Generic Camera Driver Software User Guide

ABSTRACT:

This is the Software User Guide Document for Generic Camera Driver for Linux OS.

KEYWORDS:

User Guide

APPROVED:

Revision History

VERSION	DATE	AUTHOR	CHANGE DESCRIPTION
0.1	2-August-16	Tomas Babinec	First draft
1.0	7-February-17	Cristian Tomescu	Update for RTM
1.1	27-June-17	Cristian Tomescu	OV10640 MIPI driver updates
1.2	24-July-18	Loc Nguyen Khang Ba Tran	Crucible changed: CR-VS-284 Add OV9716 driver Use low level driver command instead of ioctl command Updated for VSDK 1.2 RTM release
1.3	04-December-18	Loc Nguyen	Add new APIs: `CAM_Reset/CAM_PowerControl`

Table of Contents

1 Introduction

The purpose of this document is to describe Generic camera driver. It is intended to serve as a reference source during the development of VSDK based application.

1.1 Purpose

The purpose of this document is to define Generic camera driver internal behavior and user space interface. It is intended to serve as a reference source during the driver implementation and future use. For exact definitions and implementation details please check references and the source code.

1.2 Audience Description

This document is intended for internal use by S23V234 Vision SDK developers.

1.3 References

Id	***Title***	***Location***
[1]	*SDI SW User Guide*	*Vision sdk git, folder:* s32v234_sdk\docs\drivers
[2]	*S32v234 Reference Manual*	Sharepoint

Table 1: References

1.4 Definitions, Acronyms and Abbreviations

Term/Acronym	***Description***
API	*Application Programming Interface*
CGD	*Camera Generic driver*
HW	*Hardware*
ISP	*Image signal processor (whole image processing system)*
SDI	*Sensor Data Interface library*
SW	*Software*

Table 2: Acronyms

1.5 Document Location

This document is available in VisionSDK directory structure at the following location:

`s32v234_sdk/docs/drivers/CGD_Driver_User_Guide.pdf`

2 General Description

The cameras connected using MipiCsi (serial) or Viu (parallel) interfaces or intermediate HW like serializer/deserilizer sets are being configured using I2C bus.

The aim of the Generic camera driver is to provide a user space level API for the commonly required I2C functionality including device client management and register read/write access. It manage at kernel level the I2C hardware.

On top of the I2C API the Generic camera driver also implements the specific configuration and control for every type of camera supported by VSDK. Each type of camera has a specific set of features and registers to be configured, which makes it impractical to create a common configuration interface.

3 Functional Description

The Generic camera driver SW has 3 layers (see Figure 1). The first layer operates in kernel space and implements the minimum required I2C device interaction. Internal behavior of the kernel space layer will be described in detail in section 3.3.

The second layer is implemented as a user space abstraction layer for the low-level kernel driver API. On top of the second layer a specific sensor device code is implemented. This third layer accomplishes most of the driver functionality. Together the second and third layer are designated as Generic Camera user library. The provided user level API is explained in section 3.4.

At the moment the following sensors are supported by the Generic camera driver:

- Sony IMX224MQV on MipiCsi-2,
- Omnivision Ov10640 VIU/MipiCsi-2,
- Omnivision Ov10635 camera connected to VIU interface,
- Omnivision Ov9716 on MipiCsi,
- Maxim Serializer/Deserializer HW setup with 4 OV10640 cameras,
- TI Serializer/Deserializer HW.

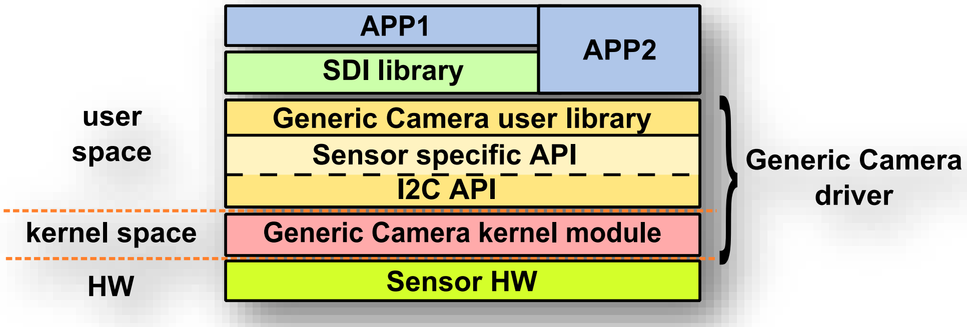

Figure 1: Generic camera driver software layout

3.1 Data types – generic

The Camera generic driver introduces the following data types and containers (see [1] for full definitions):

- Enum `I2C_BUS_IDX`:
  Enumerates possible indexes of the I2C units
- Enums `I2C_REG_ADDR_WIDTH`:
  Enumerates supported device register address widths (8 or 16 bit wide).
- Enums `GPIO_VALUE`:
  Enumerates possible indexes of GPIO value
- Enums `GPIO_PIN_CTRL_NUM`:

Enumerates supported GPIO pin

- Enums SENSOR_TYPE:
  Enumerates supported camera type
- Structure `CamI2cCfg`:
  Describes I2C related parameters of the particular camera device.
- Structure `CamGeneric`:
  Describes particular camera device parameters. Includes `CamI2cCfg` structure.
- Typedef `CamI2cClient_t`:
  Defines a pointer type (the I2C client handle).
- Structure `I2cXfrSingle`:
  Defines one I2C write transaction command.
- Structure `I2cXfrWriteCmd`:
  Defines one I2C read transaction command.
- Structure `I2cXfrWriteBatch`:
  Defines sequence of I2C write commands that should be executed as a batch.
- Structure `I2cXfrReadBatch`:
  Defines sequence of I2C read command that should be executed as a batch.
- Structure `GpioControlCmd`:
  Defines struct when set or clear GPIO pin.
- Structure CamResetPinCmd:
  Define struct to reset camera via GPIO pin.
- Structure CamPowerControlCmd:
  Define struct to control camera power via GPIO pin.

3.2 Data types – camera specific

Sony IMX224MQV:

- Structure `SONY_Geometry`:
  Describes image geometry, frame frequency and number of exposures per frame (single/dual).
- Structure `SONY_Exposure`:
  Describes configuration of exposure control parameters.
- Enum `SONY_ExpState`:
  Current state of the exposure update state machine.
- Structure `SONY_ChGain`:
  Channel gain configuration per channel.

Omnivision Ov10640:

- Structure `OV10640_Geometry`:

  Describes image geometry, frame frequency, exposure mode and embedded data configuration.

- Structure `OV10640_Exposure`:

  Describes configuration of exposure control parameters.

- Enum `OV10640_ExpTime`:

  Exposure type classified by duration.

- Enum `OV10640_ColorCh_t`:

  Exposure colors channel.

- Structure `OV10640_ChGain`:

  Channel gain configuration per channel.

Maxim serialiser/deserialiser:

- Enum `MAXIM_CHNL`:
  Enumerates the channel indexes on the MAXIM serializer/deserializer board.

- Structure `MAXIM_Cfg`:
  Describes Maxim HW setup configuration (CSI interface, sensor type, deserializer slots used).

Omnivision Ov10635:

- Structure `OV10635_Geometry`:

  Describes image geometry, frame frequency, exposure mode and embedded data configuration.

- Structure `OV10635_Exposure`:

  Describes configuration of exposure control parameters.

- Structure `OV10635_ChGain`:

  Channel gain configuration per channel.

- Enum `OV10635_ExpState`:

  Exposure status.

Omnivision Ov9716:

- Structure OV9716_Geometry:

  Describes image geometry, frame frequency, exposure mode and embedded data configuration.

- Enum OV9716_ExposureType:

  Exposure type classified by duration.
- Structure OV9716_Exposure:

  Describes configuration of exposure control parameters.
- Structure OV9716_ChGain:

  Channel gain configuration per channel.

TI UB964:

- Structure UB964RxPort_Cfg_t:

  Describes RX port specific config.
- Structure TIUB964_Cfg_t:

  Describes master tiub964 configuration.
- Enum I2C_PORT_964:

  Enumerates possible indexes in tiub964
- Enum FPD3_PORT_SEL:

  Enumerates RX port select register.
- Enum CSI_PORT_SEL_964:

  Enumerates select CSI2 TX module.
- Enum CSI_LANES_964:

  Enumerates lane configuration.
- Enum CSI_DT_964:

  Enumerates MIPI-CSI2 data type.
- Enum CSI_VT_964:

  Enumerates virtual channel id.
- Enum FV_POLARITY_964:

  Enumerates FV definition
- Enum BC_GPIO_CTL:

  Enumerates BC GPIO control

3.3 Kernel Space

3.3.1 API functions

This section, Table 3, describes functionality exported by the Generic camera driver module. It is intended to be used by upper layer SW such as IO control interface creation in case of Linux environment or directly by the user library in case of a standalone setup.

In Linux environment, the Generic camera driver is associated with special device file `cam`. In some case the driver API functions don’t have exact LLDCMD control counterpart. An example of this is the batch I2C commands. The I2C batch commands were introduced to reduce the number of switches between user and kernel space in Linux. However, it was empirically tested that this approach brings a rather negligible performance improvement.

Function, LLD command	Description
`CAM_ClientAdd` `LLDCMD_CAM_CLI_ADD`	Adds a device client for the specific I2C bus and address.
`CAM_ClientRem` `LLDCMD_CAM_CLI_REM`	Removes given device client.
`CAM_ReadSingle` `LLDCMD_CAM_REG_RDS`	Writes a single device register using one of overloaded functions with different parameters register size.
`CAM_WriteSingle` `LDCMD_CAM_REG_WRS`	Reads a single device register using one of overloaded functions with different parameters register size..
`CAM_ReadBatch` `LLDCMD_CAM_REG_RDB`	Writes a bank of device registers.
`CAM_WriteBatch` `LLDCMD_CAM_REG_WRB`	Reads a bank of device registers.
`CAM_DRV_I2C_RepeatSet` `LLDCMD_CAM_I2C_REPEAT_SET`	Sets the number of I2C retransmission's repeat when I2C transmission is failed
`CAM_DRV_I2C_RepeatGet` `LLDCMD_CAM_I2C_REPEAT_GET`	Returns the number of I2C retransmission's repeat
`ControlGpio` `LLDCMD_CAM_GPIO_CONTROL`	Set/clear the output value of GPIO pin number
`CAM_DRV_Reset` `LLDCMD_CAM_RESET`	Reset camera by setting GPIO pin
`CAM_DRV_Power_Control` `LLDCMD_CAM_POWER_CONTROL`	Control camera power by setting GPIO pin

Table 3: Generic camera driver API

For Sony IMX224 the Generic camera driver module adds the functionality mentioned in Table 4.

Function, LLD command	Description

SONY_DRV_Setup LLDCMD_SONY_CONNECT	Driver initialization. Sets-up MipiCsi2 port pins and registers frame IRQ handler.
SONY_DRV_Close LLDCMD_SONY_DISCONNECT	Driver deinitialization. Unregisters the frame IRQ handler.
SONY_DRV_ExpStateGet LLDCMD_SONY_EXP_STATUS_GET	Gets the camera exposure setup state
SONY_DRV_ExpSent LLDCMD_SONY_EXP_SENT	Sets the driver internal exposure state to SENT

Table 4: Generic camera driver SONY IMX224 sensor API

For Ov10640 sensors the Generic camera driver module adds the functionality mentioned in Table 5

Function, LLD command	Description
OV10640_DRV_Setup LLDCMD_OV10640_CONNECT	Driver initialization. Sets-up MipiCsi2 port pins and registers frame IRQ handler.
OV10640_DRV_Close LLDCMD_OV10640_DISCONNECT	Driver deinitialization. Unregisters the frame IRQ handler.

Table 5: MIPI-CSI2 driver Ov10640 sensor API

3.3.2 Usage

To register a new I2C device client CAM_DRV_ClientAdd() function has to be called providing the sensor details in CamI2cCfg_t structure pointer. If not 0 CamI2cClient_t handle is returned the function has succeeded. The returned handle is a kernel pointer to CamGeneric_t structure retyped to uintptr_t. The camera structures are stored in a double-linked list inside the kernel space part of the driver. The handles are used as a general identification of the camera devices and are intended to be passed also to usespace.

To free a previously registered I2C device the CAM_DRV_ClientRem() function has to be invoked providing the CamI2cClient_t handle as its parameter.

Any device specific configuration is supposed to be done from user space part of the driver by invoking I2C transaction using one of the Read/Write functions or related LLD commands.

To implement functionality that requires to be executed in kernel space, e.g. handling of frame start/end IRQ of the CSI HW to implement exposure control, list of LLD commands of the generic camera driver is supposed to be extended. The extension should include the *CONNECT and *DISCONNECT commands for the particular device as can be seen in the Sony IMX224 case.

3.4 User Space

The Generic camera driver SW includes a user space library to abstract the kernel space API access from user applications. Part of the user library is also the device specific code which implements users-pace drivers for particular VSDK supported sensors.

3.4.1 API

Generic camera driver user level API mentioned in Table 6 is declared in `isp_cam_generic.h` and defined in `cam_user.cpp` file.

Function	Description
`CAM_Open`	Opens "`cam`" special device file on Linux or calls `CAM_DRV_Setup` in case of standalone environment.
`CAM_Close`	Closes "`cam`" special device file on Linux or calls `CSI_DRV_Close` in case of standalone environment.
`CAM_ClientAdd`	Adds a device client for particular I2C bus and address.
`CAM_ClientRem`	Removes given device client.
`CAM_ReadBatch`	Executes batch of I2C read commands.
`CAM_WriteBatch`	Executes batch of I2C write commands.
`CAM_ReadSingle`	Reads a single register value over I2C.
`CAM_WriteSingle`	Writes a single register value over I2C.
`CAM_I2C_SetRepeat`	Set up the number of I2C retransmission's repeat when I2C transmission is failed
`CAM_I2C_GetRepeat`	Get the number of I2C retransmission's repeat
`CAM_ControlGpio`	Set/clear the output value of GPIO pin number
`CAM_Reset`	Reset CSI/VIU camera by activate reset pin and then deactivate it after specific delay time of camera.
`CAM_PowerControl`	Power up/down CSI/VIU camera by setting reset and power pin

Table 6: Generic camera user library exported functions

Besides the generic API, which can be used for all specific sensor device (Table 6), the Generic camera user library contains also code, which is specific for each VSDK supported camera device. Table 7 mentions functions that should be available for each support sensor but internally can implementation differences. All function calls mentioned in the following table must be preceded with prefix of the particular sensor. E.g. "`SONY_`".

Function	Description
`Open`	Enables HW interaction (I2C client) and sets up internal structures.

Close	Releases all resources including I2C client, reset of internal structures.
RegWrite	Writes a value over I2C to Sony camera register.
RegRead	Reads a value over I2C from Sony camera register.

Table 7: Common calls for sensor specific APIs

For **Sony IMX224** sensor `isp_cam_sony.h` and `cam_sony.cpp` files have been added to the Generic camera library introducing API mentioned in Table 8. All function calls mentioned in the following table are preceded with “`SONY_`” prefix.

Function	Description
RegConfig	Performs initial sensor setup over I2C.
ExposureStatusGet	Checks current status of exposure control state machine.
CsiEnable	Enables CSI transmission from Sony camera.
CsiDisable	Disables CSI transmission from Sony camera.
ConvGet	Reads current setup of the conversion gain.
ConvLowSet	Sets the conversion gain to low.
ConvHighSet	Sets the conversion gain to high.
ExpLinesGet	Reads current setup of the exposure line count.
ExpLinesSGet	Reads current setup of the short exposure line count.
ExpLinesSet	Sets up the exposure line count.
GainGet	Reads current setup of the analog gain.
GainSet	Sets up the analog gain value.
ExposureGet	Returns current setup of the exposure parameters from the sensor. Has two parameter SONY_IDX and SONY_Exposure_t*.
ExposureSet	Applies provided setup of the exposure parameters to the sensor. Has two parameter SONY_IDX and SONY_Exposure_t*.
ObWvGet	Gets the number of optical black lines per frame.
ObWvSet	Sets the number of optical blacklines per frame.
SyncGet	Gets the sync mode of the camera (Master or Slave).
SyncSet	Sets the sync mode of the camera (Master or Slave). Has to match HW config.
VmaxGet	Gets the maximum number of lines per frame.
VmaxSet	Sets the maximum number of lines per frame.

HmaxGet	Gets the maximum number of columns per frame.
HmaxSet	Sets the maximum number of columns per frame.
CropWvGet	Gets the crop window number of lines per frame.
CropWvSet	Sets the crop window number of lines per frame.
CropWhGet	Gets the crop window number of columns per frame.
CropWhSet	Sets the crop window number of columns per frame.
CropPvGet	Gets the crop window y (line) offset from upper left corner.
CropPvSet	Sets the crop window y (line) offset from upper left corner.
CropPhGet	Gets the crop window x (column) offset from upper left corner.
CropPhSet	Sets the crop window x (column) offset from upper left corner.
WinmodeGet	Gets the window mode register value.
WinmodeSet	Sets the window mode register value.
GeometryGet	Returns current setup of the geometry parameters from the sensor. Has two parameter SONY_IDX and SONY_Geometry_t*.
GeometrySet	Applies provided setup of the geometry parameters to the sensor. Has two parameter SONY_IDX and SONY_Geometry_t*.
ExposureControlOver	Controls exposure to achieve certain (configurable) level of overexposed pixels.
ExposureControlEqual	Controls exposure to achieve balanced brightness. Allows offset to shift the balance to favor dark or bright areas.
HistogramEnergyRelativ	Computes histogram energy divided by number of pixels.
ChGainMeasure	Computes Channel Gain values from histograms.

Table 8: Camera generic user library API specific for SONY IMX224 sensor

For **Maxim Serializer/Deserialzier** setup the Generic camera user-space library adds the functionality mentioned in Table 9. All function calls mentioned in the following table are preceded with “MAXIM_” prefix.

Function	Description
Open	Enables HW interaction (I2C client) and setups internal structures. Has one parameter MAXIM_Cfg_t*.
9296B_Open	Connect to the Maxim 9296B HW setup.
Close	Release of all resources including I2C client, reset of internal structures. Has one parameter MAXIM_Cfg_t*.
9296B_Close	Disconnect from the Maxim 9296B HW setup

CsiEnable	Starts the data receive for the specified virtual channel.
9296B_CsiEnable	Enables CSI transmission from Maxim 9296B
CsiDisable	Stops the data receive for the specified virtual channel.
9296B_CsiDisable	Disables CSI transmission from Maxim 9296B
9296B_AR0231_InitComplex	Configures Maxim 9296B HW and AR0231 camera registers. Every camera/serializer will have unique addresses
OV9716_InitComplex	Performs initial Ser/Des and Ov9716 sensor setup over I2C. Includes assignment of unique I2C addresses for serializers and sensors on each of 4 deserializer channels.
OV10640_InitComplex	Performs initial Ser/Des and Ov10640 sensor setup over I2C. Includes assignment of unique I2C addresses for serializers and sensors on each of 4 deserializer channels.
OV10640_InitSimple	Performs initial Ser/Des and Ov10640 sensor setup over I2C. Serializers and sensors on each of 4 deserializer channels are accessible only in broadcasting mode.
OV10640_ExposureSet	Configures Exposure parameters to the OV10640 cameras. Exposure config is shared among the cameras.
OV10635_InitComplex	Performs initial Ser/Des and Ov10635 sensor setup over I2C. Includes assignment of unique I2C addresses for serializers and sensors on each of 4 deserializer channels.
SONY_InitComplex	Performs initial Ser/Des and Sony sensor setup over I2C. Includes assignment of unique I2C addresses for serializers and sensors on each of 4 deserializer channels.
OV9716_ExposureSet	Updates exposure parameters in the selected camera
OV9716_ExposureTimeSet	Sets the exposure time parameter of the channel 0 OV9716 camera for the selected exposure type
OV9716_ExposureTimeGet	Reads the exposure time parameter of the channel 0 OV9716 camera for the selected exposure type
OV9716_ExposureDigitalGainSet	Sets the digital gain parameter of the channel 0 OV9716 camera for the selected exposure type
OV9716_ExposureDigitalGainGet	Reads the digital gain parameter of the channel 0 OV9716 camera for the selected exposure type
OV9716_ExposureAnalogConversionGainSet	Sets the analogic gain parameter of the channel 0 OV9716 camera for the selected exposure type.
OV9716_ExposureAnalogConversionGainGet	Reads the analogic gain parameter of the channel 0 OV9716 camera for the selected exposure type.

MAXIM_OV9716_ChgainSet	Updates channel gain values for white balancing
DeserDump	Reads an array value over I2C to Maxim connected camera register begin address 0
CAM_RegWrite	Writes a value over I2C to Maxim connected camera register.
CAM_RegRead	Reads a value over I2C from Maxim connected camera register.
SER_RegWrite	Writes a value over I2C to Maxim connected serializer register.
SER_RegRead	Reads a value over I2C from Maxim connected serializer register.
DES_RegWrite	Writes a value over I2C to Maxim connected deserializer register.
DES_RegRead	Reads a value over I2C from Maxim connected deserializer register.
AR0231_CAM_RegWrite	Writes a value over I2C to Maxim connected AR0231 camera register.
AR0231_CAM_RegRead	Reads a value over I2C from Maxim connected AR0231 camera register.
9295A_SER_RegWrite	Writes a value over I2C to Maxim connected 9295A serializer register.
9295A_SER_RegRead	Reads a value over I2C from Maxim connected 9295A serializer register.
9296B_DES_RegWrite	Writes a value over I2C to Maxim connected 9296B deserializer register.
9296B_DES_RegRead	Reads a value over I2C from Maxim connected 9296B deserializer register.

Table 9: Generic camera user library API specific for MAXIM sensor Ser/Des setup

For **Omnivision Ov10635** sensor with parallel interface the Generic camera user-space library adds the functionality mentioned in Table 10. All function calls mentioned in the following table are preceded with “OV10635_” prefix.

Function	Description
Open	Enables HW interaction (I2C client) and setups internal structures. Has one parameter VIU_IDX.
Close	Release of all resources including I2C client, reset of internal structures. Has one parameter VIU_IDX.
RegConfig	Performs initial sensor setup over I2C.
DataEnable	Enables VIU transmission from camera.
DataDisable	Disables VIU transmission from camera.

Table 10: Generic camera user library for Omnivision Ov10635 sensor

For **Omnivision Ov10640** sensor with MipiCsi2 of parallel interface the Generic camera user-space library adds the functionality mentioned in

RegWrite	Writes a value over I2C to Ov10640 camera register.
`RegRead`	Reads a value over I2C from Ov10640 camera register.

Table 12

All function calls mentioned in the following table are preceded with “`OV10640_`” prefix.

Function	Description
`Open`	Enables HW interaction (I2C client) and setups internal structures. Has one parameter CSI_IDX/VIU_IDX.
`Close`	Release of all resources including I2C client, reset of internal structures. Has one parameter CSI_IDX/VIU_IDX.
`RegConfig`	Performs initial sensor setup over I2C for both cases CSI/VIU.
`DataEnable`	Enables VIU transmission from camera.
`DataDisable`	Disables VIU transmission from camera.
`CsiEnable`	Enables CSI transmission from camera.
`CsiDisable`	Disables CSI transmission from camera.
`Mode3x12`	Configures Ov10640 camera for 3x12bit output mode (raw12).
`AEC_WB_On`	Configures Ov10640 to do AEC and apply AWB.
`ChGain`	Configures Ov10640 channel gain related registers.
`GeometrySet`	Configures geometry parameters for OV10640 camera.
`ExposureInit`	Initialize exposure setting params for OV10640 camera.
`ExposureGet`	Reads Exposure settings from OV10640 camera.
`ExposureSet`	Configures Exposure parameters to OV10640 camera.
`ModeSet`	Configures operational mode to OV10640 camera.
`ExposureTimeGet`	Returns the Exposure time for the desired exposure type.
`ExposureTimeSet`	Sets the Exposure time for the desired exposure type.
`ExposureAnalogConversionGainGet`	Gets the analogue and conversion gain for all the exposure types.
`ExposureAnalogConversionGainSet`	Sets the analogue and conversion gain for all the 3 exposure types.

ExposureDigitalGainGet	Gets the digital gain for the desired exposure type.
ExposureDigitalGainSet	Sets the digital gain for the desired exposure type.
ExposureAllExpGet	Gets the exposures parameters (time and gains) for the desired exposure type.
ExposureAllExpSet	Sets the exposures parameters (time and gains) for the desired exposure type.
ExposureTotal	Computes all the exposure parameters for the desired exposure type.
ExposureComputationInd	Computes the Exposure parameters for all the 3 exposure types.
ExposureCorrection	Compute the correction factor to be applied to the total exposure based on histogram data
ExposureLSRatioChange	Change exposure long short ratio for OV10640 camera
RegWrite	Writes a value over I2C to Ov10640 camera register.
RegRead	Reads a value over I2C from Ov10640 camera register.

Table 12: Generic camera user library API for Omnivision Ov10640 sensor

For **Omnivision Ov9716** sensor `isp_cam_ov9716.h` and `ov9716_user.cpp` files have been added to the Generic camera library introducing API mentioned in Table 13. All function calls mentioned in the following table are preceded with “`OV9716_`” prefix.

Function	Description
Open	Enables HW interaction (I2C client) and setups internal structures. Has one parameter CSI_IDX/VIU_IDX.
Close	Release of all resources including I2C client, reset of internal structures. Has one parameter CSI_IDX/VIU_IDX.
RegConfig	Performs initial sensor setup over I2C for both cases CSI/VIU.
DataEnable	Enables VIU transmission from camera.
DataDisable	Disables VIU transmission from camera.
CsiEnable	Enables CSI transmission from camera.
CsiDisable	Disables CSI transmission from camera.
ChGainSet	Configures channel gain values for white balancing.
ExposureSet	Configures Exposure parameters to Ov9716 camera.
ExposureTimeGet	Returns the Exposure time for the desired exposure type.

ExposureTimeSet	Sets the Exposure time for the desired exposure type.
ExposureAnalogConversionGainGet	Gets the analogue gain for all the exposure types.
ExposureAnalogConversionGainSet	Sets the analogue gain for all the exposure types.
ExposureDigitalGainGet	Gets the digital gain for the desired exposure type.
ExposureDigitalGainSet	Sets the digital gain for the desired exposure type.
ExposureAllExpGet	Gets the exposures parameters (time and gains) for the desired exposure type.
ExposureAllExpSet	Sets the exposures parameters (time and gains) for the desired exposure type.
RegWrite	Writes a value over I2C to Ov9716 camera register.
RegRead	Reads a value over I2C from Ov9716 camera register.

Table 13: Generic camera user library API for Omnivision Ov9716 sensor

For **TI UB964** sensor with parallel interface the Generic camera user-space library adds the functionality mentioned in Table 14. All function calls mentioned in the following table are preceded with “TIUB964_” prefix.

Function	Description
ChanSelect	Toggle UB964 Rx channel Forwarding.
CsiEnable	Enables CSI transmission from TIUB964 camera
CsiDisable	Disables CSI transmission from TIUB964 camera
OV10640_Start	Enables data transmission from TIUB964 OV10640 camera.
OV10640_Stop	Disables data transmission from TIUB964 OV10640 camera
AR0140_Init	Configures AR0140 camera registers
Init	UB964 Configuration API
Close	Close TIUB964 maxim.
TestPattern_Setup	For only standalone . Configures turn on test pattern
TestPattern_Start	For only standalone . Starts test pattern data
TestPattern_Stop	For only standalone . Stop test pattern data

Table 14: Generic camera user library API for TIUB964 sensor

4 High Level Design

4.1 System Decomposition

The Generic Camera driver belongs to the complex data preprocessing subsystem of the s32v234 platform that is wrapped and controlled by the SDI library. Part of this subsystem is visualized in Figure 1. For more information about SDI and data preprocessing please refer to [1].

The preferred way to use the Generic camera functionality in a user application is to use Sequencer graphs together with the SDI library services. In the future the SDI library will provide complete abstraction of the Generic camera driver interface and thanks to utilization of the Sequencer HW the sensor data flow management load for the host CPU is minimized.

4.2 File Structure

Generic camera driver code is located in VSDK under s3234_sdk/libs/isp/cam_generic folder. Internally it has the following structure:

- kernel
    - build-v234ce-gnu-linux-d – build folder for Linux kernel module
        - Makefile
    - include
        - ar0140_config.h - definition of AR0140 camera registers configuration
        - ar0231_config.h - definition of AR0231 camera registers configuration
        - cam_func.h – declaration of Generic camera driver functionality
        - cam.h – general declarations/definitions
        - cam_lldcmd.h – declaration of LLD commands
        - cam_types.h – declaration of MIPI-CSI related data types
        - maxim_types.h – declaration of Maxim HW related data types
        - ov10635_config_max.h – Omnivision Ov10635 register configuration for maxim setup
        - ov10635_config_mipi.h – Omnivision Ov10635 register configuration for MIPI interface
        - ov10635_config_viu.h –Omnivision OV10635 register configuration for VIU interface
        - ov10635_types.h – declaration of Omnivision OV10635 data types
        - ov10640_config.h – Omnivision Ov10640 register configuration for Maxim Ser/Des setup on MIPI-CSI interface
        - ov10640_config_mipi.h – Omnivision Ov10640 register configuration for MIPI-CSI interface

        - ov10640_config_viu.h – Omnivision Ov10640 register configuration for VIU interface
        - ov10640_func.h – declaration of Omnivision Ov10640 driver functionality
        - ov10640_types.h – declaration of Omnivision OV10640 data types
        - ov10642_config.h - definition of omnivision OV10642 camera registers configuration
        - ov9716_config_lvds.h – Omnivision Ov9716 register configuration for Maxim Ser/Des setup on MIPI-CSI interface,ov9716_config_mipi.h – Omnivision Ov9716 register configuration for MIPI-CSI interface
        - ov9716_config_mipi.h – Omnivision Ov9716 register configuration for MIPI-CSI interface,
        - ov9716_config_viu.h – Omnivision Ov9716 register configuration for VIU interface,
        - ov9716_func.h – declaration of Omnivision Ov9716 driver functionality,
        - ov9716_types.h – declaration of Omnivision Ov9716 related data types
        - sony_config_DOL2.h – Sony IMX224 register configuration for dual exposure mode
        - sony_config.h – Sony IMX224 register configuration for single exposure mode
        - sony_func.h – declaration of Sony IMX224 driver functionality
        - sony_types.h – declaration of Sony IMX224 related data types
        - tiub964_types.h – declarations for TI hardware setup drivers
        - ub964_config.h – definition of deserlizer UB964 camera registers configuration
    - src
        - cam_core.c – Linux module related functionality
        - cam_func.c – definition of the Generic camera driver functionality
        - cam_lldcmd.c – definition of LLDCMD handling
        - ov10640_func.c – definition of Ov10640 sensor specific kernel functionality.
        - sony_func.c – definition of Sony Imx224 sensor specific kernel functionality.
- user
    - build-* – build folders for supported platforms (standalone and Linux)
        - Makefile
    - src
        - cam_user.cpp – definition of user space level public API

        - maxim_user.cpp – definition of user space level public API
        - ov10635_user.cpp – definition of user space level public API
        - ov10640_user.cpp – definition of user space level public API
        - ov9716_user.cpp – definition of user space level public API
        - sony_user.cpp – definition of user space level public API
        - tiub964_user.cpp – definetion of user space level public API
    - BUILD.mk – defines build details
- Public headers (s32v234_sdk/include):
    - isp_cam_generic.h – declaration of user space level public API,
    - isp_cam_maxim.h – declaration of user space level public API for MAXIM related HW setup,
    - isp_cam_ov10635.h – declaration of user space level public API for Omnivision Ov10635 with parallel interface,
    - isp_cam_ov10640.h – declaration of user space level public API for Omnivision Ov10640,
    - isp_cam_ov9716.h – declaration of user space level public API for Omnivision Ov9716
    - isp_cam_sony.h – declaration of user space level public API for Sony IMX224 with CSI interface.
    - isp_cam_ub964_api.h – declaration of user space level public API for TI serializer/desializer
    - isp_csi.h – declarations of user space level public API for MIPI-CSI interface