

OMAP 3 Image Signal Processor (ISP) driver

Copyright © 2010 Nokia Corporation

Copyright © 2009 Texas Instruments, Inc.

Contacts: Laurent Pinchart <laurent.pinchart@ideasonboard.com>, Sakari Ailus <sakari.ailus@iki.fi>, David Cohen <dacohen@gmail.com>

Events

The OMAP 3 ISP driver does support the V4L2 event interface on CCDC and statistics (AEWB, AF and histogram) subdevs.

The CCDC subdev produces V4L2_EVENT_FRAME_SYNC type event on HS_VS interrupt which is used to signal frame start. Earlier version of this driver used V4L2_EVENT_OMAP3ISP_HS_VS for this purpose. The event is triggered exactly when the reception of the first line of the frame starts in the CCDC module. The event can be subscribed on the CCDC subdev.

(When using parallel interface one must pay account to correct configuration of the VS signal polarity. This is automatically correct when using the serial receivers.)

Each of the statistics subdevs is able to produce events. An event is generated whenever a statistics buffer can be dequeued by a user space application using the VIDIOC_OMAP3ISP_STAT_REQ IOCTL. The events available are:

- V4L2_EVENT_OMAP3ISP_AEWB
- V4L2_EVENT_OMAP3ISP_AF
- V4L2_EVENT_OMAP3ISP_HIST

The type of the event data is struct omap3isp_stat_event_status for these ioctls. If there is an error calculating the statistics, there will be an event as usual, but no related statistics buffer. In this case omap3isp_stat_event_status.buf_err is set to non-zero.

Private IOCTLs

The OMAP 3 ISP driver supports standard V4L2 IOCTLs and controls where possible and practical. Much of the functions provided by the ISP, however, does not fall under the standard IOCTLs --- gamma tables and configuration of statistics collection are examples of such.

In general, there is a private ioctl for configuring each of the blocks containing hardware-dependent functions.

The following private IOCTLs are supported:

- VIDIOC_OMAP3ISP_CCDC_CFG
- VIDIOC_OMAP3ISP_PRV_CFG
- VIDIOC_OMAP3ISP_AEWB_CFG
- VIDIOC_OMAP3ISP_HIST_CFG
- VIDIOC_OMAP3ISP_AF_CFG
- VIDIOC_OMAP3ISP_STAT_REQ
- VIDIOC_OMAP3ISP_STAT_EN

The parameter structures used by these ioctls are described in include/linux/omap3isp.h. The detailed functions of the ISP itself related to a given ISP block is described in the Technical Reference Manuals (TRMs) --- see the end of the document for those.

While it is possible to use the ISP driver without any use of these private IOCTLs it is not possible to obtain optimal image quality this way. The AEWB, AF and histogram modules cannot be used without configuring them using the appropriate private IOCTLs.

CCDC and preview block IOCTLs

The VIDIOC_OMAP3ISP_CCDC_CFG and VIDIOC_OMAP3ISP_PRV_CFG IOCTLs are used to configure, enable and disable functions in the CCDC and preview blocks, respectively. Both IOCTLs control several functions in the blocks they control. VIDIOC_OMAP3ISP_CCDC_CFG IOCTL accepts a pointer to struct omap3isp_ccdc_update_config as its argument. Similarly VIDIOC_OMAP3ISP_PRV_CFG accepts a pointer to struct omap3isp_prev_update_config. The definition of both structures is available in [1].

The update field in the structures tells whether to update the configuration for the specific function and the flag tells whether to enable or disable the function.

The update and flag bit masks accept the following values. Each separate functions in the CCDC and preview blocks is associated with a flag (either disable or enable; part of the flag field in the structure) and a pointer to configuration data for the function.

Valid values for the update and flag fields are listed here for VIDIOC_OMAP3ISP_CCDC_CFG. Values may be or'ed to configure more than one function in the same IOCTL call.

- OMAP3ISP_CCDC_ALAW

- OMAP3ISP_CCDC_LPF
- OMAP3ISP_CCDC_BLCLAMP
- OMAP3ISP_CCDC_BCOMP
- OMAP3ISP_CCDC_FPC
- OMAP3ISP_CCDC_CULL
- OMAP3ISP_CCDC_CONFIG_LSC
- OMAP3ISP_CCDC_TBL_LSC

The corresponding values for the VIDIOC_OMAP3ISP_PRV_CFG are here:

- OMAP3ISP_PREV_LUMAENH
- OMAP3ISP_PREV_INVALAW
- OMAP3ISP_PREV_HRZ_MED
- OMAP3ISP_PREV_CFA
- OMAP3ISP_PREV_CHROMA_SUPP
- OMAP3ISP_PREV_WB
- OMAP3ISP_PREV_BLKADJ
- OMAP3ISP_PREV_RGB2RGB
- OMAP3ISP_PREV_COLOR_CONV
- OMAP3ISP_PREV_YC_LIMIT
- OMAP3ISP_PREV_DEFECT_COR
- OMAP3ISP_PREV_GAMMABYPASS
- OMAP3ISP_PREV_DRK_FRM_CAPTURE
- OMAP3ISP_PREV_DRK_FRM_SUBTRACT
- OMAP3ISP_PREV_LENS_SHADING
- OMAP3ISP_PREV_NF
- OMAP3ISP_PREV_GAMMA

The associated configuration pointer for the function may not be NULL when enabling the function. When disabling a function the configuration pointer is ignored.

Statistic blocks IOCTLs

The statistics subdevs do offer more dynamic configuration options than the other subdevs. They can be enabled, disabled and reconfigured when the pipeline is in streaming state.

The statistics blocks always get the input image data from the CCDC (as the histogram memory read isn't implemented). The statistics are dequeuable by the user from the statistics subdev nodes using private IOCTLs.

The private IOCTLs offered by the AEWB, AF and histogram subdevs are heavily reflected by the register level interface offered by the ISP hardware. There are aspects that are purely related to the driver implementation and these are discussed next.

VIDIOC_OMAP3ISP_STAT_EN

This private IOCTL enables/disables a statistic module. If this request is done before streaming, it will take effect as soon as the pipeline starts to stream. If the pipeline is already streaming, it will take effect as soon as the CCDC becomes idle.

VIDIOC_OMAP3ISP_AEWB_CFG, VIDIOC_OMAP3ISP_HIST_CFG and VIDIOC_OMAP3ISP_AF_CFG

Those IOCTLs are used to configure the modules. They require user applications to have an in-depth knowledge of the hardware. Most of the fields explanation can be found on OMAP's TRMs. The two following fields common to all the above configure private IOCTLs require explanation for better understanding as they are not part of the TRM.

omap3isp_[h3a_af/h3a_aewb/hist]_config.buf_size:

The modules handle their buffers internally. The necessary buffer size for the module's data output depends on the requested configuration. Although the driver supports reconfiguration while streaming, it does not support a reconfiguration which requires bigger buffer size than what is already internally allocated if the module is enabled. It will return -EBUSY on this case. In order to avoid such condition, either disable/reconfigure/enable the module or request the necessary buffer size during the first configuration while the module is disabled.

The internal buffer size allocation considers the requested configuration's minimum buffer size and the value set on buf_size field. If buf_size field is out of [minimum, maximum] buffer size range, it's clamped to fit in there. The driver then selects the biggest value. The corrected buf_size value is written back to user application.

omap3isp_[h3a_af/h3a_aewb/hist]_config.config_counter:

As the configuration doesn't take effect synchronously to the request, the driver must provide a way to track this information to provide more accurate data. After a configuration is requested, the config_counter returned to user space application will be an

unique value associated to that request. When user application receives an event for buffer availability or when a new buffer is requested, this `config_counter` is used to match a buffer data and a configuration.

VIDIOC_OMAP3ISP_STAT_REQ

Send to user space the oldest data available in the internal buffer queue and discards such buffer afterwards. The field `omap3isp_stat_data.frame_number` matches with the video buffer's `field_count`.

References

| [\[1\]](#) `include/linux/omap3isp.h`