

libcamera Overview

© 2013 Sony Computer Entertainment Inc.
All Rights Reserved.
SCE Confidential

Table of Contents

1 Library Overview.....	3
Uses and Features	3
Files	3
Sample Programs.....	3
2 Using the Library	4
Basic Procedure	4
List of Functions	4
3 Reference Information.....	6
Device Physical Location Information Retrieval API	6
RAW8 Format.....	7
Simultaneous Use of Front and Rear Cameras	8
Judgment Method until Image Stability Status at Camera Start	8
Noise while using the cameras	8
Camera Suspension by the System.....	8
4 Notes for PlayStation®TV	9
Behavior when libcamera is Used with PlayStation®TV	9

1 Library Overview

Uses and Features

libcamera is a library of functions that perform video input from a camera device.

libcamera passes the raw data stream that is obtained from the camera directly to the application. The application is responsible for converting this raw data stream into an appropriate format so it can be displayed.

Files

The files required to use libcamera are as follows.

Filename	Description
camera.h	Header file
libSceCamera_stub.a	Stub library file

Sample Programs

Sample programs using libcamera are as follows.

sample_code/input_output_devices/api_camera/simple

This is a simple sample that illustrates the basic flow of how to use libcamera.

2 Using the Library

Basic Procedure

(1) Open Camera

As the information required to open the camera, decide the image format, the resolution and the frame rate. Also allocate the buffer required for the application to read the stream data.

Set to the `SceCameraInfo` structure the decided setting information and the allocated buffer size and address.

Call `sceCameraOpen()` to open the camera. Specify the *devnum* and `SceCameraInfo` structures as the arguments.

(2) Start stream

Call `sceCameraStart()` to start the stream.

(3) Read stream

By making the application call `sceCameraRead()` at the appropriate timing, the image data of the latest frame that can be captured at that point in time is copied to the buffer. Read the image data according to the buffer address set when the camera was open and convert and display it as appropriate. The frame number and timestamp can be obtained.

If blocking until the next frame image data is read with `sceCameraRead()`, it is recommended to read with a thread other than the rendering thread. Also, it is possible to avoid unnecessarily redundant reading by verifying active status with `sceCameraIsActive()` before reading (this can also be verified with *dwStatus* of the `SceCameraRead` structure). Refer to sample programs for examples of actual usage.

(4) Stop stream

Call `sceCameraStop()` to stop the stream.

(5) Close camera

Call `sceCameraClose()` to close the camera.

List of Functions

The functions that are provided by libcamera are listed below.

Open / Close Functions

Function	Description
<code>sceCameraOpen()</code>	Opens the camera
<code>sceCameraClose()</code>	Closes the camera

Streaming Functions

Function	Description
<code>sceCameraStart()</code>	Starts stream
<code>sceCameraRead()</code>	Reads stream data
<code>sceCameraStop()</code>	Stops stream
<code>sceCameraIsActive()</code>	Retrieves active status

Functions to Retrieve / Set Device Attributes

Function	Description
sceCameraGetSaturation ()	Retrieves the saturation
sceCameraSetSaturation ()	Sets the saturation
sceCameraGetBrightness ()	Retrieves the brightness
sceCameraSetBrightness ()	Sets the brightness
sceCameraGetContrast ()	Retrieves the contrast
sceCameraSetContrast ()	Sets the contrast
sceCameraGetSharpness ()	Retrieves the sharpness
sceCameraSetSharpness ()	Sets the sharpness
sceCameraGetReverse ()	Retrieves the mirror / flip
sceCameraSetReverse ()	Sets the mirror / flip
sceCameraGetEffect ()	Retrieves the image effects
sceCameraSetEffect ()	Sets the image effects
sceCameraGetEV ()	Retrieves the exposure compensation
sceCameraSetEV ()	Sets the exposure compensation
sceCameraGetZoom ()	Retrieves the zoom level
sceCameraSetZoom ()	Sets the zoom level
sceCameraGetAntiFlicker ()	Retrieves the anti-flicker
sceCameraSetAntiFlicker ()	Sets the anti-flicker
sceCameraGetISO ()	Retrieves ISO speed
sceCameraSetISO ()	Sets ISO speed
sceCameraGetWhiteBalance ()	Retrieves the white balance
sceCameraSetWhiteBalance ()	Sets the white balance
sceCameraGetBacklight ()	Retrieves the backlight compensation
sceCameraSetBacklight ()	Sets the backlight compensation
sceCameraGetNightmode ()	Retrieves the night mode
sceCameraSetNightmode ()	Sets the night mode
sceCameraGetAutoControlHold ()	Retrieves the auto control hold status
sceCameraSetAutoControlHold ()	Sets the auto control hold status
sceCameraGetExposureCeiling ()	Retrieves exposure ceiling
sceCameraSetExposureCeiling ()	Sets exposure ceiling

Functions to Retrieve / Set Information

Function	Description
sceCameraGetDeviceLocation ()	Retrieves the physical location information of the device

3 Reference Information

Device Physical Location Information Retrieval API

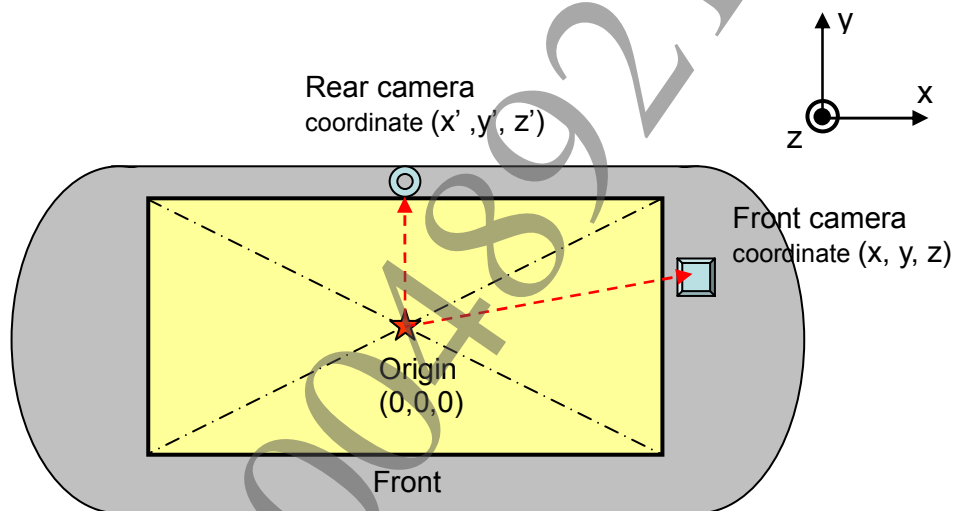
Overview

Depending on the application, the physical location information of the device may be required. libcamera can retrieve the physical location of the device from the origin (mm) by using the device physical location information retrieval API. However, the physical location of the device may change due to revisions of the main unit.

Device Physical Location Information Retrieval Method

The three-axis (`SceFVector3` type) physical location of the camera can be retrieved using `sceCameraGetDeviceLocation()`. The origin is the surface (z) of the center (x,y) of the screen. Figure 1 shows the relation between the 3 axes (X, Y, Z) and the origin.

Figure 1 Sensor Location Coordinates and Origin



Accuracy

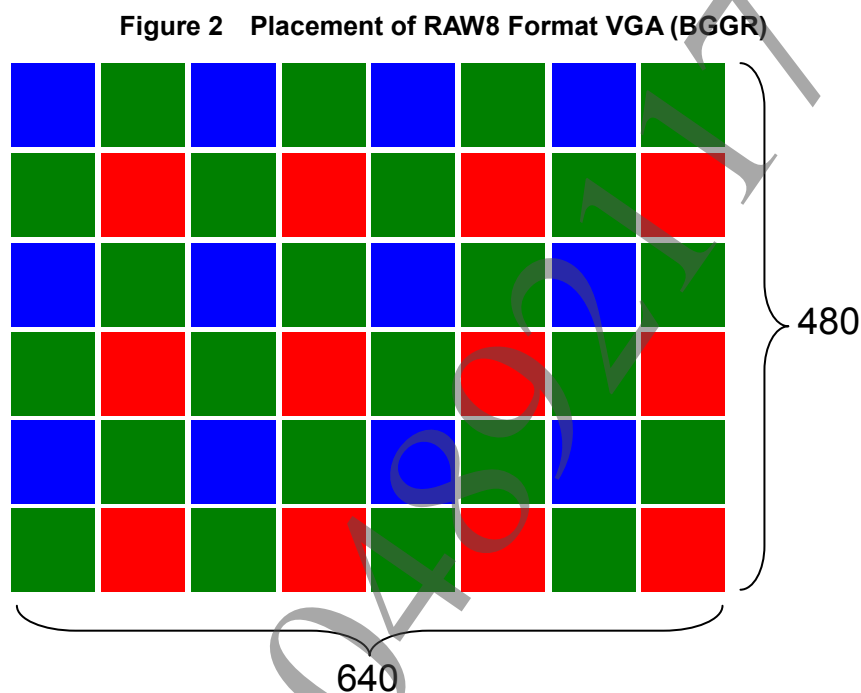
This accuracy is ± 1 mm, and the maximum/minimum values are ± 300 mm.

RAW8 Format

Overview

In addition to the YUV and RGB formats, libcamera supports the RAW8 format. Regarding the data of the RAW8 format that can be specified during `sceCameraOpen()`, the uncompressed RGB data is stored 1 pixel at a time as shown in Figure 2. Moreover, when read access with `sceCameraRead()` is successful, the RAW8 format pattern of the retrieved image can be checked with `dwRaw8Forma` of the `SceCameraRead` structure.

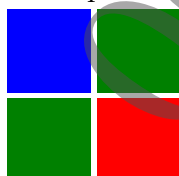
Figure 2 shows the placement of the RGB pixels of the BGGR pattern for VGA (640 x 480). The unit component of R, G and B is 1 pixel.



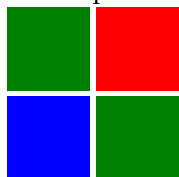
Patterns

The RAW8 format retrieved by `sceCameraRead()` consists of the following 4 patterns. It is expressed as 2 x 2 pixels.

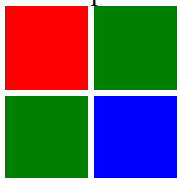
- (1) BGGR pattern (`SCE_CAMERA_RAW8_FORMAT_BGGR`)



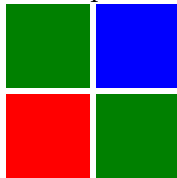
- (2) GRBG pattern (`SCE_CAMERA_RAW8_FORMAT_GRBG`)



(3) RGGB pattern (SCE_CAMERA_RAW8_FORMAT_RGGB)



(4) GBRG pattern (SCE_CAMERA_RAW8_FORMAT_GBRG)



Simultaneous Use of Front and Rear Cameras

libcamera does not permit the simultaneous use of the front and rear cameras. If it is attempted to start the stream of either the front or rear camera with `sceCameraStart()` in the state in which the one of the cameras has already started the stream through `sceCameraStart()`, the function will return an error code, resulting in a failure of the attempt to start the camera. In this case, first stop the already started camera stream with `sceCameraStop()`, and then start the other camera stream. For the error codes, refer to the "sceCameraStart" section and "Constants" in the "libcamera Reference" document.

Judgment Method until Image Stability Status at Camera Start

When the cameras start, a few images that are discolored may result for the first few frames owing to the specifications of the sensor. (For example, the images may have an overall yellowish hue.) If such occurrence is a possibility, `SCE_CAMERA_STATUS_IS_NOT_STABLE` is returned to `dwStatus` of `sceCameraRead()`, so please decide at the application level whether to display or discard such images. Refer also to the "sceCameraRead" section in the "libcamera Reference" document.

Noise while using the cameras

When using the cameras at a low frame rate (10 fps or lower) and in a low-light environment, pixel noise may occur. To avoid such noise, use the camera with a frame rate of 15 fps or higher.

Camera Suspension by the System

In the PlayStation®Vita, in some cases the cameras may be stopped by the system, disabling their use. In such cases, `SCE_CAMERA_STATUS_IS_FORCED_STOP` or `SCE_CAMERA_STATUS_IS_FORCED_STOP_POWER_CONFIG_CHANGE` will return to `dwStatus` of `sceCameraRead()`. When camera use is disabled with the power configuration control `SCE_CAMERA_STATUS_IS_FORCED_STOP_POWER_CONFIG_CHANGE` will be returned. For the power configuration control, refer to the "Power Configuration Control" chapter in the "Power Service Overview" document. In the case of other causes, `SCE_CAMERA_STATUS_IS_FORCED_STOP` will return.

4 Notes for PlayStation®TV

Behavior when libcamera is Used with PlayStation®TV

When the following APIs are called with PlayStation®TV, the error code SCE_CAMERA_ERROR_NOT_MOUNTED(0x802E0010) will return.

```
sceCameraOpen ()
sceCameraClose ()
sceCameraStart ()
sceCameraRead ()
sceCameraStop ()
sceCameraIsActive ()
sceCameraGetSaturation ()
sceCameraSetSaturation ()
sceCameraGetBrightness ()
sceCameraSetBrightness ()
sceCameraGetContrast ()
sceCameraSetContrast ()
sceCameraGetSharpness ()
sceCameraSetSharpness ()
sceCameraGetReverse ()
sceCameraSetReverse ()
sceCameraGetEffect ()
sceCameraSetEffect ()
sceCameraGetEV ()
sceCameraSetEV ()
sceCameraGetZoom ()
sceCameraSetZoom ()
sceCameraGetAntiFlicker ()
sceCameraSetAntiFlicker ()
sceCameraGetISO ()
sceCameraSetISO ()
sceCameraGetWhiteBalance ()
sceCameraSetWhiteBalance ()
sceCameraGetBacklight ()
sceCameraSetBacklight ()
sceCameraGetNightmode ()
sceCameraSetNightmode ()
sceCameraGetAutoControlHold ()
sceCameraSetAutoControlHold ()
sceCameraGetExposureCeiling ()
sceCameraSetExposureCeiling ()
sceCameraGetDeviceLocation ()
```