   : default=1 value=1 flags=update
                          gain (int)    : min=0 max=1023 step=1 default=0 value=32
flags=inactive, volatile
               horizontal_flip (bool)   : default=0 value=0
                 vertical_flip (bool)   : default=0 value=0

Camera Controls

                 auto_exposure (menu)   : min=0 max=1 default=0 value=0 flags=update
                               0: Auto Mode
                               1: Manual Mode

Image Processing Controls

                  test_pattern (menu)   : min=0 max=1 default=0 value=0
                               0: Disabled
                               1: Color bars
```

Information

"value=" field returns the current value of the control

The control value can be changed thanks to **--set-ctrl** option, for example:

```
Board $> v4l2-ctl --set-ctrl test_pattern=1
```

The control value can be changed dynamically. In the following example, the color bar is enabled/disabled while preview is running:

- Start preview in background

```
Board $> gst-launch-1.0 v4l2src ! "video/x-raw, width=1280, Height=720, framerate=
(fraction)15/1" ! queue ! autovideosink -e &
```

- Then alternate the color bar activation or not

```
Board $> v4l2-ctl --set-ctrl test_pattern=1;sleep 1;v4l2-ctl --set-ctrl test_pattern=0;
sleep 1;v4l2-ctl --set-ctrl test_pattern=1;sleep 1;v4l2-ctl --set-ctrl test_pattern=0;
killall gst-launch-1.0
```

- Horizontal/vertical flip can also be changed while preview is running:

```
Board $> v4l2-ctl --set-ctrl horizontal_flip=1;sleep 2;v4l2-ctl --set-ctrl
horizontal_flip=0;sleep 2;v4l2-ctl --set-ctrl vertical_flip=1;sleep 2;v4l2-ctl --set-ctrl
vertical_flip=0;killall gst-launch-1.0
```

4.3 Set the pixel format, resolution and framerate

Use --list-formats-ext option to get the supported pixel format, resolution and framerate:

```
Board $> v4l2-ctl --list-formats-ext
```

```
ioctl: VIDIOC_ENUM_FMT
        Index       : 0
        Type        : Video Capture
        Pixel Format: 'JPEG' (compressed)
        Name        : JFIF JPEG
                Size: Discrete 176x144
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 320x240
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 640x480
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 720x480
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 720x576
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 1024x768
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 1280x720
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 1920x1080
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 2592x1944
                        Interval: Discrete 0.067s (15.000 fps)

        Index       : 1
        Type        : Video Capture
        Pixel Format: 'UYVY'
        Name        : UYVY 4:2:2
                Size: Discrete 176x144
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 320x240
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 640x480
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 720x480
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
```

```
                Size: Discrete 720x576
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 1024x768
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 1280x720
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 1920x1080
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 2592x1944
                        Interval: Discrete 0.067s (15.000 fps)

        Index       : 2
        Type        : Video Capture
        Pixel Format: 'YUYV'
        Name        : YUYV 4:2:2
                Size: Discrete 176x144
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 320x240
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 640x480
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 720x480
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 720x576
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 1024x768
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 1280x720
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 1920x1080
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 2592x1944
                        Interval: Discrete 0.067s (15.000 fps)

        Index       : 3
        Type        : Video Capture
        Pixel Format: 'RGBP'
        Name        : 16-bit RGB 5-6-5
                Size: Discrete 176x144
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 320x240
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 640x480
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 720x480
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 720x576
                        Interval: Discrete 0.067s (15.000 fps)
                        Interval: Discrete 0.033s (30.000 fps)
                Size: Discrete 1024x768
                        Interval: Discrete 0.067s (15.000 fps)
```

```
                Interval: Discrete 0.033s (30.000 fps)
        Size: Discrete 1280x720
                Interval: Discrete 0.067s (15.000 fps)
                Interval: Discrete 0.033s (30.000 fps)
        Size: Discrete 1920x1080
                Interval: Discrete 0.067s (15.000 fps)
                Interval: Discrete 0.033s (30.000 fps)
        Size: Discrete 2592x1944
                Interval: Discrete 0.067s (15.000 fps)
```

In order to change the camera configuration, select first the framerate using **--set-parm** option:

```
Board $> v4l2-ctl --set-parm=30
```

Then select the wanted resolution and/or pixel format using **--set-fmt-video** option:

```
Board $> v4l2-ctl --set-fmt-video=width=320,height=240,pixelformat=RGBP
```

4.4 Set the framerate

- With V4l2-ctl, use **--set-parm** option giving the framerate numerator only, the denominator is fixed to 1 (only integer framerate values are allowed):

```
Board $> v4l2-ctl --set-parm=<framerate numerator>
```

Take 100 VGA pictures at 30fps:

```
Board $> v4l2-ctl --set-parm=30;v4l2-ctl --set-fmt-video=width=640,height=480,
pixelformat=JPEG --stream-mmap --stream-count=100 --stream-to=pics@30fps.jpeg
```

Replay at 30fps using gst-play:

```
Board $> gst-play-1.0 pics@30fps.jpeg --videosink="videorate ! video/x-raw, framerate=
(fraction)30/1 ! autovideosink"
```

Take 100 VGA pictures at 15fps:

```
Board $> v4l2-ctl --set-parm=15;v4l2-ctl --set-fmt-video=width=640,height=480,
pixelformat=JPEG --stream-mmap --stream-count=100 --stream-to=pics@15fps.jpeg
```

Replay at 15fps using gst-play:

```
Board $> gst-play-1.0 pics@15fps.jpeg --videosink="videorate ! video/x-raw, framerate=
(fraction)15/1 ! autovideosink"
```

- With GStreamer, using **framerate** caps:

```
Board $> gst-launch-1.0 v4l2src ! "video/x-raw, ... framerate=(fraction)<numerator>
/<denominator>" ! ...
```

Preview VGA@30fps

```
Board $> gst-launch-1.0 v4l2src ! "video/x-raw, width=640, Height=480, framerate=
(fraction)30/1" ! queue ! autovideosink -e
```

Preview VGA@15fps

```
Board $> gst-launch-1.0 v4l2src ! "video/x-raw, width=640, Height=480, framerate=
(fraction)15/1" ! queue ! autovideosink -e
```

4.5 Grab a raw frame

Capture a QVGA RGB565 raw frame on disk:

```
Board $> v4l2-ctl --set-fmt-video=width=320,height=240,pixelformat=RGBP --stream-mmap --
stream-count=1 --stream-to=grab-320x240-rgb565.raw
```

In order to display it, this raw frame must be converted first to JPEG:

```
Board $> gst-launch-1.0 filesrc location= grab-320x240-rgb565.raw blocksize=153600 !
"video/x-raw, format=(string)RGB16, width=(int)320, height=(int)240, framerate=(fraction)
30/1" ! videoconvert ! jpegenc ! filesink location=grab-320x240-rgb565.jpeg
```

Then weston-image utility can be used to display this JPEG file:

```
Board $> weston-image grab-320x240-rgb565.jpeg
```

4.6 Fullscreen preview

```
Board $> gst-launch-1.0 v4l2src ! "video/x-raw, width=1280, Height=720, framerate=
(fraction)15/1" ! queue ! autovideosink -e
```

Information

Please note that GStreamer overwrites all the parameters that could have been previously set on the video device (for ex. parameters set through V4l2-ctl commands such as resolution, pixel format, framerate, ...)

4.7 Take a picture

Capture a 5Mp JPEG:

```
Board $> v4l2-ctl --set-parm=15; v4l2-ctl --set-fmt-video=width=2592,height=1944,
pixelformat=JPEG --stream-mmap --stream-count=1 --stream-skip=3 --stream-to=pic-5Mp.jpeg;
v4l2-ctl --set-parm=30
```

Then display it:

```
Board $> weston-image pic-5Mp.jpeg
```

You can check the picture resolution using gst-typefind:

```
Board $> gst-typefind-1.0 pic-5Mp.jpeg
```

```
pic-5Mp.jpeg - image/jpeg, width=(int)2592, height=(int)1944, sof-marker=(int)0
```

4.8 Pictures streaming over network

Refer to How to stream camera over network article to get some examples on how to stream camera content over network.

5 How to trace and debug

5.1 How to monitor

5.1.1 Check of devicetree configuration

Here are some commands to verify that DCMI is enabled, check which camera sensor is used and log other details about devicetree settings:

```
 rm devicetree.txt
 echo "[devicetree]" >> devicetree.txt
 echo "|-[dcmi]" >> devicetree.txt
 find /proc/device-tree/soc | grep dcmi | sed 's/\/proc\/device-tree\/soc\//| |-/' >>
devicetree.txt
 echo "|" >> devicetree.txt
 echo "|-[camera:" | tr -d "\n" >> devicetree.txt
 cat /proc/device-tree/soc/i2c*/camera*/compatible >> devicetree.txt
 echo "]" >> devicetree.txt
 find /proc/device-tree/soc | grep camera | sed 's/\/proc\/device-tree\/soc\//  |-/' >>
devicetree.txt
 echo "" >> devicetree.txt
 cat devicetree.txt
```

```
[devicetree]
|-[dcmi]
| |-dcmi@4c006000
| |-dcmi@4c006000/compatible
| |-dcmi@4c006000/clocks
| |-dcmi@4c006000/resets
| |-dcmi@4c006000/pinctrl-1
| |-dcmi@4c006000/port
| |-dcmi@4c006000/port/endpoint
| |-dcmi@4c006000/port/endpoint/hsync-active
| |-dcmi@4c006000/port/endpoint/vsync-active
| |-dcmi@4c006000/port/endpoint/remote-endpoint
| |-dcmi@4c006000/port/endpoint/bus-width
| |-dcmi@4c006000/port/endpoint/pclk-sample
| |-dcmi@4c006000/port/endpoint/phandle
| |-dcmi@4c006000/port/endpoint/linux,phandle
| |-dcmi@4c006000/port/endpoint/name
| |-dcmi@4c006000/port/name
| |-dcmi@4c006000/clock-names
| |-dcmi@4c006000/status
| |-dcmi@4c006000/interrupts
| |-dcmi@4c006000/dma-names
| |-dcmi@4c006000/phandle
| |-dcmi@4c006000/reg
| |-dcmi@4c006000/pinctrl-0
| |-dcmi@4c006000/dmas
| |-dcmi@4c006000/linux,phandle
| |-dcmi@4c006000/name
| |-dcmi@4c006000/pinctrl-names
| |-pin-controller/dcmi-sleep@0
| |-pin-controller/dcmi-sleep@0/pins
| |-pin-controller/dcmi-sleep@0/pins/pinmux
| |-pin-controller/dcmi-sleep@0/pins/name
| |-pin-controller/dcmi-sleep@0/phandle
| |-pin-controller/dcmi-sleep@0/linux,phandle
```

```
| |-pin-controller/dcmi-sleep@0/name
| |-pin-controller/dcmi@0
| |-pin-controller/dcmi@0/pins
| |-pin-controller/dcmi@0/pins/pinmux
| |-pin-controller/dcmi@0/pins/bias-disable
| |-pin-controller/dcmi@0/pins/name
| |-pin-controller/dcmi@0/phandle
| |-pin-controller/dcmi@0/linux,phandle
| |-pin-controller/dcmi@0/name
|
|-[camera:ovti,ov5640]
  |-i2c@40013000/camera@3c
  |-i2c@40013000/camera@3c/compatible
  |-i2c@40013000/camera@3c/powerdown-gpios
  |-i2c@40013000/camera@3c/DOVDD-supply
  |-i2c@40013000/camera@3c/clocks
  |-i2c@40013000/camera@3c/rotation
  |-i2c@40013000/camera@3c/port
  |-i2c@40013000/camera@3c/port/endpoint
  |-i2c@40013000/camera@3c/port/endpoint/hsync-active
  |-i2c@40013000/camera@3c/port/endpoint/vsync-active
  |-i2c@40013000/camera@3c/port/endpoint/remote-endpoint
  |-i2c@40013000/camera@3c/port/endpoint/bus-width
  |-i2c@40013000/camera@3c/port/endpoint/pclk-sample
  |-i2c@40013000/camera@3c/port/endpoint/phandle
  |-i2c@40013000/camera@3c/port/endpoint/data-shift
  |-i2c@40013000/camera@3c/port/endpoint/linux,phandle
  |-i2c@40013000/camera@3c/port/endpoint/name
  |-i2c@40013000/camera@3c/port/name
  |-i2c@40013000/camera@3c/clock-names
  |-i2c@40013000/camera@3c/status
  |-i2c@40013000/camera@3c/reset-gpios
  |-i2c@40013000/camera@3c/phandle
  |-i2c@40013000/camera@3c/reg
  |-i2c@40013000/camera@3c/linux,phandle
  |-i2c@40013000/camera@3c/name
```

5.2 How to trace

5.2.1 V4L2 userland API tracing

Tracing of V4L2 userland API[2] can be enabled using command:

```
Board $> echo 0x3 >  /sys/devices/platform/soc/*.dcmi/video4linux/video*/dev_debug
```

Traces are output in kernel log buffer:

```
Board $> dmesg
```

```
[10130.641469] video0: VIDIOC_TRY_FMT: type=vid-cap, width=640, height=480,
pixelformat=YUYV, field=none, bytesperline=
1280, sizeimage=614400, colorspace=8, flags=0x0, ycbcr_enc=1, quantization=1, xfer_func=2
[10130.641550] video0: VIDIOC_S_FMT: type=vid-cap, width=640, height=480,
pixelformat=YUYV, field=none, bytesperline=12
80, sizeimage=614400, colorspace=8, flags=0x0, ycbcr_enc=1, quantization=1, xfer_func=2
[10130.641597] video0: VIDIOC_G_PARM: type=vid-cap, capability=0x1000, capturemode=0x0,
timeperframe=1/30, extendedmode
=0, readbuffers=2
```

```
[10130.641638] video0: VIDIOC_G_PARM: type=vid-cap, capability=0x1000, capturemode=0x0,
timeperframe=1/30, extendedmode
=0, readbuffers=2
[10130.641681] video0: VIDIOC_S_PARM: type=vid-cap, capability=0x1000, capturemode=0x0,
timeperframe=1/30, extendedmode
=0, readbuffers=2
[10130.641740] video0: VIDIOC_G_CTRL: error -22: id=0x980927, value=0
[10130.642770] video0: VIDIOC_REQBUFS: count=0, type=vid-cap, memory=mmap
[10130.642819] video0: VIDIOC_CREATE_BUFS: index=0, count=0, memory=mmap, type=vid-cap,
width=640, height=480, pixelfor
mat=YUYV, field=none, bytesperline=1280, sizeimage=614400, colorspace=8, flags=0x0,
ycbcr_enc=1, quantization=1, xfer_f
unc=2
[10130.658541] video0: VIDIOC_G_CTRL: error -22: id=0x980927, value=0
[10130.662770] video0: VIDIOC_REQBUFS: count=3, type=vid-cap, memory=mmap
[10130.662852] video0: VIDIOC_QUERYBUF: 00:00:00.00000000 index=0, type=vid-cap,
flags=0x00002000, field=any, sequence=
0, memory=mmap, bytesused=0, offset/userptr=0x0, length=614400
[10130.662892] timecode=00:00:00 type=0, flags=0x00000000, frames=0, userbits=0x00000000
[10130.662917] video0: VIDIOC_QUERYBUF: 00:00:00.00000000 index=1, type=vid-cap,
flags=0x00002000, field=any, sequence=
0, memory=mmap, bytesused=0, offset/userptr=0x96000, length=614400
[10130.662952] timecode=00:00:00 type=0, flags=0x00000000, frames=0, userbits=0x00000000
[10130.662967] video0: VIDIOC_QUERYBUF: 00:00:00.00000000 index=2, type=vid-cap,
flags=0x00002000, field=any, sequence=
0, memory=mmap, bytesused=0, offset/userptr=0x12c000, length=614400
[10130.663002] timecode=00:00:00 type=0, flags=0x00000000, frames=0, userbits=0x00000000
[10130.666880] video0: VIDIOC_STREAMON: type=vid-cap
[10130.857484] video0: VIDIOC_CREATE_BUFS: index=3, count=1, memory=mmap, type=vid-cap,
width=640, height=480, pixelfor
mat=YUYV, field=none, bytesperline=1280, sizeimage=614400, colorspace=8, flags=0x0,
ycbcr_enc=1, quantization=1, xfer_f
unc=2
[10130.857585] video0: VIDIOC_QUERYBUF: 00:00:00.00000000 index=3, type=vid-cap,
flags=0x00002000, field=any, sequence=
0, memory=mmap, bytesused=0, offset/userptr=0x1c2000, length=614400
[10130.857627] timecode=00:00:00 type=0, flags=0x00000000, frames=0, userbits=0x00000000
[10131.022069] video0: VIDIOC_STREAMOFF: type=vid-cap
```

5.2.2 V4L2 core framework tracing

Tracing of the V4L2 core framework[3] can be enabled using commands:

```
Board $> echo 0x3 > /sys/module/videobuf2_core/parameters/debug
Board $> echo 0x3 > /sys/module/videobuf2_v4l2/parameters/debug
```

Traces are output in kernel log buffer:

```
Board $> dmesg
```

```
[11875.487933] vb2-core: __setup_offsets: buffer 0, plane 0 offset 0x00000000
[11875.501731] vb2-core: __setup_offsets: buffer 1, plane 0 offset 0x001fb000
[11875.514901] vb2-core: __setup_offsets: buffer 2, plane 0 offset 0x003f6000
[11875.532298] vb2-core: __setup_offsets: buffer 3, plane 0 offset 0x005f1000
[11875.540019] vb2-core: __vb2_queue_alloc: allocated 4 buffers, 1 plane(s) each
[11875.563689] vb2_dc_mmap: mapped dma addr 0xf1900000 at 0xb4f05000, size 2076672
[11875.571174] vb2-core: vb2_mmap: buffer 0, plane 0 successfully mapped
[11875.589656] vb2-core: vb2_core_qbuf: qbuf of buffer 0 succeeded
[11875.595684] vb2_dc_mmap: mapped dma addr 0xf1b00000 at 0xb4d0a000, size 2076672
[11875.603062] vb2-core: vb2_mmap: buffer 1, plane 0 successfully mapped
```

```
[11875.609668] vb2-core: vb2_core_qbuf: qbuf of buffer 1 succeeded
[11875.615642] vb2_dc_mmap: mapped dma addr 0xf1d00000 at 0xb4b0f000, size 2076672
[11875.623016] vb2-core: vb2_mmap: buffer 2, plane 0 successfully mapped
[11875.629628] vb2-core: vb2_core_qbuf: qbuf of buffer 2 succeeded
[11875.635617] vb2_dc_mmap: mapped dma addr 0xf1f00000 at 0xb4914000, size 2076672
[11875.642952] vb2-core: vb2_mmap: buffer 3, plane 0 successfully mapped
[11875.649715] vb2-core: vb2_core_qbuf: qbuf of buffer 3 succeeded
[11875.734058] vb2-core: vb2_core_streamon: successful
[11875.961291] vb2-core: vb2_buffer_done: done processing on buffer 0, state: 6
[11875.967036] vb2-core: vb2_core_dqbuf: returning done buffer
[11875.972437] vb2-core: vb2_core_dqbuf: dqbuf of buffer 0, with state 0
[11876.094639] vb2-core: vb2_buffer_done: done processing on buffer 1, state: 6
[11876.100367] vb2-core: vb2_core_dqbuf: returning done buffer
[11876.105788] vb2-core: vb2_core_dqbuf: dqbuf of buffer 1, with state 0
```

5.2.3 DCMI V4L2 kernel driver tracing

DCMI dynamic debug traces[7] can be enabled using command:

```
Board $> echo "module stm32_dcmi +p" > /sys/kernel/debug/dynamic_debug/control
```

Here is an example with a 5Mp jpeg capture:

```
Board $> gst-launch-1.0 v4l2src ! image/jpeg, width=2592, height=1944 ! fakesink
Board $> dmesg
```

```
[12706.715949] stm32-dcmi 4c006000.dcmi: Sensor format set to 0x4001 2592x1944
[12706.721548] stm32-dcmi 4c006000.dcmi: Buffer format set to JPEG 2592x1944
[12707.365947] stm32-dcmi 4c006000.dcmi: Sensor format set to 0x4001 2592x1944
[12707.371551] stm32-dcmi 4c006000.dcmi: Buffer format set to JPEG 2592x1944
[12707.437537] stm32-dcmi 4c006000.dcmi: Sensor format set to 0x4001 2592x1944
[12707.443042] stm32-dcmi 4c006000.dcmi: Buffer format set to JPEG 2592x1944
[12707.459767] stm32-dcmi 4c006000.dcmi: Setup queue, count=4, size=5038848
[12707.518914] stm32-dcmi 4c006000.dcmi: buffer[0] phy=0xf1900000 size=5038848
[12707.526068] stm32-dcmi 4c006000.dcmi: buffer[1] phy=0xf1e00000 size=5038848
[12707.533299] stm32-dcmi 4c006000.dcmi: buffer[2] phy=0xf2300000 size=5038848
[12707.541456] stm32-dcmi 4c006000.dcmi: buffer[3] phy=0xf2800000 size=5038848
[12707.551443] stm32-dcmi 4c006000.dcmi: Start streaming, starting capture
[12707.820885] stm32-dcmi 4c006000.dcmi: buffer[0] done seq=0, bytesused=499936
[12708.087436] stm32-dcmi 4c006000.dcmi: buffer[1] done seq=1, bytesused=447472
[12708.306415] stm32-dcmi 4c006000.dcmi: Stop streaming, errors=0 (overrun=0), buffers=2
[12708.319095] stm32-dcmi 4c006000.dcmi: Start streaming, starting capture
[12708.333571] stm32-dcmi 4c006000.dcmi: Stop streaming, errors=0 (overrun=0), buffers=0
```

5.3 How to debug

5.3.1 Errors

Errors are unconditionnaly traced in kernel log:

```
Board $> dmesg
[   87.233672] stm32-dcmi 4c006000.dcmi: Some errors found while streaming: errors=1
(overrun=1), buffers=24
```

5.3.2 Memory tracking

Frames require large chunks of contiguous memory, they are allocated by V4L2 framework through DMA backend. Those allocations can be traced using:

```
Board $> echo "module dma_contiguous +p" > /sys/kernel/debug/dynamic_debug/control
Board $> echo "module videobuf2_dma_contig +p" > /sys/kernel/debug/dynamic_debug/control
```

Here is the trace after a VGA preview

```
[11311.617688] vb2_dc_mmap: mapped dma addr 0xf1900000 at 0xb3b6a000, size 614400
[11311.617986] vb2_dc_mmap: mapped dma addr 0xf1a00000 at 0xb3ad4000, size 614400
[11311.618071] vb2_dc_mmap: mapped dma addr 0xf1b00000 at 0xb3a3e000, size 614400
[11311.764146] vb2_dc_mmap: mapped dma addr 0xf1c00000 at 0xb307c000, size 614400
```

4 frames of VGA YUV422 frames: 640x480x2=614400 bytes

6 Source code location

6.1 User space

- V4L2 utilities source code
- V4L2 libraries source code
- Yavta source code

6.2 Kernel space

- V4L2 core source code
- stm32-dcmi V4L2 driver source code
- i2c camera sensor V4L2 drivers source code

7 References

- Information about V4L2 Linux kernel framework on wikipedia.
- 2.02.1 Linux Media Infrastructure userspace API » Part I - Video for Linux API
- 3.03.1 Media subsystem kernel internal API » 1. Video4Linux devices
- MB1379 camera daughter board
- STM32MP157x-EV1 Evaluation board CN7 Camera sensor connector
- STM32MP15 evaluation board
- How to use the kernel dynamic debug

Video 4 Linux version 2

Digital Camera Memory Interface

Application programming interface

Inter-Integrated Circuit (Bi-directional 2-wire bus standard for efficient inter-IC control.)

Direct Memory Access

Stable: 07.10.2019 - 08:51 Revision: 07.10.2019 - 08:50

A quality version of this page, approved on *7 October 2019*, was based off this revision.

Contents

1 Overview

v4l2-ctl is a V4L2 utility allowing to control the camera subsystem [1]. It is based on V4L2 Linux kernel interface[2].

2 Usage

```
Board $> v4l2-ctl --help
```

```
General/Common options:
  --all               display all information available
  -C, --get-ctrl=<ctrl>[,<ctrl>...]
                      get the value of the controls [VIDIOC_G_EXT_CTRLS]
  -c, --set-ctrl=<ctrl>=<val>[,<ctrl>=<val>...]
                      set the value of the controls [VIDIOC_S_EXT_CTRLS]
  -D, --info          show driver info [VIDIOC_QUERYCAP]
  -d, --device=<dev> use device <dev> instead of /dev/video0
                      if <dev> starts with a digit, then /dev/video<dev> is used
  -e, --out-device=<dev> use device <dev> for output streams instead of the
                      default device as set with --device
                      if <dev> starts with a digit, then /dev/video<dev> is used
  -h, --help          display this help message
  --help-all          all options
  --help-io           input/output options
  --help-misc         miscellaneous options
  --help-overlay      overlay format options
  --help-sdr          SDR format options
  --help-selection    crop/selection options
  --help-stds         standards and other video timings options
  --help-streaming    streaming options
  --help-tuner        tuner/modulator options
  --help-vbi          VBI format options
  --help-vidcap       video capture format options
  --help-vidout       vidout output format options
  --help-edid         edid handling options
  -k, --concise       be more concise if possible.
  -l, --list-ctrls    display all controls and their values [VIDIOC_QUERYCTRL]
  -L, --list-ctrls-menus
                      display all controls and their menus [VIDIOC_QUERYMENU]
  -r, --subset=<ctrl>[,<offset>,<size>]+
                      the subset of the N-dimensional array to get/set for control <ctrl>,
                      for every dimension an (<offset>, <size>) tuple is given.
  -w, --wrapper       use the libv4l2 wrapper library.
  --list-devices      list all v4l devices
  --log-status        log the board status in the kernel log [VIDIOC_LOG_STATUS]
  --get-priority      query the current access priority [VIDIOC_G_PRIORITY]
  --set-priority=<prio>
                      set the new access priority [VIDIOC_S_PRIORITY]
                      <prio> is 1 (background), 2 (interactive) or 3 (record)
  --silent            only set the result code, do not print any messages
  --sleep=<secs>      sleep <secs>, call QUERYCAP and close the file handle
  --verbose           turn on verbose ioctl status reporting
```

A longer help is also available:

```
Board $> v4l2-ctl --help-all
```

```
General/Common options:
  --all               display all information available
  -C, --get-ctrl=<ctrl>[,<ctrl>...]
                      get the value of the controls [VIDIOC_G_EXT_CTRLS]
  -c, --set-ctrl=<ctrl>=<val>[,<ctrl>=<val>...]
                      set the value of the controls [VIDIOC_S_EXT_CTRLS]
  -D, --info          show driver info [VIDIOC_QUERYCAP]
  -d, --device=<dev> use device <dev> instead of /dev/video0
                      if <dev> starts with a digit, then /dev/video<dev> is used
  -e, --out-device=<dev> use device <dev> for output streams instead of the
                      default device as set with --device
                      if <dev> starts with a digit, then /dev/video<dev> is used
  -h, --help          display this help message
  --help-all          all options
  --help-io           input/output options
  --help-misc         miscellaneous options
  --help-overlay      overlay format options
  --help-sdr          SDR format options
  --help-selection    crop/selection options
  --help-stds         standards and other video timings options
  --help-streaming    streaming options
  --help-tuner        tuner/modulator options
  --help-vbi          VBI format options
  --help-vidcap       video capture format options
  --help-vidout       vidout output format options
  --help-edid         edid handling options
  -k, --concise       be more concise if possible.
  -l, --list-ctrls    display all controls and their values [VIDIOC_QUERYCTRL]
  -L, --list-ctrls-menus
                      display all controls and their menus [VIDIOC_QUERYMENU]
  -r, --subset=<ctrl>[,<offset>,<size>]+
                      the subset of the N-dimensional array to get/set for control <ctrl>,
                      for every dimension an (<offset>, <size>) tuple is given.
  -w, --wrapper       use the libv4l2 wrapper library.
  --list-devices      list all v4l devices
  --log-status        log the board status in the kernel log [VIDIOC_LOG_STATUS]
  --get-priority      query the current access priority [VIDIOC_G_PRIORITY]
  --set-priority=<prio>
                      set the new access priority [VIDIOC_S_PRIORITY]
                      <prio> is 1 (background), 2 (interactive) or 3 (record)
  --silent            only set the result code, do not print any messages
  --sleep=<secs>      sleep <secs>, call QUERYCAP and close the file handle
  --verbose           turn on verbose ioctl status reporting

Tuner/Modulator options:
  -F, --get-freq      query the frequency [VIDIOC_G_FREQUENCY]
  -f, --set-freq=<freq>
                      set the frequency to <freq> MHz [VIDIOC_S_FREQUENCY]
  -T, --get-tuner     query the tuner settings [VIDIOC_G_TUNER]
  -t, --set-tuner=<mode>
                      set the audio mode of the tuner [VIDIOC_S_TUNER]
                      Possible values: mono, stereo, lang2, lang1, bilingual
  --tuner-index=<idx> Use idx as tuner idx for tuner/modulator commands
  --list-freq-bands   display all frequency bands for the tuner/modulator
                      [VIDIOC_ENUM_FREQ_BANDS]
  --get-modulator     query the modulator settings [VIDIOC_G_MODULATOR]
  --set-modulator=<txsubchans>
                      set the sub-carrier modulation [VIDIOC_S_MODULATOR]
                      <txsubchans> is one of:
                      mono:       Modulate as mono
                      mono-rds:   Modulate as mono with RDS (radio only)
                      stereo:     Modulate as stereo
                      stereo-rds: Modulate as stereo with RDS (radio only)
                      bilingual:  Modulate as bilingual
                      mono-sap:   Modulate as mono with Second Audio Program
```

```
                      stereo-sap: Modulate as stereo with Second Audio Program
  --freq-seek=dir=<0/1>,wrap=<0/1>,spacing=<hz>,low=<freq>,high=<freq>
                      perform a hardware frequency seek [VIDIOC_S_HW_FREQ_SEEK]
                      dir is 0 (seek downward) or 1 (seek upward)
                      wrap is 0 (do not wrap around) or 1 (wrap around)
                      spacing sets the seek resolution (use 0 for default)
                      low and high set the low and high seek frequency range in MHz

Input/Output options:
  -I, --get-input     query the video input [VIDIOC_G_INPUT]
  -i, --set-input=<num>
                      set the video input to <num> [VIDIOC_S_INPUT]
  -N, --list-outputs display video outputs [VIDIOC_ENUMOUTPUT]
  -n, --list-inputs   display video inputs [VIDIOC_ENUMINPUT]
  -O, --get-output    query the video output [VIDIOC_G_OUTPUT]
  -o, --set-output=<num>
                      set the video output to <num> [VIDIOC_S_OUTPUT]
  --set-audio-output=<num>
                      set the audio output to <num> [VIDIOC_S_AUDOUT]
  --get-audio-input   query the audio input [VIDIOC_G_AUDIO]
  --set-audio-input=<num>
                      set the audio input to <num> [VIDIOC_S_AUDIO]
  --get-audio-output query the audio output [VIDIOC_G_AUDOUT]
  --set-audio-output=<num>
                      set the audio output to <num> [VIDIOC_S_AUDOUT]
  --list-audio-outputs
                      display audio outputs [VIDIOC_ENUMAUDOUT]
  --list-audio-inputs
                      display audio inputs [VIDIOC_ENUMAUDIO]

Standards/Timings options:
  --list-standards    display supported video standards [VIDIOC_ENUMSTD]
  -S, --get-standard
                      query the video standard [VIDIOC_G_STD]
  -s, --set-standard=<num>
                      set the video standard to <num> [VIDIOC_S_STD]
                      <num> a numerical v4l2_std value, or one of:
                      pal or pal-X (X = B/G/H/N/Nc/I/D/K/M/60) (V4L2_STD_PAL)
                      ntsc or ntsc-X (X = M/J/K) (V4L2_STD_NTSC)
                      secam or secam-X (X = B/G/H/D/K/L/Lc) (V4L2_STD_SECAM)
  --get-detected-standard
                      display detected input video standard [VIDIOC_QUERYSTD]
  --list-dv-timings   list supp. standard dv timings [VIDIOC_ENUM_DV_TIMINGS]
  --set-dv-bt-timings
                      query: use the output of VIDIOC_QUERY_DV_TIMINGS
                      index=<index>: use the index as provided by --list-dv-timings
                      or specify timings using cvt/gtf options as follows:
                      cvt/gtf,width=<width>,height=<height>,fps=<frames per sec>
                      interlaced=<0/1>,reduced-blanking=<0/1/2>,reduced-fps=<0/1>
                      The value of reduced-blanking, if greater than 0, indicates
                      that reduced blanking is to be used and the value indicate the
                      version. For gtf, there is no version 2 for reduced blanking, and
                      the value 1 or 2 will give same results.
                      reduced-fps = 1, slows down pixel clock by factor of 1000 / 1001,
allowing
                      to support NTSC frame rates like 29.97 or 59.94.
                      Reduced fps flag takes effect only with reduced blanking version 2
and,
                      when refresh rate is an integer multiple of 6, say, fps = 24,30,60
etc.
                      or update all or part of the current timings fields:
                      width=<width>,height=<height>,interlaced=<0/1>,
                      polarities=<polarities mask>,pixelclock=<pixelclock Hz>,
                      hfp=<horizontal front porch>,hs=<horizontal sync>,
                      hbp=<horizontal back porch>,vfp=<vertical front porch>,
                      vs=<vertical sync>,vbp=<vertical back porch>,
                      il_vfp=<vertical front porch for bottom field>,
```

```
                     il_vs=<vertical sync for bottom field>,
                     il_vbp=<vertical back porch for bottom field>.
                     clear: start with zeroed timings instead of the current timings.
                     set the digital video timings according to the BT 656/1120
                     standard [VIDIOC_S_DV_TIMINGS]
  --get-dv-timings   get the digital video timings in use [VIDIOC_G_DV_TIMINGS]
  --query-dv-timings query the detected dv timings [VIDIOC_QUERY_DV_TIMINGS]
  --get-dv-timings-cap
                     get the dv timings capabilities [VIDIOC_DV_TIMINGS_CAP]

Video Capture Formats options:
  --list-formats     display supported video formats [VIDIOC_ENUM_FMT]
  --list-formats-ext display supported video formats including frame sizes
                     and intervals
  --list-framesizes=<f>
                     list supported framesizes for pixelformat <f>
                     [VIDIOC_ENUM_FRAMESIZES]
                     pixelformat is the fourcc value as a string
  --list-frameintervals=width=<w>,height=<h>,pixelformat=<f>
                     list supported frame intervals for pixelformat <f> and
                     the given width and height [VIDIOC_ENUM_FRAMEINTERVALS]
                     pixelformat is the fourcc value as a string
  --list-fields      list supported fields for the current format
  -V, --get-fmt-video
                     query the video capture format [VIDIOC_G_FMT]
  -v, --set-fmt-video
  --try-fmt-video=width=<w>,height=<h>,pixelformat=<pf>,field=<f>,colorspace=<c>,
                  xfer=<xf>,ycbcr=<y>,quantization=<q>,premul-alpha,bytesperline=<bpl>
                     set/try the video capture format [VIDIOC_S/TRY_FMT]
                     pixelformat is either the format index as reported by
                       --list-formats, or the fourcc value as a string.
                     The bytesperline option can be used multiple times, once for each
plane.
                     premul-alpha sets V4L2_PIX_FMT_FLAG_PREMUL_ALPHA.
                     <f> can be one of the following field layouts:
                       any, none, top, bottom, interlaced, seq_tb, seq_bt,
                       alternate, interlaced_tb, interlaced_bt
                     <c> can be one of the following colorspaces:
                       smpte170m, smpte240m, rec709, 470m, 470bg, jpeg, srgb,
                       adobergb, bt2020, dcip3
                     <xf> can be one of the following transfer functions:
                       default, 709, srgb, adobergb, smpte240m, smpte2084, dcip3, none
                     <y> can be one of the following Y'CbCr encodings:
                       default, 601, 709, xv601, xv709, bt2020, bt2020c, smpte240m
                     <q> can be one of the following quantization methods:
                       default, full-range, lim-range

Video Output Formats options:
  --list-formats-out display supported video output formats [VIDIOC_ENUM_FMT]
  --list-fields-out  list supported fields for the current output format
  --get-fmt-video-out
                     query the video output format [VIDIOC_G_FMT]
  --set-fmt-video-out
  --try-fmt-video-out=width=<w>,height=<h>,pixelformat=<pf>,field=<f>,colorspace=<c>,
                      xfer=<xf>,ycbcr=<y>,quantization=<q>,premul-alpha,bytesperline=<bpl>
                     set/try the video output format [VIDIOC_S/TRY_FMT]
                     pixelformat is either the format index as reported by
                       --list-formats-out, or the fourcc value as a string.
                     premul-alpha sets V4L2_PIX_FMT_FLAG_PREMUL_ALPHA.
                     The bytesperline option can be used multiple times, once for each
plane.
                     <f> can be one of the following field layouts:
                       any, none, top, bottom, interlaced, seq_tb, seq_bt,
                       alternate, interlaced_tb, interlaced_bt
                     <c> can be one of the following colorspaces:
                       smpte170m, smpte240m, rec709, 470m, 470bg, jpeg, srgb,
                       adobergb, bt2020, dcip3
```

```
                      <xf> can be one of the following transfer functions:
                        default, 709, srgb, adobergb, smpte240m, smpte2084, dcip3, none
                      <y> can be one of the following Y'CbCr encodings:
                        default, 601, 709, xv601, xv709, bt2020, bt2020c, smpte240m
                      <q> can be one of the following quantization methods:
                        default, full-range, lim-range

Video Overlay options:
  --list-formats-overlay
                      display supported overlay formats [VIDIOC_ENUM_FMT]
  --find-fb           find the fb device corresponding with the overlay
  --overlay=<on>      turn overlay on (1) or off (0) (VIDIOC_OVERLAY)
  --get-fmt-overlay   query the video or video output overlay format [VIDIOC_G_FMT]
  --set-fmt-overlay
  --try-fmt-overlay=chromakey=<key>,global_alpha=<alpha>,
                            top=<t>,left=<l>,width=<w>,height=<h>,field=<f>
                      set/try the video or video output overlay format [VIDIOC_TRY/S_FMT]
                      <f> can be one of:
                      any, none, top, bottom, interlaced, seq_tb, seq_bt,
                      alternate, interlaced_tb, interlaced_bt
                      If the width or height changed then the old clip list and bitmap will
                      be invalidated.
  --clear-clips       clear any old clips, to be used in combination with --try/set-fmt-
overlay
  --clear-bitmap      clear any old bitmap, to be used in combination with --try/set-fmt-
overlay
  --add-clip=top=<t>,left=<l>,width=<w>,height=<h>
                      Add an entry to the clip list. May be used multiple times.
                      This clip list will be passed to --try/set-fmt-overlay
  --add-bitmap=top=<t>,left=<l>,width=<w>,height=<h>
                      Set the bits in the given rectangle in the bitmap to 1. May be
                      used multiple times.
                      The bitmap will be passed to --try/set-fmt-overlay
  --get-fbuf          query the overlay framebuffer data [VIDIOC_G_FBUF]
  --set-fbuf=chromakey=<b>,src_chromakey=<b>,global_alpha=<b>,local_alpha=<b>,
local_inv_alpha=<b>,fb=<fb>
                      set the overlay framebuffer [VIDIOC_S_FBUF]
                      <b> is 0 or 1
                      <fb> is the framebuffer device (/dev/fbX)
                      if <fb> starts with a digit, then /dev/fb<fb> is used

VBI Formats options:
  --get-sliced-vbi-cap
                      query the sliced VBI capture capabilities
                      [VIDIOC_G_SLICED_VBI_CAP]
  --get-sliced-vbi-out-cap
                      query the sliced VBI output capabilities
                      [VIDIOC_G_SLICED_VBI_CAP]
  -B, --get-fmt-sliced-vbi
                      query the sliced VBI capture format [VIDIOC_G_FMT]
  --get-fmt-sliced-vbi-out
                      query the sliced VBI output format [VIDIOC_G_FMT]
  -b, --set-fmt-sliced-vbi
  --try-fmt-sliced-vbi
  --set-fmt-sliced-vbi-out
  --try-fmt-sliced-vbi-out=<mode>
                      set/try the sliced VBI capture/output format to <mode>
                      [VIDIOC_S/TRY_FMT], <mode> is a comma separated list of:
                      off:      turn off sliced VBI (cannot be combined with
                                other modes)
                      teletext: teletext (PAL/SECAM)
                      cc:       closed caption (NTSC)
                      wss:      widescreen signal (PAL/SECAM)
                      vps:      VPS (PAL/SECAM)
  --get-fmt-vbi       query the VBI capture format [VIDIOC_G_FMT]
  --get-fmt-vbi-out   query the VBI output format [VIDIOC_G_FMT]
  --set-fmt-vbi
```

```
  --try-fmt-vbi
  --set-fmt-vbi-out
  --try-fmt-vbi-out=samplingrate=<r>,offset=<o>,samplesperline=<spl>,
                    start0=<s0>,count0=<c0>,start1=<s1>,count1=<c1>
                    set/try the raw VBI capture/output format [VIDIOC_S/TRY_FMT]
                    samplingrate: samples per second
                    offset: horizontal offset in samples
                    samplesperline: samples per line
                    start0: start line number of the first field
                    count0: number of lines in the first field
                    start1: start line number of the second field
                    count1: number of lines in the second field

SDR Formats options:
  --list-formats-sdr display supported SDR capture formats [VIDIOC_ENUM_FMT]
  --get-fmt-sdr      query the SDR capture format [VIDIOC_G_FMT]
  --set-fmt-sdr=<f>  set the SDR capture format [VIDIOC_S_FMT]
                     parameter is either the format index as reported by
                     --list-formats-sdr-cap, or the fourcc value as a string
  --try-fmt-sdr=<f>  try the SDR capture format [VIDIOC_TRY_FMT]
                     parameter is either the format index as reported by
                     --list-formats-sdr-cap, or the fourcc value as a string
  --list-formats-sdr-out
                     display supported SDR output formats [VIDIOC_ENUM_FMT]
  --get-fmt-sdr-out  query the SDR output format [VIDIOC_G_FMT]
  --set-fmt-sdr-out=<f>
                     set the SDR output format [VIDIOC_S_FMT]
                     parameter is either the format index as reported by
                     --list-formats-sdr-out, or the fourcc value as a string
  --try-fmt-sdr-out=<f>
                     try the SDR output format [VIDIOC_TRY_FMT]
                     parameter is either the format index as reported by
                     --list-formats-sdr-out, or the fourcc value as a string

Selection/Cropping options:
  --get-cropcap      query the crop capabilities [VIDIOC_CROPCAP]
  --get-crop         query the video capture crop window [VIDIOC_G_CROP]
  --set-crop=top=<x>,left=<y>,width=<w>,height=<h>
                     set the video capture crop window [VIDIOC_S_CROP]
  --get-cropcap-output
                     query crop capabilities for video output [VIDIOC_CROPCAP]
  --get-crop-output  query the video output crop window [VIDIOC_G_CROP]
  --set-crop-output=top=<x>,left=<y>,width=<w>,height=<h>
                     set the video output crop window [VIDIOC_S_CROP]
  --get-cropcap-overlay
                     query crop capabilities for video overlay [VIDIOC_CROPCAP]
  --get-crop-overlay query the video overlay crop window [VIDIOC_G_CROP]
  --set-crop-overlay=top=<x>,left=<y>,width=<w>,height=<h>
                     set the video overlay crop window [VIDIOC_S_CROP]
  --get-cropcap-output-overlay
                     query the crop capabilities for video output overlays
                     [VIDIOC_CROPCAP]
  --get-crop-output-overlay
                     query the video output overlay crop window [VIDIOC_G_CROP]
  --set-crop-output-overlay=top=<x>,left=<y>,width=<w>,height=<h>
                     set the video output overlay crop window [VIDIOC_S_CROP]
  --get-selection=target=<target>
                     query the video capture selection rectangle [VIDIOC_G_SELECTION]
                     See --set-selection command for the valid <target> values.
  --set-selection=target=<target>,flags=<flags>,top=<x>,left=<y>,width=<w>,height=<h>
                     set the video capture selection rectangle [VIDIOC_S_SELECTION]
                     target=crop|crop_bounds|crop_default|compose|compose_bounds|
                            compose_default|compose_padded|native_size
                     flags=le|ge
  --get-selection-output=target=<target>
                     query the video output selection rectangle [VIDIOC_G_SELECTION]
                     See --set-selection command for the valid <target> values.
```

```
  --set-selection-output=target=<target>,flags=<flags>,top=<x>,left=<y>,width=<w>,
height=<h>
                      set the video output selection rectangle [VIDIOC_S_SELECTION]
                      See --set-selection command for the arguments.

Miscellaneous options:
  --wait-for-event=<event>
                      wait for an event [VIDIOC_DQEVENT]
                      <event> is the event number or one of:
                      eos, vsync, ctrl=<id>, frame_sync, source_change=<pad>,
                      motion_det
                      where <id> is the name of the control
                      and where <pad> is the index of the pad or input
  --poll-for-event=<event>
                      poll for an event [VIDIOC_DQEVENT]
                      see --wait-for-event for possible events
  -P, --get-parm      display video parameters [VIDIOC_G_PARM]
  -p, --set-parm=<fps>
                      set video framerate in <fps> [VIDIOC_S_PARM]
  --get-output-parm   display output video parameters [VIDIOC_G_PARM]
  --set-output-parm=<fps>
                      set output video framerate in <fps> [VIDIOC_S_PARM]
  --get-jpeg-comp     query the JPEG compression [VIDIOC_G_JPEGCOMP]
  --set-jpeg-comp=quality=<q>,markers=<markers>,comment=<c>,app<n>=<a>
                      set the JPEG compression [VIDIOC_S_JPEGCOMP]
                      <n> is the app segment: 0-9/a-f, <a> is the actual string.
                      <markers> is a colon separated list of:
                      dht:      Define Huffman Tables
                      dqt:      Define Quantization Tables
                      dri:      Define Restart Interval
  --encoder-cmd=cmd=<cmd>,flags=<flags>
                      Send a command to the encoder [VIDIOC_ENCODER_CMD]
                      cmd=start|stop|pause|resume
                      flags=stop_at_gop_end
  --try-encoder-cmd=cmd=<cmd>,flags=<flags>
                      Try an encoder command [VIDIOC_TRY_ENCODER_CMD]
                      See --encoder-cmd for the arguments.
  --decoder-cmd=cmd=<cmd>,flags=<flags>,stop_pts=<pts>,start_speed=<speed>,
                      start_format=<none|gop>
                      Send a command to the decoder [VIDIOC_DECODER_CMD]
                      cmd=start|stop|pause|resume
                      flags=start_mute_audio|pause_to_black|stop_to_black|
                            stop_immediately
  --try-decoder-cmd=cmd=<cmd>,flags=<flags>
                      Try a decoder command [VIDIOC_TRY_DECODER_CMD]
                      See --decoder-cmd for the arguments.

Video Streaming options:
  --stream-count=<count>
                      stream <count> buffers. The default is to keep streaming
                      forever. This count does not include the number of initial
                      skipped buffers as is passed by --stream-skip.
  --stream-skip=<count>
                      skip the first <count> buffers. The default is 0.
  --stream-sleep=<count>
                      sleep for 1 second every <count> buffers. If <count> is 0,
                      then sleep forever right after streaming starts. The default
                      is -1 (never sleep).
  --stream-to=<file> stream to this file. The default is to discard the
                      data. If <file> is '-', then the data is written to stdout
                      and the --silent option is turned on automatically.
  --stream-to-host=<hostname[:port]> stream to this host. The default port is 8362.
  --stream-poll       use non-blocking mode and select() to stream.
  --stream-mmap=<count>
                      capture video using mmap() [VIDIOC_(D)QBUF]
                      count: the number of buffers to allocate. The default is 3.
  --stream-user=<count>
```

```
                      capture video using user pointers [VIDIOC_(D)QBUF]
                      count: the number of buffers to allocate. The default is 3.
  --stream-dmabuf     capture video using dmabuf [VIDIOC_(D)QBUF]
                      Requires a corresponding --stream-out-mmap option.
  --stream-from=<file> stream from this file. The default is to generate a pattern.
                      If <file> is '-', then the data is read from stdin.
  --stream-from-host=<hostname[:port]> stream from this host. The default port is 8362.
  --stream-loop       loop when the end of the file we are streaming from is reached.
                      The default is to stop.
  --stream-out-pattern=<count>
                      choose output test pattern. The default is 0.
  --stream-out-square
                      show a square in the middle of the output test pattern.
  --stream-out-border
                      show a border around the pillar/letterboxed video.
  --stream-out-sav    insert an SAV code in every line.
  --stream-out-eav    insert an EAV code in every line.
  --stream-out-pixel-aspect=<aspect
                      select a pixel aspect ratio. The default is to autodetect.
                      <aspect> can be one of: square, ntsc, pal
  --stream-out-video-aspect=<aspect
                      select a video aspect ratio. The default is to use the frame ratio.
                      <aspect> can be one of: 4x3, 14x9, 16x9, anamorphic
  --stream-out-alpha=<alpha-value>
                      value to use for the alpha component, range 0-255. The default is 0.
  --stream-out-alpha-red-only
                      only use the --stream-out-alpha value for the red colors,
                      for all others use 0.
  --stream-out-rgb-lim-range
                      Encode RGB values as limited [16-235] instead of full range.
  --stream-out-hor-speed=<speed>
                      choose speed for horizontal movement. The default is 0,
                      and the range is [-3...3].
  --stream-out-vert-speed=<speed>
                      choose speed for vertical movement. The default is 0,
                      and the range is [-3...3].
  --stream-out-perc-fill=<percentage>
                      percentage of the frame to actually fill. The default is 100%.
  --stream-out-mmap=<count>
                      output video using mmap() [VIDIOC_(D)QBUF]
                      count: the number of buffers to allocate. The default is 4.
  --stream-out-user=<count>
                      output video using user pointers [VIDIOC_(D)QBUF]
                      count: the number of buffers to allocate. The default is 4.
  --stream-out-dmabuf
                      output video using dmabuf [VIDIOC_(D)QBUF]
                      Requires a corresponding --stream-mmap option.
  --list-patterns     list available patterns for use with --stream-pattern.
  --list-buffers      list all video buffers [VIDIOC_QUERYBUF]
  --list-buffers-out list all video output buffers [VIDIOC_QUERYBUF]
  --list-buffers-vbi list all VBI buffers [VIDIOC_QUERYBUF]
  --list-buffers-vbi-out
                      list all VBI output buffers [VIDIOC_QUERYBUF]
  --list-buffers-sliced-vbi
                      list all sliced VBI buffers [VIDIOC_QUERYBUF]
  --list-buffers-sliced-vbi-out
                      list all sliced VBI output buffers [VIDIOC_QUERYBUF]
  --list-buffers-sdr
                      list all SDR RX buffers [VIDIOC_QUERYBUF]
  --list-buffers-sdr-out
                      list all SDR TX buffers [VIDIOC_QUERYBUF]

EDID options:
  --set-edid=pad=<pad>[,type=<type>|file=<file>][,format=<fmt>][modifiers]
                      <pad> is the input or output index for which to set the EDID.
                      <type> can be 'hdmi', 'hdmi-4k-170mhz', 'hdmi-4k-300mhz', 'hdmi-4k-
600mhz',
```

```
                    'dvid' or 'vga'. A predefined EDID suitable for that connector type
will be
                    set. It has a 1920x1080p60 native resolution for the non-4k variants
and a
                    3840x2160 resolution for the 4k variants (4kp30 YCbCr 4:2:0 for
170mhz, 4kp30
                    for 300 mhz and 4kp60 for 600 mhz).
                    If <file> is '-', then the data is read from stdin, otherwise it is
                    read from the given file. The file format must be in hex as in get-
edid.
                    The 'type' or 'file' arguments are mutually exclusive. One of the two
                    must be specified.
                    <fmt> is one of:
                    hex:    hex numbers in ascii text (default)
                    raw:    raw binary EDID content

                    [modifiers] is a comma-separate list of EDID modifiers:

                    CEA-861 Header modifiers:
                    underscan: toggle the underscan bit.
                    audio: toggle the audio bit.
                    ycbcr444: toggle the YCbCr 4:4:4 bit.
                    ycbcr422: toggle the YCbCr 4:2:2 bit.

                    Speaker Allocation Data Block modifiers:
                    fl-fr: Front Left and Front Right.
                    lfe: Low Frequence Effects.
                    fc: Front Center.
                    rl-rr: Rear Left and Rear Right.
                    rc: Rear Center.
                    flc-frc: Front Left Center and Front Right Center.
                    rlc-rrc: Rear Left Center and Rear Right Center.
                    flw-frw: Front Left Wide and Front Right Wide.
                    flh-frh: Front Left High and Front Right High.
                    tc: Top Center.
                    fch: Front Center High.

                    HDMI Vendor-Specific Data Block modifiers:
                    pa=<pa>: change the physical address.
                    y444: toggle the YCbCr 4:4:4 Deep Color bit.
                    30-bit: toggle the 30 bits/pixel bit.
                    36-bit: toggle the 36 bits/pixel bit.
                    48-bit: toggle the 48 bits/pixel bit.
                    graphics: toggle the Graphics Content Type bit.
                    photo: toggle the Photo Content Type bit.
                    cinema: toggle the Cinema Content Type bit.
                    game: toggle the Game Content Type bit.

                    HDMI Forum Vendor-Specific Data Block modifiers:
                    scdc: toggle the SCDC Present bit.

                    CEA-861 Video Capability Descriptor modifiers:
                    qy: toggle the QY YCC Quantization Range bit.
                    qs: toggle the QS RGB Quantization Range bit.
                    s-pt=<0-3>: set the PT Preferred Format Over/underscan bits.
                    s-it=<0-3>: set the IT Over/underscan bits.
                    s-ce=<0-3>: set the CE Over/underscan bits.

                    CEA-861 Colorimetry Data Block modifiers:
                    xvycc-601: toggle the xvYCC 601 bit.
                    xvycc-709: toggle the xvYCC 709 bit.
                    sycc: toggle the sYCC 601 bit.
                    adobe-ycc: toggle the Adobe YCC 601 bit.
                    adobe-rgb: toggle the Adobe RGB bit.
                    bt2020-rgb: toggle the BT2020 RGB bit.
                    bt2020-ycc: toggle the BT2020 YCC bit.
                    bt2020-cycc: toggle the BT2020 cYCC bit.
```

```
                    CEA-861 HDR Static Metadata Data Block modifiers:
                    sdr: toggle the Traditional gamma SDR bit.
                    hdr: toggle the Traditional gamma HDR bit.
                    smpte2084: toggle the SMPTE ST 2084 bit.
  --clear-edid=<pad>
                    <pad> is the input or output index for which to clear the EDID.
  --info-edid=<pad>  print the current EDID's modifiers
                    <pad> is the input or output index for which to get the EDID.
  --get-edid=pad=<pad>,startblock=<startblock>,blocks=<blocks>,format=<fmt>,file=<file>
                    <pad> is the input or output index for which to get the EDID.
                    <startblock> is the first block number you want to read. Default 0.
                    <blocks> is the number of blocks you want to read. Default is
                    all blocks.
                    <fmt> is one of:
                    hex:    hex numbers in ascii text (default)
                    raw:    can be piped directly into the edid-decode tool
                    carray: c-program struct
                    If <file> is '-' or not the 'file' argument is not supplied, then
the data
                    is written to stdout.
  --fix-edid-checksums
                    If specified then any checksum errors will be fixed silently.
```

3 Examples

Refer to V4L2 camera overview for examples of usage.

4 References

- V4L2 utilities
- Information about V4L2 Linux kernel framework on wikipedia

Video 4 Linux version 2

Linux® is a registered trademark of Linus Torvalds.

BlueTooth

Receive

Transmit

Extended Display Identification Data (HDMI standard)

High-Definition Multimedia Interface (HDMI standard)

High Dynamic Range (HDMI standard)

Stable: 07.10.2019 - 08:36 / Revision: 07.10.2019 - 08:34

A quality version of this page, approved on *7 October 2019*, was based off this revision.

1 Overview

Yavta stands for "Yet Another V4L2 Test Application". This is a test application based on V4L2 Linux® kernel interface that allows testing, debugging and controlling camera subsystem.

Warning

Yavta is only available in STM32MP1 Distribution Package weston image

2 How to use Yavta

```
/usr/sbin # yavta --help
Usage: yavta [options] device
Supported options:
-c, --capture[=nframes]         Capture frames
-C, --check-overrun             Verify dequeued frames for buffer overrun
-d, --delay                     Delay (in ms) before requeuing buffers
-D, --display                   Display frames
-f, --format format             Set the video format
-F, --file[=name]               Read/write frames from/to disk
        For video capture devices, the first '#' character in the file name is
        expanded to the frame sequence number. The default file name is
        'frame-#.bin'.
-h, --help                      Show this help screen
-i, --input input               Select the video input
-I, --fill-frames               Fill frames with check pattern before queuing them
-l, --list-controls             List available controls
-n, --nbufs n                   Set the number of video buffers
-p, --pause                     Pause before starting the video stream
-q, --quality n                 MJPEG quality (0-100)
-r, --get-control ctrl          Get control 'ctrl'
-R, --realtime=[priority]       Enable realtime RR scheduling
-s, --size WxH                  Set the frame size
-t, --time-per-frame num/denom  Set the time per frame (eg. 1/25 = 25 fps)
-u, --userptr                   Use the user pointers streaming method
-w, --set-control 'ctrl value'  Set control 'ctrl' to 'value'
    --enum-formats              Enumerate formats
    --enum-inputs               Enumerate inputs
    --no-query                  Don't query capabilities on open
    --offset                    User pointer buffer offset from page start
    --requeue-last              Requeue the last buffers before streamoff
    --skip n                    Skip the first n frames
    --sleep-forever             Sleep forever after configuring the device
    --stride value              Line stride in bytes
```

3 Examples

- Getting supported controls/formats/resolutions:

```
yavta -l --enum-formats --enum-inputs /dev/video0
```

```
Device /dev/video0 opened.
Device `STM32 Camera Memory Interface' on `platform:dcmi' is a video capture device.
--- Image Processing Controls (class 0x009f0001) ---
control 0x009f0903 `Test Pattern' min 0 max 1 step 1 default 0 current 1.
1 control found.
- Available formats:
        Format 0: RGB565 (50424752)
        Type: Video capture (1)
        Name: 16-bit RGB 5-6-5
        Frame size: 640x480 ()
        Frame size: 320x240 ()
        Frame size: 160x120 ()

- Available inputs:
        Input 0: Camera.

Video format: RGB565 (50424752) 160x120 (stride 320) buffer size 38400
```

- Capturing 10 frames on the disk (default format/resolution):

```
yavta -F /dev/video0 --capture=10
```

```
/usr/sbin # yavta -F /dev/video0 --capture=10
Device /dev/video0 opened.
Device `STM32 Camera Memory Interface' on `platform:dcmi' is a video capture device.
Video format: RGB565 (50424752) 160x120 (stride 320) buffer size 38400
8 buffers requested.
length: 38400 offset: 0
Buffer 0 mapped at address 0xc0710000.
length: 38400 offset: 40960
Buffer 1 mapped at address 0xc0720000.
length: 38400 offset: 81920
Buffer 2 mapped at address 0xc0730000.
length: 38400 offset: 122880
Buffer 3 mapped at address 0xc0740000.
length: 38400 offset: 163840
Buffer 4 mapped at address 0xc0750000.
length: 38400 offset: 204800
Buffer 5 mapped at address 0xc0760000.
length: 38400 offset: 245760
Buffer 6 mapped at address 0xc0770000.
length: 38400 offset: 286720
Buffer 7 mapped at address 0xc0780000.
0 (0) [-] 0 38400 bytes 1378.374325 1378.374535 17.819 fps
1 (1) [-] 1 38400 bytes 1378.407623 1378.407822 30.032 fps
2 (2) [-] 2 38400 bytes 1378.440939 1378.441111 30.016 fps
3 (3) [-] 3 38400 bytes 1378.474288 1378.474450 29.986 fps
4 (4) [-] 4 38400 bytes 1378.507624 1378.507827 29.998 fps
5 (5) [-] 5 38400 bytes 1378.540989 1378.541196 29.972 fps
```

```
6 (6) [-] 6 38400 bytes 1378.574288 1378.574489 30.031 fps
7 (7) [-] 7 38400 bytes 1378.607624 1378.607827 29.998 fps
8 (0) [-] 8 38400 bytes 1378.640959 1378.641162 29.999 fps
9 (1) [-] 9 38400 bytes 1378.674288 1378.674488 30.004 fps
Captured 10 frames in 0.356282 seconds (28.067590 fps, 1077795.461669 B/s).
```

Video 4 Linux version 2