Stettbacher Signal Processing
Margrit-Rainer-Strasse 12a
CH-8050 Zürich

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

O-3000 Camera Series

XML Command Specification

Version 1.20
2013-06-19

Abstract: The command interface of the O-3000 camera series uses XML style commands. This document lists all commands and explains how to use them.

Contents

1 General XML Considerations

1.1 Terminology

For general terminology, i.e. the definition of terms like *tag*, *element*, *content*, *attribute*, etc., refer to the XML specification [1] or to [2].

1.2 Character Set

1.2.1 Valid Characters

Valid characters are limited to ASCII range 20_h to $7F_h$, plus carriage-return (CR, ASCII code $0D_h$) and line-feed (LF, ASCII code $0A_h$).

1.2.2 Reserved Characters

Characters '<' (opening angled bracket, ASCII code $3C_h$) and '>' (closing angled bracket, ASCII code $3E_h$) are for use in tags only.

1.3 Telegram Types

XML telegrams have different purposes:

- device configuration.
- device state retrieval.

Communication is initiated by the host usually (polling). Unsolicited messages such as indication of skipped frames may be sent asynchronously, i.e. outside a request/response exchange.

1.4 General XML Message Format

Each XML telegram is organised in hierarchical levels, as shown below:

```
<level_1>
    <level_2>
        <level_3> value </level_3>
        ...
    </level_2>
    ...
</level_1>
```

The following restrictions apply:

- The protocol does not imply a maximum number of levels. The depth (number of levels) results from the definition of the parameters.
- Exactly one first-level element (root element) is allowed. The root element is also called telegram frame.
- Any number (including zero) of elements are allowed on each level.
- Values (i.e. content other than child elements) are only allowed on the 2nd level and below.
- Upper and lowercase letters are allowed.
- Attributes (e.g. <foo val="1234">) and empty-element tags are not supported.

For more information see the examples in the following chapters.

1.5 Element Hierachy

The O-3000 camera's XML message format is derived from the general XML message format defined in 1.4. To each level in the general format, a specific meaning in the camera XML specification is mapped as follows. See table 1 for more comments.

```
<device_tag>
    <command>
        <parameter>   </parameter>
        <parameter_group>
            <parameter>   </parameter>
            ...
        </parameter_group>
        ...
    </command>
    ...
</device>
```

Level	Symbol	Description	Ref.
1	device	Device type targeted by the message.	2.1
2	command	Commands sent to target; also used as wrapper for various responses. Some commands require/provide parameters, while others don't.	2.2
3 (and below)	parameter (or _group)	Command parameters; two types exist: 1. Single value parameter (includes vectors and matrices). 2. Parameter group, containing further parameters or parameter groups.	2.4

Table 1: Element hierarchy.

2 XML Messages

This chapter explaines how to use camera specific XML messages by illustrating different examples.

2.1 Device Level

The device tags in table 2 are used for O-3000 cameras.

Device tag	Description
camera	O-3000 camera (series)

Table 2: Device tags.

2.2 Command Level

The 2nd level specifies commands. See table 3 for an overview of command tags.

Command tag	Parameters	Description
set	see 2.4	Writes parameter value to a camera device. This command requires at least one parameter.
get	see 2.4	Gets parameter value from a camera device. This command requires at least one parameter.
my	see 2.4	Indicates response to a *get* command.
reset	none	Reloads default values.
restart	none	Restarts device.
stream	none	Starts video streaming.
stop	none	Stops video streaming.
snapshot	none	Acquires a single image.
warning	see 2.5	Warning message, see 2.5 for details.
error	see 2.5	Indicates an error, see 2.5 for details.

Table 3: Command tags.

2.3 Message Syntax

Before we get through all the available camera parameters, we first show how to implement specific XML messages with several examples. The user will be able afterwards to implement his own application messages.

2.3.1 Set Camera Parameters

Single parameter Table 4 shows the XML message to crop the window geometry (ROI: region of interest) to a resolution of 800 x 600 px.

Request	Response
<camera> <set> <window> (0 799 0 599) </window> </set> </camera>	No XML response in case of success.

Table 4: Crop window geometry.

Multiple parameters The message in table 5 will crop the geometry alike and, furthermore, it will adjust the frame rate to 20 fps. Note: there is no limitation on the number of parameters to set simultaneously. If one specific parameter is supplied multiple times, the last entry takes effect.

Request	Response
<camera> <set> <window> (0 799 0 599) </window> <frame_rate> 20 </frame_rate> </set> </camera>	No XML response in case of success.

Table 5: Crop window geometry and set frame rate.

Lower level parameters The XML message in table 6 will set the acquistion mode to *time* with the corresponding exposure time to 520 microseconds. Note: Combinations of higher and lower level parameters are possible.

Request	Response
<camera> <set> <acquisition> </mode> time </mode> </time> 0.00052 </time> </acquisition> </set> </camera>	No XML response in case of success.

Table 6: Set acquisition mode and the exposure time.

2.3.2 Get Camera Parameters

Single parameter Table 7 shows how to read the camera's ID.

Request	Response
<camera> <get> <model_id> </model_id> </get> </camera>	<camera> <my> <model_id> 1 </model_id> </my> </camera>

Table 7: Retrieve camera ID.

Mutliple parameters The XML message in tabel 8 will get the camera's installed software version and the window size. Note: There is no limitation on the number of parameters to set simultaneously.

Request	Response

<camera>
  <get>
  <sw_version>
  </sw_version>
  <window>
  </window>
  </get>
</camera>
	<camera>
	  <my>
	  <sw_version>
	  1.2
	  </sw_version>
	  <window>
	  (0 1279 0 959)
	  </window>
	  </my>
	</camera>

Table 8: Retrieve software version and current crop window geometry.

Lower level parameters Table 9 shows the XML message to get the weights (gain) of the different pixels.

Request	Response

<camera>
  <get>
  <color_weights>
  </color_weights>
  </get>
</camera>
	<camera>
	  <my>
	  <color_weights>
	  <red>
	  10.000000
	  </red>
	  <greenr>
	  10.000000
	  </greenr>
	  <greenb>
	  10.000000
	  </greenb>
	  <blue>
	  20.000000
	  </blue>
	  </color_weights>
	  </my>
	</camera>

Table 9: Retrieve color weights.

2.3.3 Start/Stop Video Streaming

The XML message in table 10 will start a camera stream (asynchronous call back, no XML response). This is a 2nd level XML message without parameters. To stop the camera streaming process simply substitute 'stream' with 'stop' in the message of table 10.

Request	Response
<camera> <stream> </stream> </camera>	Asynchronous call back, no XML response.

Table 10: Start video streaming.

2.3.4 Take a Snapshot

Table 11 shows the XML message to take a camera snapshot. A snapshot is a single image. Note: Streaming will be stopped and has to be restarted manually after sending the *snapshot* command.

Request	Response
<camera> <snapshot> </snapshot> </camera>	Asynchronous call back, no XML response.

Table 11: Take a snapshot.

2.3.5 Restart the Camera

The command from table 12 will restart the camera.

Request	Response
<camera> <restart> </restart> </camera>	No XML response in case of success.

Table 12: Restart camera.

2.3.6 Reset to Factory Settings

Table 13 shows the command to reset the camera to its factory settings.

Request	Response
<camera> <reset> </reset> </camera>	No XML response in case of success.

Table 13: Reset camera.

2.3.7 Invalid Request

The message in table 14 cannot be sent this way because the paramater *sw_version* is a read-only parameter. Therefore, the camera will send a response informing about that.

Request	Response
<camera> <set> <sw_version> 1.2 </sw_version> </set> </camera>	<camera> <error> <parameter> sw_version </parameter> <code> -2 </code> <message> Read-only parameter </message> </error> </camera>

Table 14: Invalid request.

2.4 Parameter Level

The 3rd and lower levels specify parameters (see table below). Parameters may contain single values, or may be grouped where appropriate. They are accessed by the commands *set* and *get*.

Parameter Tags	Type	Unit	Acc	Description/Notes
model_id	int	-	ro	Identifies camera model.
model_name	string	-	ro	Camera product name.
hw_version	string	-	ro	Hardware version.
sw_version	string	-	ro	Software version.
xml_version	string	-	ro	XML protocol version.
color_mode	enum	-	ro	Color mode. Valid values: color, mono
serial_number	int	-	ro	Serial number.
optical_format	string	-	ro	Optical format, e.g. 1/3”.
pixel_size	int_vec	μm	ro	Size of a pixel on the image sensor (2 values.: x y).
temperature	float	°C	ro	Temperature of image sensor.
area	int_vec	px	ro	Usable area of image sensor. 4 values: index 0: min. x-coordinate index 1: max. x-coordinate index 2: min. y-coordinate index 3: max. y-coordinate

Parameter Tags	Type	Unit	Acc	Description/Notes
window	int_vec	px	r/w	Crop window (ROI: region of interest). 4 values: index 0: start x-coordinate index 1: end x-coordinate index 2: start y-coordinate index 3: end y-coordinate
shutter_type	enum	-	ro	Shutter type. Valid values: rolling, global
frame_rate	float	fps	r/w	Frame rate; affects exposure.
acquisition				Exposure settings
mode	enum	-	r/w	Exposure mode. Valid values: time (manual exposure), brightness (automatic exposure), sensitivity (manual exposure)
time	float	s	r/w	Exposure time (time exposure mode only; affects frame rate)
time_range	float_vec	s	ro	Exposure time range. 2 Values: index 0: min. exposure time index 1: max. exposure time
brightness	float	%	r/w	Brightness value.
brightness_range	float_vec	%	ro	Brightness range. 2 Values: index 0: min. brightness index 1: max. brightness
sensitivity	float	%	r/w	The sensor's sensitivity (sensitivity exposure mode only).
sensitivity_range	float_vec	%	ro	Sensitivity range. 2 Values: index 0: min. sensitivity index 1: max. sensitivity
color_weights				Color weighting of the individual color components.
red	float	%	r/w	Weight of red pixels.
greenr	float	%	r/w	Weight of green pixels (adjacent to red pixels).
greenb	float	%	r/w	Weight of green pixels (adjacent to blue pixels).
blue	float	%	r/w	Weight of blue pixels.
mirroring	enum	-	r/w	Readout order of image sensor. Valid values: none: normal order x: reverse order in horizontal direction y: reverse order in vertical direction xy: reverse order in horizontal and vertical direction
data				

Parameter Tags	Type	Unit	Acc	Description/Notes
format	enum	-	r/w	Video format settings. Valid values: see [4].
advanced_functions				Advanced/special functions not relevant for basic operations.
downsampling	int_vec	-		Reduce sensor noise by enabling binning. See note below. Valid values: (1 1): normal readout (2 1): 2x binning in horiz. direction (1 2): 2x binning in vert. direction (2 2): 2x binning in horiz. and vert. direction
statistics				Statistics/usage information.
data_rate	int	bytes/s	ro	Current data rate.
data_sent	int	bytes	ro	Total number of bytes sent.

Table 15: Parameters.

Notes:

- Data type int_vec is a vector of integers.
- Data type enum is a (string) value out of a set of predefined values individual for every parameter.
- Binning in horizontal direction by factor 2 reduces resolution in the same direction by a factor of 2. Same principle applies to vertical direction and for other binning factors (if available).

Access mode	Description
ro	Read-only parameter. Write attempts (command set) will cause a warning or an error.
wo	Write -only parameter. Read attempts (command get) will cause a warning or an error.
r/w	Read/write parameter. Parameter can be read (command get) or written (command set). Note: changing certain parameters might cause a short interrupt in the stream, while the camera switches to standby mode and back.

Table 16: Parameter access modes.

2.5 Unsolicited Messages

Messages in this context contain error messages in response to an invalid request, or asynchronous messages sent by the device. Such messages consist of command *error* in level 2 and the parameters *parameter*, *code*, and *message* in level 3.

Command tag	Description
error	Indicates error situation; either in response to a request, or asynchronously sent by the device.
warning	In response to a request, indicates that the request caused a non-critical problem.

Table 17: Commands.

Parameter tag	Data type	Access	Description/Notes
parameter	string	ro	Contains parameter or source that triggered the message.
code	integer	ro	Error code (see table).
message	string	ro	Error description.

Table 18: Parameters.

2.6 Data Types

A number of data types are supported, as described below. Note that trailing and leading white spaces around parameter values or between tags don't care and will be ignored by the XML parser.

Data type	Description	Examples
int	Decimal integer numbers (with optional leading sign character) and hexadecimal numbers (with 0x prefix).	12576 +12576 -175 0x25
float	Decimal floating point numbers (with optional sign character/-exponent).	211123.007 +53.567 -1.2345 54.34e5 10.394e-3
int_vec float_vec	Vector of integers or floats.	(1e3 4.7 -2.5e-3 +34 873.e-6)
int_mat float_mat	Matrix of integers or floats.	((1 3)(4.3 -34.5e-2))
string	String of ASCII characters (refer to 1.2 for limitations).	Hello World!
enum	Subset of strings, i.e. the string may only assume a set of values which are context-specific (dependent on the parameter).	yes, no enabled, disabled none, x, y, xy

Table 19: Data types.

3 Error Handling

In terms of compatibility to different devices/device revisions, the implementation of the XML protocol is as far as possible error-tolerant regarding unknown or obsolete elements. This means, the device (O-3000 camera) sends an error response to the application for elements which are not recognised, but otherwise ignore that part of the message. It also won't shut down or result in failure in such described cases.

The camera tries to react as smart as possible upon invalid parameters, e.g. parameters not within a valid range and similar. In such a case, the camera will assume and apply a value, which means, it might limit a value, or round it up/down. Additionally, the camera will return a warning to inform the application. The application might verify the value applied with a *get* request. Such errors and warnings will be passed to the calling instance by callback functions. For a detailed explanation see Driver API in [3].

However, error-tolerance as described above does not accept any malformed XML messages.

4 Appendix

4.1 Crop Window Definition

In order to mask out irrelevant or undesired parts of the image sensor's active area, or to improve the frame rate, the read-out (crop) area can be limited. By default, the crop window extends to the full image sensor active area, i.e. the crop area and the image sensor area coincide.

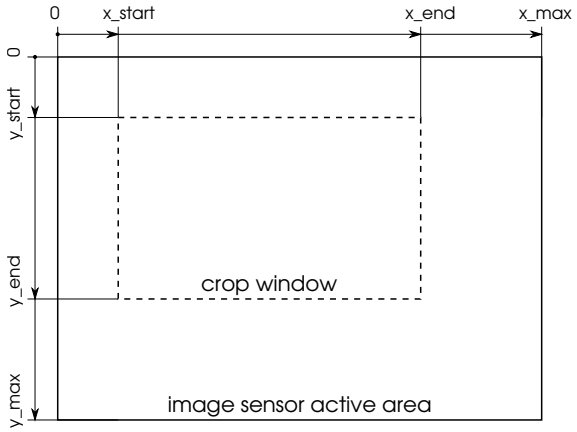

Figure 1: Crop window.

4.2 Data Formats

Please see [4] for futher information.

5 Literature and References

[1] W3C: Extensible Markup Language (XML), http://www.w3.org/XML/

[2] Wikipedia: XML, http://en.wikipedia.org/wiki/Xml#Key_terminology

[3] Stettbacher Signal Processing (2013). "O-3000 Camera Series - Camera Driver Package Documentation", Stettbacher Signal Processing, version 1.20 or above, 2013-05-24.

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