O3D2xx Programmers Guide

For Software-Version: 404x

PRELIMINARY

ifm electronic GmbH

TOE

Revision History

Date & Author	Description
21-December-2010, TO-hl	fixed description of data size in image header
09-September-2009, TO-ce	Fixed wrong description of "MDAXMLSetFrontendData"
19-February-2009, TOE-ce	O3D2xx version introduced.
26-February-2008, TOE-ff	Draft & Review

Disclaimer

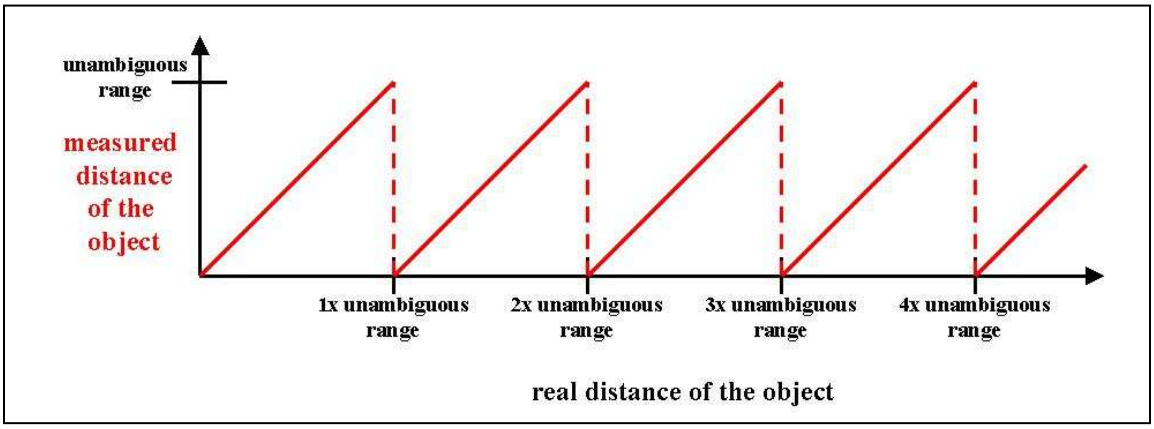
Table of Contents

1. Introduction

The O3D is a distance measuring camera. It contains a 64x50 Pixel PMD-Matrix, which provides intensity and distance images for each pixel of the scene. The O3D provides two interfaces, one is an ethernet connection for parametrizing the sensor and obtaining data, the other is a process connection providing digital input/output lines and an analog output.
The communication with the O3D is based on two TCP/IP ports. The first port is configuring the camera settings based on an XML-RPC server-protocol. The XML-RPC interface library can be obtained from www.xmlrpc.com [1]. The second port transfers image-data based on a TCP/IP bytestream socket. Each picture is queried by sending a single byte to the sensor via this socket.

1.1. O3D Basics

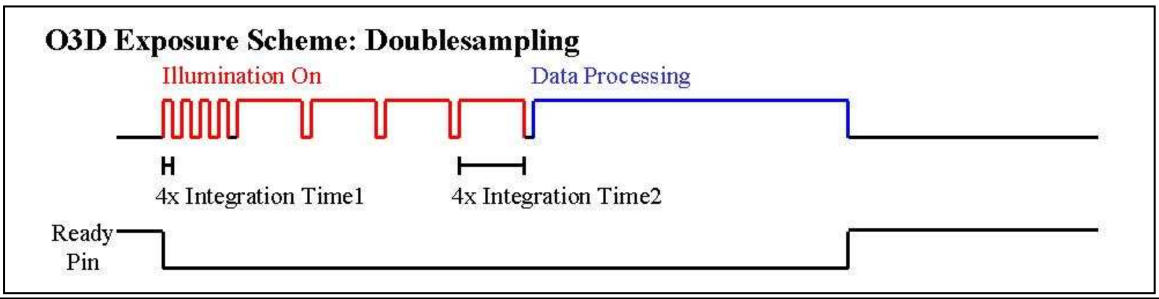
The distance measurement is based on the time-of-flight principle. The PMD-technology enables a matrix-arrangement of the detector elements on a ½''-chip. Due to the distance measurement principle the sensor can detect distances up to a unambiguous range of 6.5m. Any object beyond is seen modulo the unambiguous range, eg an object 8m away will be seen at 1.5m.

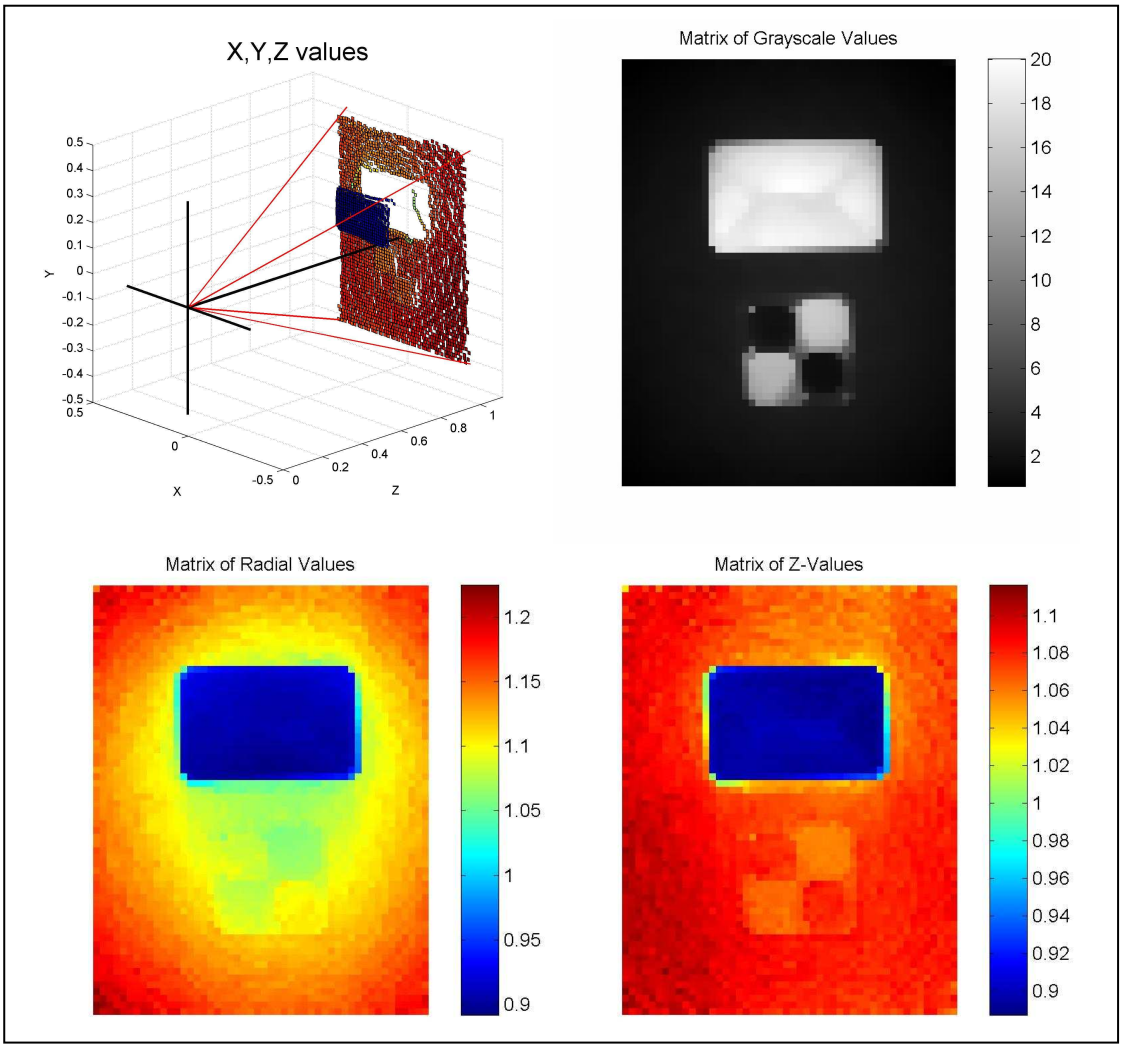


The unambiguous range can be increased by using a dual frequency mode. In this mode two exposures at different frequencies are taken and combined, such that the unambiguous range is increased to 45m while the framerate is halfed.

Frequency mode	Single FRQ0	Single FRQ1	Single FRQ2	Dual FRQ3
Unambiguous Range	6.51m	7.38m	7.40m	45m

The O3D illuminates the scene with its internal modulated infrared LED light source. The reflected light is collected by the lens and passed onto the PMD-Matrix. In order to measure the distance four exposures are required. These are carried out in quick succession and are followed by approx. 30ms data processing time, in which the image is calculated based on raw data.





The accuracy of the distance measurement improves with higher collected light levels. Thus distances of dark objects are noisier than those of bright objects. Since the sensor illuminates the scene, the light level falls quadratically with distance. Thus if the distance of an object is doubled, its light level decreases to a fourth of the original value. The combination of both effects, distance and reflectivity, produces a high dynamic range of light levels in a scene. Therefore black objects, having a diffuse reflectivity of 5%, can be detected up to distances of about 4m. Retroreflecting objects can be detected up to distances of 150m, so they will pass through the unambiguous range several times.
Since light is used as the measuring media, any strong ambient light, like the sun or spotlights, can increase the noise of the measurement. Be assured that maximum effort has been applied, to reduce this effect.

1.2. Output Data Formats

The O3D determines the distance and the greyscale value for each pixel. The distance information is provided in several formats. A combination of several types is possible.
The distance is intrinsically measured in a spherical coordinate system, center of this coordinate system is the theoretical focus point of the sensor. Consequently the sensor is able to provide a matrix with radial distance values. Using the focal length of the lens the three matrices of the Cartesian X, Y and Z coordinates are calculated.
Again, each matrix can be queried individually or in combination with others.

1.3. HeartBeat

If the communication client to O3D is interrupted and the client tries to reconnect to the O3D, this attempt will fail as the O3D still has its ports open. To avoid this situation the heartbeat function can be enabled. If the O3D doesn't receive a XML-heartbeat command every 10 seconds, it will close its ports and the client is able to reconnect with the camera.
If there is no need to keep the XML-Communication open, because there is no need to modify sensor parameters or releasing software triggers, the XMLRPC Port can be closed.
The Data-Communication Port provides a TCP-Keepalive Timeout with a value of 10 seconds. After detecting a communication error the sensor will close all communication ports and resets the communication servers. After this a client can reconnect. While the Display shows "OnLi" there is no possibility to connect to the sensor with a separate client.

1.4. Image Acquisition Settings

There are several image acquisition settings to optimize the camera performance. As with any regular camera an important parameter is the exposure time, which is called the integration time. Generally the longer the integration time the more distant objects can be detected. At the same time bright objects in the near neighborhood will start to saturate.
To be able to see close and distant objects at the same time, a double integration scheme is the standard working mode of the sensor. Here, two measurements, one using a long, one using a short integration time are carried out and for each pixel the appropriate values are selected.
To optimize the integration time it is recommended to work in single integration mode without any filters. The best integration time regime is found when most pixels are neither saturated nor underexposed. If singular pixels are saturated, they are being taken care of by the second short integration time. If the first long integration time is too long, all pixels will be taken from the short integration time and therefore the performance doesn't improve with longer integration time.
Unless speed is the ultimate priority, it is always recommended to operate in double sampling mode. In most applications a ratio of 10 between long integration time and short integration time is appropriate.
The integration time of one exposure can be varied in the range from 1 to 5000µs. The total integration time for one image is four times the exposure time plus a small read-out time.
The default value of the long time integration time is 2000µs.

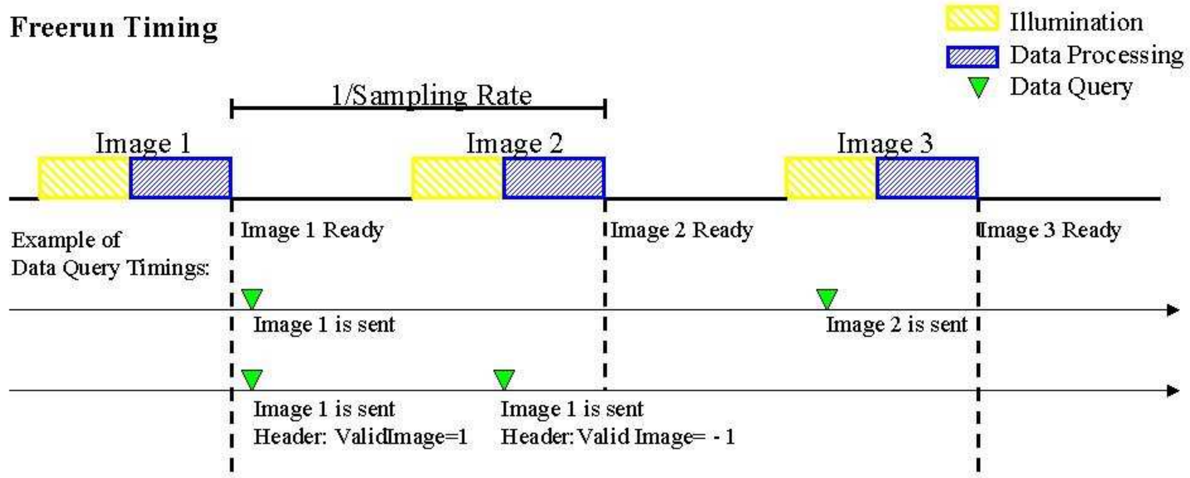
1.5. Image Acquisition Timing

The O3D can operate in different timing schemes that define when images are taken. These modes are the free-running and the triggered mode.
In free-running the sensor is continuously taking images according to current acquisition mode (single/double integration, single/dual frequency, image averaging). The time span of the current mode is called evaluation time and is provided in the image header. In order to operate a particular framerate set the inter-frame-mute-time (ifm-time) accoring to:

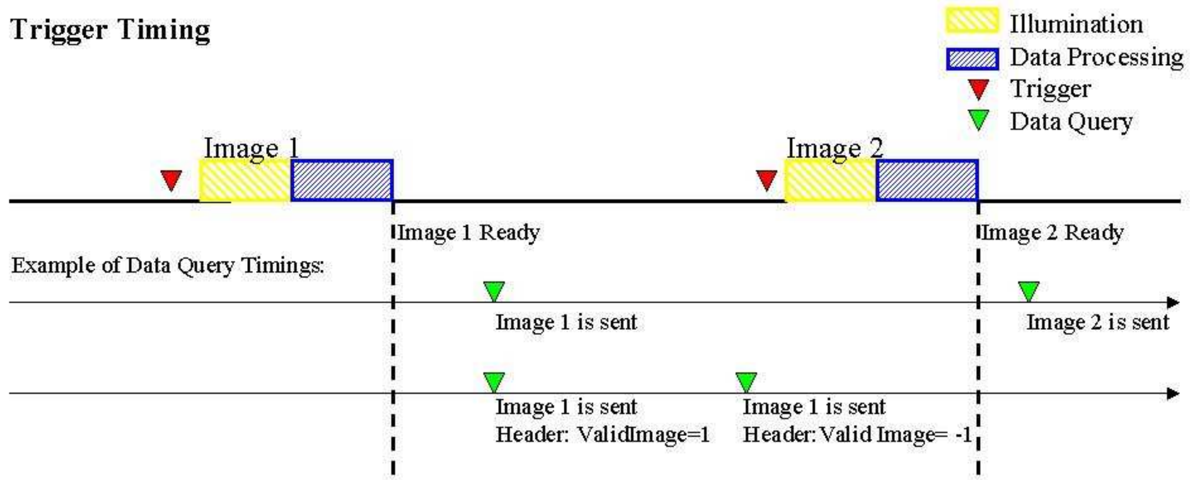
$$\text{ifmtime} = \frac{1}{\text{framerate}} - \text{evaluationtime}$$

Since the sensor has a powerful illumination it will heat up. There is a temperature protection, so the sensor can not be damaged in a regular environment.
Whenever data is queried by sending a string of lowercase letters (e.g. 'xyz') on the data channel, the camera responds the data of the last exposure time. All image types in the query-string derive from the same exposure time.
If the time interval between two queries is long, all images except the last one taken are lost.
If the time interval between two queries is short, the same image is sent twice. The header of

the second dataset will indicate in the ValidImage entry if the image has already been sent. Both headers will have the same timestamp.
In order to synchronize with the free-running sensor it is recommended to use a string starting with a capital letter (e.g. 'Xyz'). The capital letter 'X' indicates that the data must come from a picture acquisition that has not been sent before. The following lower case letters mean that the 'yz'-values come from the same exposure time as the 'X'-values.
If the time interval between two queries (e.g. 'Xyz' and 'Xyz') is short, the answer of the second query is delayed until the data of the next image acquisition is available. If the query-string is 'XYZ', the 'X', 'Y' and 'Z'-values come from consecutive images.



In triggered mode, the camera takes images only upon a trigger event. The trigger can be issued either through hardware (positive edge/negative edge on Pin 2) or software (XML-RPC-command). The data query is independent of the trigger event and is operating in analogy to the freerun mode. If a capital letter is queried and no trigger-event has happened, no data will be sent.



The AverageDetermination-Setting determines the number of images, that are averaged before the data processing is carried out. The image acquisition is automatically carried out at maximum framerate.

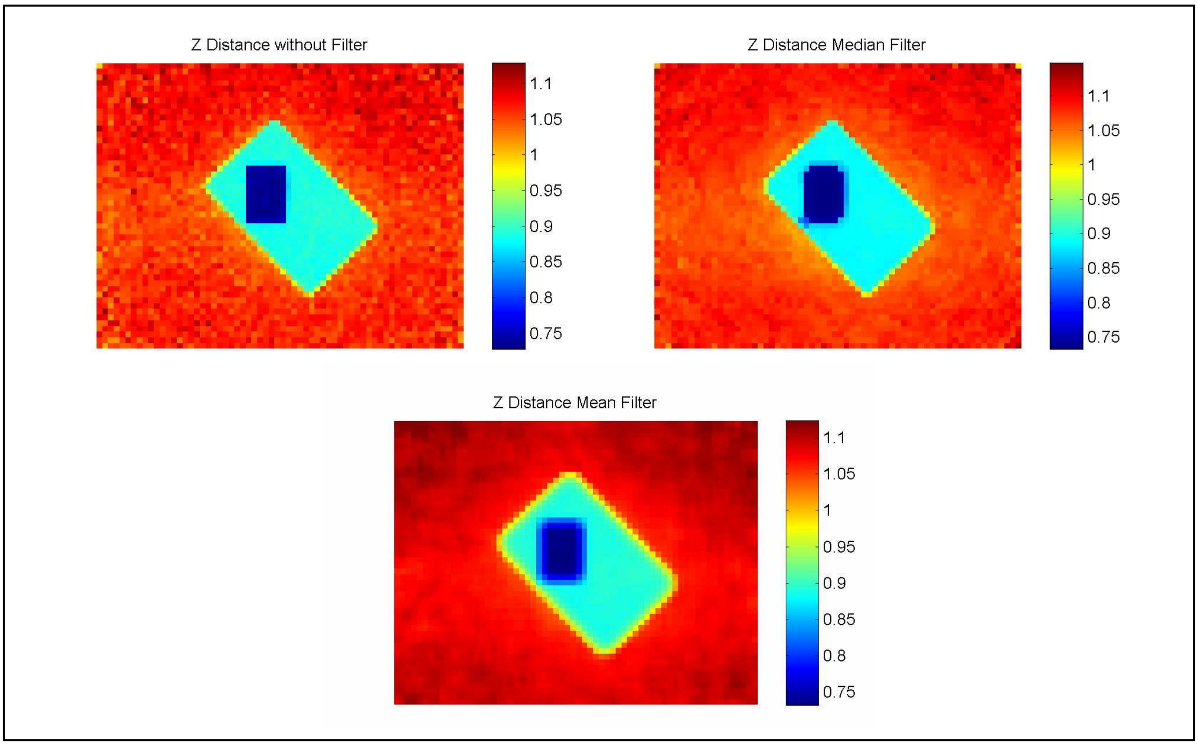
1.6. Temperature Limit

If the temperature limit is exceeded, the sensor stops the illumination and issues a temperature error. It is still operating and will respond to any incoming XML-commands, but it cannot take new images. If an image is queried the last valid image is sent. The image header (valid

image) is set to zero, indicating that the image is old. After cooling down the sensor automatically restarts the illumination.
A temperature error is likely to occur at high integration times and fast framerates together with a poor thermal coupling of the mounting.

1.7. Image Processing

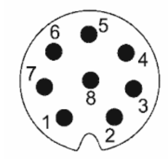
There is a choice of several data filters available. There are two categories, one performing in the spatial and one in the temporal domain.



The spatial filters act on one image, so that each measurement is independent of its predecessors. The median filter is the default filter, since it edge-conserving and still provides reasonable filtering. Even more powerful is the mean-filter, which smoothes edges.

1.8. Process Connection

The process connection of the O3D is the 8-pin-connector. In addition to the power supply there are 6 pins labeled Trigger, Ready, IN1, IN2, OUT1 and OUT2. The electrical characteristics of the digital input lines are OFF=0V and ON=24V/3mA. The digital outputs are OFF=0V and ON=24V/max100mA. The analog output range is 0V to 10V/max10mA (min. burden 1kΩ) in voltage configuration and 4mA to 20mA/max10V (max. burden 500Ω) in current configuration.



1- +24V
2- Trigger-input
3- GND
4- OUT1 / Analog out
5- Ready
6- OUT2
7- IN1
8- IN2

The process connection pins are controlled through 8bit-register settings. The sensor has to be in framegrabbermode (`MDAXMLSetProgram=7`), otherwise the sensor software periodically overwrites the register values. Registers contain data for read and for write. The sensor may modify the content of the registers. It will therefore usually be the case after having set a

register value (e.g. Reg37=0), that reading the register will return a different value (e.g. Reg37=8).
The registers can be accessed through the functions `xmlOPS_GetIORegister(REGADDR)` and `xmlOPS_SetIORegister(REGADDR,VAL)` (see 2.1 for further details).

Digital Input:
The state of all digital inputs is represented by the content of Reg41:

Bit	7	6	5	4	3	2	1	0
Short-name	x	x	Trigger	In2	In1	x	x	x
Resetvalue	0	0	0	0	0	0	0	0

The Trigger pin is a digital input only. If the image acquisition timing is hardware trigger, the trigger signal has to be connected to the trigger pin, otherwise the pin is free to be used as input pin (Reg41, Bit 5).
The IN1- and IN2-pin can be digital input pins. The input signal can be obtained through register Reg41, Bit3 and Bit4.

Digital Output:
The IN1-, IN2-, OUT1- and OUT2-pin can be digital output pins. The 8bit output register controls the timing of the output pin. A register value 0 sets the pin permanently OFF and a 255 sets the pin permanently ON. Other values generate an ON-Pulse, where the pulselength is the registervalue times 10ms, eg. a registervalue of 10 sets the pin ON for 100ms and then OFF.

OUT1:
The OUT1 configuration can be digital input, analog output current or analog putput voltage. The configuration is controlled through various registers:

- Digital Out Reg37=00, Reg22=0 Reg36 = Value
- Analog Current Reg37=16, Reg22=0, Reg36=0 Reg23 = Value
- Analog Voltage Reg37=144, Reg22=0, Reg36=0 Reg23 = Value

The configuration has to be set once prior to setting the value. The analog value is represented by an 8bit-value (0...255) in register Reg23, with 0 being the lowest output (0V/4mA) and 255 the highest output (10V/20mA).

Summary:

Function	**Pin #**	**Digital IN 0V - 24V/3mA**	**Digital OUT 0V - 24V/100mA**	**Analog OUT CURRENT 4mA – 20mA**	**Analog OUT VOLTAGE 0V – 10V**
Trigger	2	**Reg41, Bit5** 1=ON 0=OFF			
OUT1	4		**Reg37 = 0** **Reg22 = 0** DIGITAL OUT **Reg36** 0 =OFF 1..254=ON-Pulse 255 =ON	**Reg37 = 16** **Reg22 = 0** **Reg36 = 0** ANA. CURRENT **Reg23** 0...255 = 4...20mA	**Reg37 = 144** **Reg22 = 0** **Reg36 = 0** ANA. VOLTAGE **Reg23** 0...255 = 0...10V
OUT2	6		**Reg35**		

			0 =OFF 1..254=ON-Pulse 255 =ON		
IN1	7	**Reg41, Bit3** 1=ON 0=OFF	**Reg32** 0 =OFF 1..254=ON-Pulse 255 =ON		
IN2	8	**Reg41, Bit4** 1=ON 0=OFF	**Reg33** 0 =OFF 1..254=ON-Pulse 255 =ON		

2. XML-RPC Functions

In order to provide access to the sensor's image data and to configure the sensor, an interface is provided using the XML-RPC protocol [1]. If you are implementing software to communicate with the sensor you have to create a XML-RPC client to talk to the sensor's server. A free XML-RPC library is available [2].
The server listens for a connection on port 8080 by default. This value can be changed. The following XML-RPC functions may be called on the server side:
Each function name is to be sent to the server with a set of appropriate XML-RPC values (arguments and results) having a syntax like

```
pXmlClient->execute ("xml-rpc_function_name", args, result);
```

The syntax depends on the used XML-RPC library. The code examples below refers to Version 0.7 of XmlRpc++ by Chris Morley.

`xml-rpc_function_name` is a pointer to a constant char array containing the function to be called in the sensor.
`args` is an array of XML-RPC values passed as parameters to the sensor.
`result` is a XML-RPC value array containing the response of the server after returning from the executed command.

2.1. Sensor Connection Settings

`MDAXMLConnectCP`
has to be called at the beginning of a XML-RPC session

args[0]	IP-address of the client
args[1]	parameter is mandatory. 0=HeartBeat ON, 1=HeartBeat OFF; If ON MDAXMLHeartbeat function has to be called once within 10 seconds.
result[0]	Error code, 0 for no error
result[1]	firmware version
result[2]	sensor type

`MDAXMLDisconnectCP`

args[0]	IP-address of the client
result[0]	Error code, 0 for no error

`MDAXMLGetIP`

args	no arguments
result[0]	Error code, 0 for no error
result[1]	IP String

`MDAXMLSetIP`

args[0]	0
args[1]	0
args[2]	IP String
result[0]	Error code, 0 for no error , Modification becomes effective after reboot!

`MDAXMLGetSubNetmask`

args	no arguments
result[0]	Error code, 0 for no error
result[1]	IP String

`MDAXMLSetSubNetmask`

args[0]	0
args[1]	0
args[2]	IP String
result[0]	Error code, 0 for no error, Modification becomes effective after reboot!

MDAXMLGetGatewayAddress

args	no arguments
result[0]	Error code, 0 for no error
result[1]	IP String

MDAXMLSetGatewayAddress

args[0]	0
args[1]	0
args[2]	IP String
result[0]	Error code, 0 for no error, Modification becomes effective after reboot!

MDAXMLGetDHCPMode

args	no arguments
result[0]	Error code, 0 for no error
result[1]	1=DHCP-Mode 0=Static IP

MDAXMLSetDHCPMode

args[0]	0
args[1]	0
args[2]	1=DHCP-Mode 0=Static IP
result[0]	Error code, 0 for no error, Modification becomes effective after reboot!

MDAXMLGetXmlPortCP

args	no arguments
result[0]	Error code, 0 for no error
result[1]	Current port number from 1 to 65535, default 8080

MDAXMLSetXmlPortCP

args[0]	Valid port number from 1 to 65535, default 8080
result[0]	Error code, 0 for no error, Modification becomes effective after reboot!

MDAXMLGetTCPPortCP

args	no arguments
result[0]	Error code, 0 for no error
result[1]	Current port number from 1 to 65535, default: 50002

MDAXMLSetTCPPortCP

args [0]	Valid port number from 1 to 65535, default 50002
result[0]	Error code, 0 for no error, Modification becomes effective after reboot!

MDAXMLHeartBeat

args	no arguments
result[0]	Error code, 0 for no error

xmlOPS_GetIORegister

args[0]	register address to read
result[0]	Error code, 0 for no error
result[1]	8 bit register content

xmlOPS_SetIORegister

args[0]	register address to write
args[1]	8 bit value to be written If MDAXMLSetProgram is not set to 7, this value can be overwritten by the program
result[0]	Error code, 0 for no error
result[1]	data that was written

2.2. *Image Acquisition Settings*

`MDAXMLSetWorkingMode`

`args[0]`	parameter contains a 1 to turn on the image server and a 0 to turn it off.
`result[0]`	Error code, 0 for no error
`result[1]`	TCP Port

`MDAXMLGetFrontendData`

`args`	no arguments
`result[0]`	Error code, 0 for no error
`result[1]`	Integer, always 0
`result[2]`	Modulation Frequency 0 (23MHz) 1 (20.4MHz) 2 (20.6MHz)
`result[3]`	Integer, 0 for single sampling an 1 for double sampling mode
`result[4]`	Integer, always 0
`result[5]`	Integer value between 1 and 5000 representing the integration time in microseconds for the first integration time (short).
`result[6]`	Integer value between 1 and 5000 representing the integration time in microseconds for the second integration time (long)
`result[7]`	Integer, 20 always
`result[8]`	Actual Inter Frame Mute Time

`MDAXMLSetFrontendData`

`args[0]`	Integer, always 0
`args[1]`	Modulation Frequency 0 (23MHz) 1 (20.4MHz) 2 (20.6MHz)
`args[2]`	Integer, 0 for single sampling an 1 for double sampling mode (default 1)
`args[3]`	Integer, always 0
`args[4]`	Integer value between 1 and 5000 representing the integration time in microseconds for the first integration time. (default 125) INT1 < INT2
`args[5]`	Integer value between 1 and 5000 representing the integration time in microseconds for the second integration time (default200) INT1 < INT2
`args[6]`	Integer, always 20
`args[7]`	Inter Frame Mute Time in microseconds. Controles the Framerate.Integer value between 0 and 10000.
`result[0]`	Error code, 0 for no error
`result[1]`	Error hint: bit coded check of parameters
`result[2]`	Modulation Frequency 0 (23MHz) 1 (20.4MHz) 2 (20.6MHz)
`result[3]`	Integer, 0 for single sampling an 1 for double sampling mode
`result[4]`	Integer, always 0
`result[5]`	Integer value between 1 and 5000 representing the integration time in microseconds for the first integration time.
`result[6]`	Integer value between 1 and 5000 representing the integration time in microseconds for the second integration time
`result[7]`	Integer, always 20
`result[8]`	Actual Inter Frame Mute Time in microseconds.
	When sending configuration data: The user may increase the sampling rate, if the mounting of the sensor ensures enough heat dissipation. If the sensor temperature exceeds a critical value, the illumination switches off and the temperature alarm is set. The sensor is still operational and will respond the last valid data, but indicating that the data is not valid in the header. When the sensor cooled off the illumination automatically resumes work.

`MDAXMLGetTrigger`

`args`	no arguments
`result[0]`	Error code, 0 for no error
`result[1]`	Currently used trigger type. 1 for positive edge, 2 for negative edge and 3 for free running (hardware trigger from the trigger input pin of the sensor is disabled) and 4

	for software trigger

MDAXMLSetTrigger

args[0]	0
args[1]	0
args[2]	Trigger type. 1 for positive edge, 2 for negative edge, 3 for free running (hardware trigger from the trigger input pin of the sensor is disabled) and 4 for software trigger
result[0]	Error code, 0 for no error

MDAXMLTriggerImage

args	no arguments
result[0]	Error code, 0 for no error

MDAXMLGetDebounceTrigger

args	no arguments
result[0]	Error code, 0 for no error
result[1]	int, 0 if trigger debouncing is turned off

MDAXMLSetDebounceTrigger

args[0]	int, 0 for turning trigger debouncing OFF, for any other number ON
result[0]	Error code, 0 for no error

MDAXMLGetAverageDetermination

args	no arguments
result[0]	Error code, 0 for no error
result[1]	int

MDAXMLSetAverageDetermination

args [0]	0
args [1]	0
args [2]	Number of images, that are being averaged
result[0]	Error code, 0 for no error

2.3. Image Processing Settings

MDAXMLGetProgram

args	no arguments
result[0]	Error code, 0 for no error
result[1]	Program number -> should be 7 for the framegrabber, otherwise IO-Pins can change during operation

MDAXMLSetProgram

args [0]	0
args [1]	0
args [2]	7 for enabling frame grabber modus. Must be called prior to use sensor. Every Call sets the Image Acquistion Parameters to their default values (Integration Times: 2000 + 125, Double Sampling, Continous Trigger, Framerate 4Hz, Median Filter: ON)
result[0]	Error code, 0 for no error

MDAXMLSetMedianFilterStatus

args [0]	int 0 = off, everything else on
result[0]	Error code, 0 for no error

MDAXMLGetMedianFilterStatus

result[0]	Error code, 0 for no error
result[1]	int 0 = off, everything else on

MDAXMLSetMeanFilterStatus

Args [0]	int 0 = off, everything else on
Result[0]	Error code, 0 for no error

MDAXMLGetMeanFilterStatus

Result[0]	Error code, 0 for no error
Result[1]	int 0 = off, everything else on

3. Error Messages

In some communication protocols, the result is filled with an error information. The returned error code can be one of the following:

MEANING	CODE
SENSOR_NO_ERRORS	0
SENSOR_FIRST_ERROR_ID	-100
SENSOR_ALREADY_CONNECTED	-102
SENSOR_NO_ACTIVE_PRODUCT	-103
SENSOR_NOT_PRODUCT_LIST	-301
SENSOR_COULD_NOT_COPY_FILE	-900
SENSOR_COULD_NOT_OPEN_FILE	-901
SENSOR_PRODUCT_NOT_FOUND	-902
SENSOR_SEARCHING_FOR_INDEX	-903
SENSOR_PRODUCT_NAME_EXITS	-904
SENSOR_COULD_NOT_SAVE_XML	-905
SENSOR_ACTIVE_PRODUCT_INVALID	-906
SENSOR_PRODUCT_ID_INCORRECT	-907
SENSOR_SEARCHING_PRODUCT	-908
SENSOR_COULD_NOT_COPY_PRODUCT_DIR	-910
SENSOR_INVALID_TRIGGER_MODE	-1000
SENSOR_INVALID_NAME	-1100
SENSOR_INVALID_LOCATION	-1101
SENSOR_INVALID_IP	-1102
SENSOR_INVALID_NETMASK	-1103
SENSOR_INVALID_GATEWAY	-1104
SENSOR_INVALID_XMLRPC_PORT	-1105
SENSOR_INVALID_UDP_PORT	-1106
SENSOR_INVALID_EXT_SWITCH_PARM	-1107
SENSOR_INVALID_DEBOUNCED_TRIGGER	-1108
SENSOR_INVALID_LASER_MODE	-1109
SENSOR_INVALID_MAC_ADDRESS	-1110
SENSOR_INVALID_SW_VERSION	-1111
SENSOR_SAVING_SENSOR_PRODUCT	-1150
SENSOR_SAVING_NETWORK_PARMS	-1151
SENSOR_SAVING_GLOBAL_SETTINGS	-1152
SENSOR_WRITING_IP_ADDRESS	-1171
SENSOR_WRITING_NETMASK	-1172
SENSOR_WRITING_GATEWAY	-1173
SENSOR_GENERALDATABASE_XML_NOT_OPENED	-1200
SENSOR_READING_GENERALDATABASE_XML	-1201
SENSOR_INVALID_COMMON_INFO	-1202
SENSOR_INVALID_NETWORK_PARMS	-1203
SENSOR_OPENING_FILE	-1500
SENSOR_INVALID_PRODUCT_UPLOADED	-1501
SENSOR_RESULT_ID_NOT_AVAILABLE	-1600
SENSOR_IMAGER_NO_MEM	-1700
SENSOR_IMAGER_OPENING	-1701
SENSOR_IMAGER_NOT_OPENED	-1702
SENSOR_IMAGER_CAPTURING	-1703
SENSOR_IMAGER_GETTING_IMG	-1704
SENSOR_IMAGER_SHARED_MEM_CREATION	-1705
SENSOR_FTW_ERROR	-1800

SENSOR_PATH_ERROR	-1801
SENSOR_ERROR_HANDLING_FLASH	-1802
SENSOR_ERROR_INVALID_SAMPLINGRATE	-2001
SENSOR_ERROR_GENERAL	-4000
SENSOR_ERROR_SHORT_CIRCUIT	-4001
SENSOR_ERROR_OVER_TEMP	-4002
SENSOR_ERROR_HARDWARE	-4003
SENSOR_NO_VALID_IMAGE	-7001
SENSOR_INVALID_PARAM	-7200
SENSOR_SP1_EQUAL_SP2	-7201
SENSOR_SP2_EQUAL_SP1	-7202
SENSOR_SP1_TOO_SMALL	-7203
SENSOR_SP1_TOO_BIG	-7204
SENSOR_SP2_TOO_SMALL	-7205
SENSOR_SP2_TOO_BIG	-7206
SENSOR_ANALOG_STARTPOINT_TOO_BIG	-7207
SENSOR_ANALOG_ENDPOINT_TOO_SMALL	-7208
SENSOR_PRODUCT_NOT_OPENED	-7300

4. Data Interface

If turned on, the data-server listens to a blocking socket (default 50002) for a socket connection. If a connection is established (only one at a time), it waits for a querystring eg 'id' of 'ID'. The following image types can be requested singularly or in combination:

Imagetype	String	Return Value
radial distance	d/D	1
grey image	i/I	2
x coordinate of line of sight vector	e/E	5
y coordinate of line of sight vector	f /F	6
z coordinate of line of sight vector	g/G	7
cartesian x coordinates	x/X	8
cartesian y coordinates	y/Y	9
cartesian z coordinates	z/Z	10
stop the socket connection	q	

The string always has to be sent prior to waiting for the data on the client side. The order of the different images is not fixed. Every demanded image-type is sent with a header and data. The size of the header is 376 bytes (94 float32 values=94*4 byte) and has the following format:

Beware that the byte order of the header plus image data is in big-endian.

```
typedef struct ImageHeaderInformation {
/** @brief Overall data size in Bytes: header + image */
float       DataSize;
/** @brief Size of the header */
float       HeaderSize;
/** @brief type of image cf. IMAGE_HEADER::ImageTypes */
float       ImageType;
/** @brief consecutive version number */
float       Version;
/** @brief single or double integration */
float       SamplingMode;
/** @brief illu status 0,1,2,3 bit coded */
float       IlluMode;
/** @brief frequency mode cf. ModulationFrequency */
float       FrequencyMode;
/** @brief unambiguous range of current frequency */
float       UnambiguousRange;
/** @brief time needed by image evaluation [ms] */
float       EvaluationTime;
/** @brief first integration time single sampling mode [ms] */
float       IntegrationTime_Exp0;
/** @brief second integration time double sampling mode [ms] */
float       IntegrationTime_Exp1;
/** @brief timestamp */
T_TIMESTAMP TimeStamp;
/** @brief median filter status */
float       MedianFilter;
/** @brief mean filter status */
float       MeanFilter;
float       internal_a1;
float       internal_a2;
float       internal_a3;
float       internal_a4;
/** @brief displays if image is valid or not */
float       ValidImage;
```

```
float        ErrorCode;
float        internal_b1;
float        internal_b2;
float        internal_b3;
/** @brief configured trigger mode */
float        CurrentTriggerMode;
float        internal_c1;
float        internal_c2;
float        internal_c3;
float        internal_c4;
/** @brief Inter Frame Mute time*/
float        IfmTime;
float        internal_d1;
float        internal_d2;
float        internal_d3;
float        internal_d4;
float        internal_d5;
float        internal_d6;
float        internal_d7;
float        internal_d8;
float        internal_d9;
float        internal_d10;
float        internal_d11;
float        internal_d12;
float        internal_d13;
float        internal_d14;
float        internal_d15;
float        internal_d16;
float        internal_d17;
float        internal_d18;
float        internal_d19;
float        internal_d20;
float        internal_d21;
float        internal_d22;
float        internal_d23;
float        internal_d24;
float        internal_d25;
float        internal_d26;
float        internal_d27;
float        internal_d28;
float        internal_d29;
float        internal_d30;
float        internal_d31;
float        internal_d32;
float        internal_d33;
float        internal_d34;
float        internal_d35;
float        internal_d36;
float        internal_d37;
float        internal_d38;
float        internal_d39;
float        internal_d40;
float        internal_d41;
float        internal_d42;
float        internal_d43;
float        internal_d44;
float        internal_d45;
float        internal_d46;
float        internal_d47;
float        internal_d48;
float        internal_d49;
float        internal_d50;
```

```
float        internal_d51;
float        internal_d52;
float        internal_d53;
float        internal_d54;
float        internal_d55;
float        internal_d56;
float        internal_d57;
float        internal_d58;
float        internal_d59;
float        internal_d60;
float        internal_d61;
float        internal_d62;
float        internal_d63;
float        internal_d64;
}T_IMAGEHEADER;
```

Possible values for the “ImageType” attribute are:

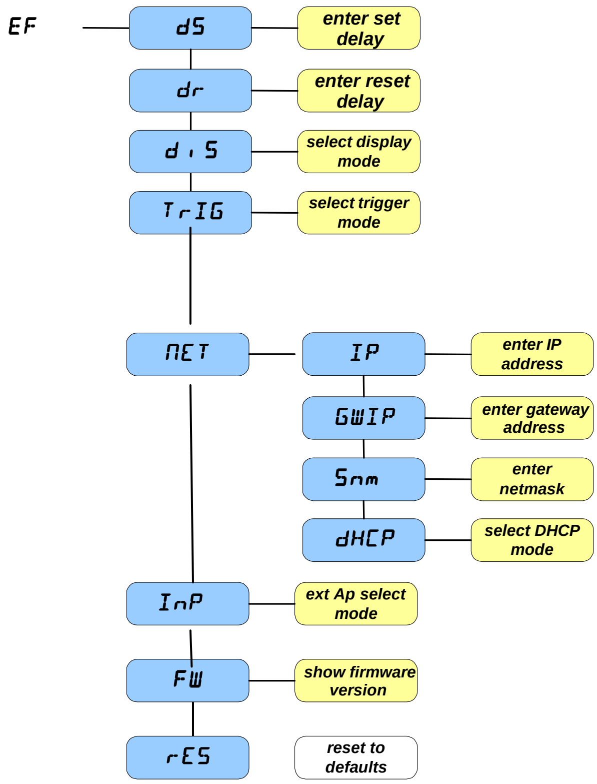
```
enum ImageTypes
{
         INVALID_IMAGE = 0,
         DISTANCE_IMAGE,
         INTERNAL_DATA_A,
         AMPLITUDE_IMAGE,
         INTERNAL_DATA_B,
         NORMAL_X_IMAGE,
         NORMAL_Y_IMAGE,
         NORMAL_Z_IMAGE,
         KARTESIAN_X_IMAGE,
         KARTESIAN_Y_IMAGE,
         KARTESIAN_Z_IMAGE,
         INTERNAL_DATA_C,
         SEGMENTATION_IMAGE
};

typedef struct TimeStamp
{
    float Seconds;
    float Useconds;
}T_TIMESTAMP;
```

Beware that the byte order of the header plus image data is in big-endian.

5. Sensor teach menu

For further parameterisation the sensor provides a two-button interface. For entering the menu push the MODE/ENTER button. For navigating through the first hierarchy of the menu use the same button. For entering in a lower menu hierarchy point (like NET) push the Set-button. For changing a value push the Set button more than 5 seconds.



EF	Menu point for extra settings
dS	Here you can enter a value between 0 or 5 seconds in 100 ms steps for a time in which the logical output has to be high until it will be put on the output
dr	Here you can enter a value between 0 or 5 seconds in 100 ms steps for a time in which the logical output has to be low until it will be put on the output.
diS	This menu point is used to select the display mode. d1 - d3 is for normal mode, rd1 - rd3 for a 180° reversed mode. The numbers 1 to 3 define how often the measured value shall be updated in the display: • 1 ⇒ 50ms • 2 ⇒ 200ms • 3 ⇒ 600ms
TrIG	Here the trigger mode of the sensor is selected. • PoS ⇒ trigger on positive edge • nEG ⇒ trigger on negative edge • Frun ⇒ no trigger needed, free run mode
NET	Menu point for network settings
IP	Here you can change the IP-address of the sensor
GWA	Here you can change the gateway-address of the sensor
SNM	Here you can change the subnetmask of the sensor
dHCP	Turn the dhcp mode eihter on or oFF .
InP	Turn the external application switching on or of (on/oFF). If turned on, the logical state of the input connectors of the sensor will be analyzed.
SW	This will show the software version of the sensor.
rES	Reset the sensor to factory settings.

5.1. How to change the IP-address

This section describes how the two button menu is used to change the IP-address of the sensor. Changing the gateway-address and the subnetmask follows the same procedure. Make sure no client is connected to the sensor using the XML-RPC interface.

- Press the MODE/ENTER-button → EF will appear in the display.
- Press the Set-Button to enter extended function menu.
- Press the MODE/ENTER-button until NET is shown.
- Press the SET-button → IP is shown.
- Press the SET-button → the first group of the IP-address will appear, a leading A plus 3 digits. If no button is pressed, the next group is shown after 5 seconds.
- Press the SET-button for more than 5 seconds. → A is blinking

Now you are in edit-mode. Blinking indicates the cursor. Waiting for more than 5 seconds will move the cursor to the right. Pressing SET-Button enters a new value. Pressing MODE/ENTER leaves the edit mode.

- Press the SET-button, to switch to the next group (b) or to increase the digit by 1.
- When all values are set to the appropriate value, press the MODE/ENTER-button to leave the edit mode. Blinking resumes.
- Press MODE/ENTER until WAIT is shown
- Wait until the sensor is operating again. If the power is interupted at this stage, the sensor can be corrupted.
- Reboot the Sensor to activate the new settings

6. References

[1] http://www.xmlrpc.com
[2] http://sourceforge.net/projects/xmlrpcpp