Stettbacher Signal Processing
Neugutstrasse 54
CH-8600 Dübendorf

Phone: +41 43 299 57 23
Fax: +41 43 299 57 25
E-Mail: cam@stettbacher.ch

O-3000 Camera Series

User's Guide

Version 2.10
2015-08-10

Abstract: This user's guide highlights the camera features and gives an overview over the device's basic functions. In addition for programmers and developers, it is a starting point and references to all necessary resources.

Contents

1 Intention

This user guide will highlight the camera features and give you an overview over the device's basic functions. In addition for programmers or developers, it is a starting point and will reference to all necessary resources. Please contact Stettbacher Signal Processing (SSP) if you have questions.

2 Camera Features

Technical data are specified in [2]. This section covers the functional features of the O-3010 monochrome and the O-3020 color devices in more detail. More devices will follow.

2.1 Functions

This chapter focuses on the camera's main functionalities and on how to use them. To use a specific camera function you can simply do it by sending appropriate XML commands [6] to the camera and receiving respectively.

2.1.1 Overview

The following is a list of all supported XML commands and parameters. Note that cameras do not necessarily support all of the functions implied by the XML specification [6].

Functions:

set	reset	snapshot	warning
get	restart	stream	error
		stop	

Parameters:

model_id	color_mode	acquisition	temperature
model_name	optical_format	color_weights	statistics
hw_version	pixel_size	mirroring	
sw_version	area	data	
xml_version	window		
serial_number	shutter_type		

2.1.2 Streaming and Snapshots

There are two possibilities to get image data from the camera device: single image snapshots or a continuous video stream. They only differ in the way they need to be addressed and in the result. A snapshot is a video stream in principle with just one picture. To get image data from the camera, see XML commands *streaming* and *snapshot* in [6].

2.1.3 Region of Interest

The region of interest (ROI) is an image area on the sensor to be defined. Usually it is set to full resolution of 1280 x 960 px (1.2 MP). It can be cropped to a smaller region of interest e.g. to reach higher frame rates. To set a particular region, use the *window* command described in [6].

2.1.4 Frame Rate

You can choose a frame rate. However, note, that the frame rate may affect the selected exposure time or at least the valid exposure *time_range*. Therefore, the *frame_rate* setting and the *time* acquisition mode have to be considered together.

Also note, that the *frame_rate* setting depends on the performance of the computer that is connected to the camera. In case of synchronisation errors (dropped frames) a rate reduction is necessary. With smaller window regions a higher frame rate is possible. For detailed explanations please see [6].

2.1.5 Exposure Adjustments

There are three different exposure modes, called *brightness*, *time* and *sensitivity*. To change the exposure mode, use the *aquisition mode* command.

The former mode *brightness* will, thanks to the camera's auto-exposure and statistics engine, adjust the image brightness automatically to the value set (in percent).

The latter two modes *time* and *sensitivity* are settings which can be adjusted independently and are part of the manual exposure mode. Exposure time can be set at least to 16 microseconds and moreover the sensor's gain (sensitivity) can be set to a value between 0 and 100 percent. To adjust these values you can send appropriate XML commands. All values can be set at any time but only one mode can be active at one time. For a detailed explanation please see [6].

The tags *time_range*, *brightness_range* and *sensitivity_range* serve to read from the camera the valid upper and lower limits for each parameter.

2.1.6 White Balance

You can adjust the different *color_weights* by setting their values *red*, *greenr*, *greenb*, and *blue* in the range from 0 to 100 percent. Thus, it allows you to make the white balance of the image manually. For more information see [6] and further application notes on the subject.

2.1.7 Mirroring

The mirroring mode allows you to flip the picture up/down or left/right. It can be done by changing values of parameter *mirroring*. For more information please see [6].

2.1.8 Data Format

The camera is supporting three different data formats (bit depths). You can choose between 8 bit, 12 bit and HDR[1] (20 bit compressed). This feature is most important for image processing algorithms or machine/computer vision applications because many monitors or print media are not able to distinguish more than 8 bit resolution per color channel. For more information about the data format identifiers please see [7] and for usage of the *data format* XML commands [6] as usual.

Note, that a monochrome camera only supports *mono* fomats and a color camera only supports the *bayer* modes.

2.1.9 Downsampling

Among the *advanced_functions* these cameras also support a *downsampling* mode, which is also called binning. In this mode, neighbouring pixels are combined. This reduces the image size and the image noise.

Note, that these cameras only support no binning, vertical binning and vertical & horizontal binning. Horizontal binning alone is not supported.

[1] High Dynamic Range.

2.2 Hardware Interfaces

- Main communication interface:
  - High-speed USB 2.0 interface.
  - Flat micro-B connector on board-level devies.
  - Standing mini-B connector on devices with housing.
  - Please contact SSP if you prefer a standing mini-B connector on board-level devices.
- Supply voltage output:
  - A supply voltage output is available for external use on board-level cameras.
  - 5.0 V nominal, typically 4.5 to 5.5 V delivered by USB host, 50 mA peak.
  - 5-pin, 1.25 mm male vertical connector (Würth Elektronik, no. 653’005’114’822).
- PWM outputs:
  - Two pulse width modulation outputs (plus reference ground) are available on board-level cameras.
  - Typical use: control of one or two light sources.
  - 5-pin, 1.25 mm male vertical connector (Würth Elektronik, no. 653’005’114’822).
  - Please contact SSP if you wish to use this feature.
- Flash synchronization:
  - One synchronization output (plus reference ground) is available on board-level cameras.
  - 5-pin, 1.25 mm male vertical connector (Würth Elektronik, no. 653’005’114’822).
  - Please contact SSP if you wish to use this feature.
- Synchronization of multiple cameras:
  - One pin (plus reference ground) can be used as synchronization input or output on board-level cameras.
  - Typical use: stereoscopic vision, stereopsis, camera arrays.
  - 2-pin, 1.25 mm male vertical connector (Würth Elektronik, no. 653’002’114’822).
  - Please contact SSP if you wish to use this feature.
- Lens holder:
  - The board-level cameras are available with C-, S- and CS-mount lens holders.
  - The camera housing provides a C- and CS-mount lens holder.
  - The board-level cameras feature pairs of 2.1 mm mounting holes with distance 20.0 mm and 22.0 mm to accept proprietary lens holders.
  - Please contact SSP for support or for other lens holder options.

2.3 Software Interfaces

Note: The O-3000 cameras are no webcams. They do not support the UVC standard used in plug-and-play consumer stuff. Instead the O-3000 cameras offer a simple and flexible interface, optimized for industrial vision.

There are two different levels, allowing access to the camera. The more comfortable one is through the open-source camera driver, available in the O-3000 install package. However, it is also possible to access the camera directly through the USB endpoints, without using the driver. The second way requires more programming effort but leads to a very slim and optimized solution. And of course the open-source driver can be used as a guideline for the second way.

In any case, communication to the camera consists of two different paths, one for messages to and from the camera and one to transport image data. The message path is used to send configuration commands to camera and to read information from the camera, such as the software version, camera settings, and so on.

2.3.1 Driver-Level Access

The O-3000 driver provides a simple and light-weight API to communicate with the O-3000 camera series. It abstracts user applications from the USB internas and deals with low-level interfacing which is tedious to implement and hard to debug if done the wrong way. The driver is written in plain C. For use in Java, such as the O-3000 demo application, the JNI (Java Native Interface) wrapper may be used. See [4] for more information.

2.3.2 USB-Level Access

Mainly for embedded projects, it is mandatory to be able to interact directly with the camera. On the host side, that means interacting with the USB interface. You do not have to fully understand the USB 2.0 standard, but some basic knowledge is essential. Each USB connection provides a set of independent sub-channels, called pipes. The host-side end of the pipe are buffers, the device-side end consists of so-called endpoints. Each endpoint is uniquely addressed by the USB device's address and the endpoint number. Endpoint 0 is reserved for device control; the other endpoints (up to 15 for high-speed devices) are implemented application specific. An endpoint offers an IN channel (device to host) and an OUT channel (host to device), although not both of them have to be implemented.

The O-3000 cameras utilise two USB endpoints in bulk mode for messages and for image data. Details are explained in [5].

3 Install Package

The install package, available from the download section of the O-3000 website (see [1]), contains the O-3000 driver and a demo application to operate the O-3000 cameras with a PC (Linux or Windows) or a Mac. The install package also includes the entire source code of the driver and the demo application.

Check [3] for support about setting up the install package or running the demo application.

3.1 Demo Application

With the demo application you can use the O-3000 camera and test some of its operating modes (see [3]).

At this point, we focus on the software architecture of the demo application. We start with the system and its boundaries. Figure 1 shows the particular modules in interaction with each other. On top we have the actual demo application (written in Java) which is connected through several layers to the camera. All modules described are part of the driver package. We will come to that later on in section 3.2. Basically, the top layer application (e.g. the Java demo application) is connecting through the JNI[2] wrapper and via the camera driver functions and libusb respectively to the operating system. The operating system itself is using system calls to communicate with the O-3000 camera series device.

3.2 Source Code

The install package includes all the sources and creates a file structure on your system according to the driver documentation [4]. Basically, there are four important directories you will need to take into consideration for the further programming or developing process. It is:

- The bin directory which helds the java demo app.
- The include directory in which the header files are stored.
- The lib directory in which the libraries are.
- The src directory where all the source code is located.

[2] Java Native Interface.

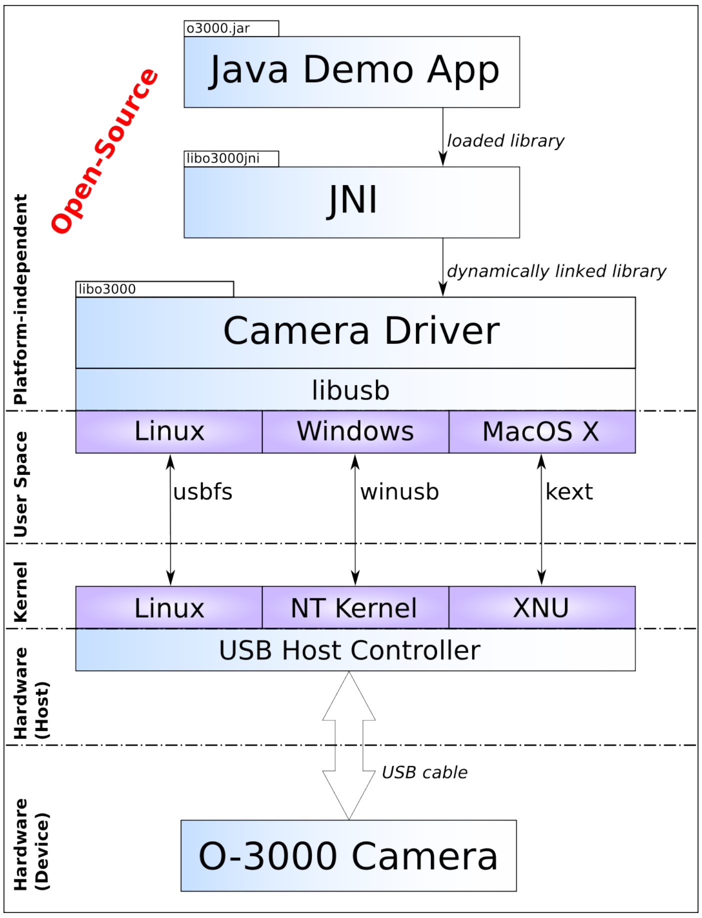

Figure 1: System overview.

The O-3000 camera series device comes with a driver package hiding the complexity concerning USB communication [5] (based on the open-source project libusb) and therefore can be seen as the application programming interface (API). The whole package consists of three different parts to illustrate a working example and can be modified under terms of GPL and LGPL:

- The O-3000 camera driver (binaries and C sources).
- A Demo Application, an implementation of a GUI software for this camera (binaries and Java source).
- A Java Native Interface (JNI, binaries and C sources) acting as a link between the O-3000 camera driver and the Java Demo Application.

For detailed information about software installation, building binaries from source code, about the source code itself, and range of functions please see [4].

4 Literature and References

[1] Stettbacher Signal Processing (2015). Website for the O-3000 products: http://www.open-cam.ch.

[2] Stettbacher Signal Processing (2013). “O-3000 Camera Series - Data Sheet”, Stettbacher Signal Processing, version 1.20, 2013-09-09 (or later).

[3] Stettbacher Signal Processing (2014). “O-3000 Camera Series - First Steps”, Stettbacher Signal Processing, version 1.00, 2014-09-22 (or later).

[4] Stettbacher Signal Processing (2013). “O-3000 Camera Series - Camera Driver Package Documentation”, Stettbacher Signal Processing, version 1.20, 2013-05-24.

[5] Stettbacher Signal Processing (2013). “O-3000 Camera Series - Camera Low Level Programming Interface”, Stettbacher Signal Processing, version 1.20, 2013-05-24.

[6] Stettbacher Signal Processing (2013). “O-3000 Camera Series - XML Command User Specification”, Stettbacher Signal Processing, version 1.20, 2013-05-24.

[7] Stettbacher Signal Processing (2013). “O-3000 Camera Series - Image Frame Format Description”, Stettbacher Signal Processing, version 1.20, 2013-05-24.