EN

weCat3D MLSL & MLWL

2D/3D Profile Sensors

Operating Instructions

Translation of the Original Operating Instruction
Subject to change without notice
Available as PDF version only
Status: 20.02.2024
Version: 4.2.0
www.wenglor.com

Table of Contents

1. Change Index of Operating Instructions

Version	Date	Description/Change	Firmware/ Software version
1.0.0	20.06.2016	Initial version of the operating instructions	FW: 1.0.0 FW: 1.0.1
1.1.0	25.09.2017	• Updating of laser warnings • Use of the MLSL2 for intended purpose • Technical data for the MLSL • Measuring fields of the MLSL2 • MLSL2 housing dimensions • MLSL1/MLSL2 layout • LED display with LED laser • Link/Act LED description • M8 tightening torque • S74 connection technology • MLSL1/MLSL2 system overview • External 24 V laser shutdown • Default settings: direction of rotation, signal selection • Note regarding programming interfaces • Web server optimization • Web server updating: » Laser status » Measuring rate » ROI, profile and trigger settings • Using more than one sensor • Firmware update • OLED display: encoder, display (rotate), configuration	FW: 1.1.0
1.2.0	28.06.2018	• Technical addenda • Temperature specifications updated • “Firmware Update” section removed	FW: 1.1.0
1.2.1	12.12.2018	• Begin of measuring rate • Change in descriptions: measuring range X, measuring field	FW: 1.1.0
1.3.1	06.03.2019	• Correction of connection diagram • Update of layout graphics • Description OPT3013	FW: 1.1.1
1.4.0	18.04.2019	• New: Modus Live Image	FW: 1.1.4
1.5.0	26.06.2019	• Temperature switch-off	FW: 1.1.6
1.6.0	24.09.2019	• Laser Warnings	FW: 1.1.6
1.7.0	03.12.2019	• Description "Extended source" • Measuring rate updated • Comment service life	FW: 1.1.6

Version	Date	Description/Change	Firmware/ Software version
1.7.1	21.01.2019	• Amendment service life of laser	FW: 1.2.0
1.7.2	25.03.2020	• Dimensional Drawings MLSL1xxx • Adaption Electrical Connection	FW: 1.2.0
1.7.3	22.06.2020	• Addition to the technical data	FW: 1.2.0
1.8.0	27.10.2020	• Addition OPT3042 (Attachment)	FW: 1.2.2
1.9.0	29.03.2021	• Adaption measuring field MLWL2x5	FW: 1.2.2
2.0.0	04.05.2021	• New operating mode Smart weCat3D • Addition MLWL033	FW: 2.0.0
3.0.0	17.03.2022	• Adaption of Limitation of Liability • Implementation of DLL und GigE Vision Interface Protocol • Assembly instructions for cooling units • Assembly instructions for screening grid retainers • Cable outlet with angle plugs • Adaption of system overviews	FW: 2.1.0 SDK: 1.5.0 GigE: 2.0.1
3.0.1	28.03.2022	• Adaption current consumption of MLZLxxx • Bugfix SDK: Reduction of reconnect time • Bugfix SDK: Sync error after sending two “SetAcquisitionStart” commands	FW: 2.1.0 SDK: 1.5.1 GigE: 2.0.1
3.0.2	27.04.2022	• Indication of protective foil ZLSE010	FW: 2.1.0 SDK: 1.5.1 GigE: 2.0.1
3.0.3	12.05.2022	• Adaption "Reset sensor settings" • Adaption of section 11.6.14 and 11.6.15	FW: 2.1.0 SDK: 1.5.1 GigE: 2.0.1
3.0.4	16.05.2022	• Note pixel format Mono16, section 12.3.1	FW: 2.1.0 SDK: 1.5.1 GigE: 2.0.1
3.1.0	25.10.2022	• Adaption webserver • Added new SDK functions • Added UDP connection • Added specifications MLZL • Adaption of section 7.3 • Added sensor group M2SL	FW: 2.3.0 SDK: 1.6.0 GigE (external): 2.0.1 GigE (embedded): 2.1.1
3.2.0	19.04.2023	• Note on degree of protection, section 4	FW: 2.3.0 SDK: 1.6.0 GigE (external): 2.0.1 GigE (embedded): 2.1.1
4.0.0	20.09.2023	• Adding section „6.3.3 Safe Laser Shutdown (EN ISO 13849-2)“ • Several corrections • Update of dimensional drawings	FW: 2.3.0 SDK: 1.6.0 GigE (external): 2.0.1 GigE (embedded): 2.1.1

Version	Date	Description/Change	Firmware/ Software version
4.0.1	04.10.2023	• Minor Bugfixes	FW: 2.3.0 SDK: 1.6.0 GigE (external): 2.0.1 GigE (embedded): 2.1.1
4.1.0	27.11.2023	• Adding MLSL123S50 (section 9.1.3) • Bugfix GetScannerState (section 10 and 12) • Note on output of data type (section 10.7)	FW: 2.3.0 SDK: 1.6.0 GigE (external): 2.0.1 GigE (embedded): 2.1.1
4.2.0	20.02.2024	• Additions and bugfix Technical Data (sections 4 and 9.1.1.1)	FW: 2.3.0 SDK: 1.6.0 GigE (external): 2.0.1 GigE (embedded): 2.1.1

2. General

2.1 Information Concerning these Instructions

- These instructions enable safe and efficient use

of the standard products
  - » MLxLxxx

of the application dedicated sensors
  - » MLZLxxx
  - » M2SLxxx
  - » MLSL123S50

and of special devices
  - » MLWL033
  - » OPT3013
  - » OPT3042

- These instructions are an integral part of the product and must be kept on hand for the entire duration of its service life.
- Local accident prevention regulations and national work safety regulations must be complied with as well.
- The product is subject to further technical development, and thus the information contained in these operating instructions may also be subject to change. The current version can be found at www.wenglor.com in the product's separate download area.

NOTE!
The operating instructions must be read carefully before using the product and must be kept on hand for later reference.

2.2 Explanations of Symbols

- Safety precautions and warnings are emphasized by means of symbols and attention-getting words.
- Safe use of the product is only possible if these safety precautions and warnings are adhered to.

The safety precautions and warnings are laid out in accordance with the following principle:

ATTENTION-GETTING WORD
Type and Source of Danger!
Possible consequences in the event that the hazard is disregarded.
- Measures for averting the hazard.

The meanings of the attention-getting words, as well as the scope of the associated hazards, are listed below.

DANGER!
This word indicates a hazard with a high degree of risk which, if not avoided, results in death or severe injury.

WARNING!
This word indicates a hazard with a medium degree of risk which, if not avoided, may result in death or severe injury.

CAUTION!
This word indicates a hazard with a low degree of risk which, if not avoided, may result in minor or moderate injury.

ATTENTION:
This word draws attention to a potentially hazardous situation which, if not avoided, may result in property damage.

NOTE!
A note draws attention to useful tips and suggestions, as well as information regarding efficient, error-free use.

2.3 Limitation of Liability

- The product has been developed taking into account the state of the art as well as the applicable standards and guidelines.
- We reserve the right to make technical changes.
- A valid declaration of conformity can be found at www.wenglor.com in the download area of the product.
- wenglor sensoric elektronische Geräte GmbH (hereinafter "wenglor") accepts no liability for:
  - » Failure to observe the operating manual,
  - » Unsuitable or improper use of the product,
  - » Excessive use, incorrect or negligent treatment of the product,
  - » Incorrect installation or commissioning,
  - » Use of untrained personnel,
  - » Use of unauthorized spare parts or
  - » Improper or unauthorized changes, modifications or repair work to the products.
- This operating manual does not contain any guarantees/warrantees from wenglor with regard to the processes described or certain product properties.
- wenglor assumes no liability with regard to printing errors or other inaccuracies contained in this operating manual, unless it can be proven that wenglor was aware of the errors at the time the operating manual was created.

2.4 Copyrights

- The contents of these instructions are protected by copyright law.
- All rights are reserved by wenglor.
- Commercial reproduction or any other commercial use of the provided content and information, in particular graphics and images, is not permitted without previous written consent from wenglor.

3. For Your Safety

3.1 Use for Intended Purpose

The weCat3D Profile Sensors are used for profile measurement and they are designed for use in industry and laboratories.

3.2 Function Principle

2D/3D Profile Sensors project a laser line (1) onto the object to be detected (2) and generate an accurate, linearized height profile with an internal camera (3) which is set up at a triangulation angle (4), see Fig. 1. Thanks to its uniform, open interface, the weCat3D series can be incorporated directly by a program library (available for Windows and Linux, see section 10), a TCP/IP socket protocol (see section 11), or the GigE Vision standard (see section 12) without an additional control unit. Alternatively, wenglor offers its own software packages for implementing your application. Individualized selection from a great variety of working ranges, laser classes and light types (red and blue light) ensures maximized flexibility for two and three-dimensional object detection.

The weCat3D offers several different types of sensors:

- MLSL: The standard sensor for most applications
- MLWL: The high performance standard sensor
- M2SL: Sensor with the performance of MLSL, but with a stainless steel housing and IP67/IP69K rating
- MLZL: Sensor with integrated cooling and purging, dedicated for welding applications
- Special devices like OPT3013 as UV version

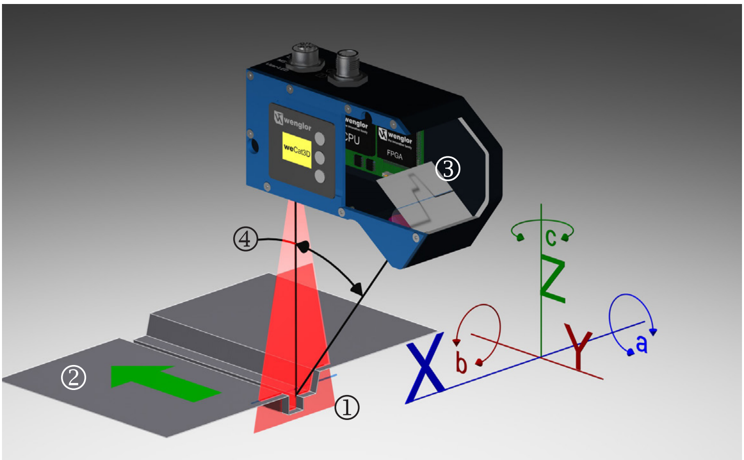


Fig. 1: Function principle of a weCat3D profile sensor

① = Laser line
② = Object
③ = Integrated camera
④ = Triangulation angle

3.3 Use for Other than the Intended Purpose

- The product is not a safety component in accordance with the EG Machinery Directive (except for the profile sensors MLSL2xxS40 with safe laser shutdown, see section 6.3.3).
- The product is not suitable for use in potentially explosive atmospheres.

DANGER!
Risk of personal injury or property damage in case of use for other than the intended purpose!
Use for other than the intended purpose may lead to hazardous situations.
- Instructions regarding use for intended purpose must be observed.

3.4 Personnel Qualifications

- Suitable technical training is a prerequisite.
- In-house electronics training is required.
- Trained personnel who use the product must have uninterrupted access to the operating instructions.
- Valid laser protection requirements must always be adhered to.

ATTENTION!
Risk of personal injury or property damage in case of incorrect initial start-up and maintenance!
Personal injury and damage to equipment may occur.
- Adequate training and qualification of personnel.

3.5 Modification of Products

ATTENTION!
Risk of personal injury or property damage if the product is modified!
Personal injury and damage to equipment may occur. Non-observance may result in loss of the CE marking and the guarantee may be rendered null and void.
- Modification of the product is impermissible.

3.6 General Safety Precautions

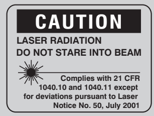
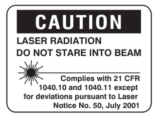
NOTE!

- These instructions are an integral part of the product and must be kept on hand for the entire duration of its service life.
- In the event of possible changes, the respectively current version of the operating instructions can be accessed at www.wenglor.com in the product's separate download area.
- Read the operating instructions carefully before using the product.
- The sensor must be protected against contamination and mechanical influences.

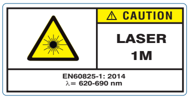
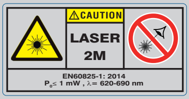
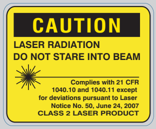
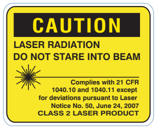
3.7 Laser/LED Warnings

Warning labels are included with the products depending on laser class and type of light. The respective warning labels must be attached to the system in a plainly visible fashion.

3.7.1 Warnings According to Standard EN 60825-1:2007

Laser Class	IEC EN 60825-1	FDA/CFR
Laser Class 1M (EN 60825-1) Applicable standards and safety regulations must be observed.		Not applicable
Laser Class 2M red (EN 60825-1) Applicable standards and safety regulations must be observed.		
Laser Class 2M blue (EN 60825-1) Applicable standards and safety regulations must be observed.		
Laser Class 3R red (EN 60825-1) Applicable standards and safety regulations must be observed. The laser outlet is identified on the device.		
Laser Class 3R blue (EN 60825-1) Applicable standards and safety regulations must be observed. The laser outlet is identified on the device.		
Laser Class 3B blue (EN 60825-1) Applicable standards and safety regulations must be observed. The laser outlet is identified on the device.		

3.7.2 Warnings According to Standard EN 60825-1:2014

Laser Class	IEC EN 60825-1	FDA/CFR
Laser Class 1M (EN 60825-1) Applicable standards and safety regulations must be ovserved.		Not applicable
Laser Class 2M red (EN 60825-1) Applicable standards and safety regulations must be ovserved.		
Laser Class 2M blue (EN 60825-1) Applicable standards and safety regulations must be ovserved.		
Laser Class 3R red (EN 60825-1) Applicable standards and safety regulations must be ovserved. The laser outlet is identified on the device.		
Laser Class 3R blue (EN 60825-1) Applicable standards and safety regulations must be ovserved. The laser outlet is identified on the device.		
Laser Class 3B blue (EN 60825-1) Applicable standards and safety regulations must be ovserved. The laser outlet is identified on the device.		

3.7.3 weCat and Extended Source

The sensors use a line laser. A line laser is an extended source. As a result, the C_6 factor (see IEC EN 60825-1:2014) must be taken into account when evaluating the laser class. Due to the fact that C_6 is greater than or equal to 1, the maximum permissible exposure value (MPE) for thermic retinal hazard is increased by a factor of C_6, assuming that angular extension of the source (measured at the observer's eye) is greater than α_{min}, where α_{min} is equal to 1.5 mrad. And thus as compared with a collimated laser beam, maximum output radiation can be higher for the same laser class.

3.8 Approvals and protection class

NOTE!
Several sensors do not have an UL certification. Please see datasheet of the sensor.

4. Technical Data MLxLxxx

The technical data for M2SL, MLZL and special devices can be found in section 9.

Technical Data / Order No.	MLSL	MLWL
Environmental Conditions		
Ambient temperature	0...45 °C	
Storage temperature	–20…70 °C	
Max. ambient light	5000 Lux	
EMC	DIN EN 61000-6-2; 61000-6-4	
Shock resistance per DIN IEC 68-2-27	30 g / 11 ms	
Vibration resistance per DIN IEC 60068-2-6	6 g (10...55 Hz)	
Athmosperic humidity	5...95 %, non-condensing	
Electrical Data		
Supply Voltage	18...30 V	
Current consumption (Ub = 24 V)[1]	300 mA	300 mA
Measuring rate	200…4000 Hz	175…6000 Hz
Measuring rate (subsampling)	800... 4000 Hz[2]	350...6000 Hz[3]
Number of I/Os	4	
Switching Output Voltage Drop	< 1,5 V	
Switching Output/Switching Current	100 mA	
Can be switched to NC or NO operation	yes	
PNP / NPN / push-pull	yes	
Short-circuit proof	yes	
Reverse polarity protected	yes	
Overload-proof	yes	
Interface	Ethernet TCP/IP	
Transmission speed	100/1000 MBit/s	
Protection class	III	
Integrated web server	yes	
Mechanical Data		
Housing	Aluminum, powder-coated Plastic, ABS	Aluminum, anodised
Optic cover	Plastic, PMMA	Glass
Degree of Protection	IP67[4]	
Connection Type	M12×1; 12-pin.	
Ethernet connector type	M12×1; 8-pin., X-coded	
Connection External 24 V laser shutdown[5]	M12×1, 8-pin	---
Connection safe laser switch-off (EN ISO 13849-2)[6]	M12×1, 8-pin	---

[1] Increased current consumption (1000 mA) for weCat3D MLWL and MLSL2 with laser class 3B.
[2] Subsampling in X and Z
[3] Subsampling in Z
[4] Only valid if all plugs are connected/closed by cables or caps with corresponding protection class.
[5] Only MLSL2xx with laser class 3R and 3B
[6] Only MLSL2xxS40

NOTE!
The warm-up phase lasts roughly 15 minutes.

Order No. / Technical Data	MLSL1x1	MLSL1x2	MLSL1x3	MLSL1x4
Optical Data				
Working range Z	72…108 mm	65…125 mm	90…280 mm	100…500
Measuring range Z	36 mm	60 mm	190 mm	400 mm
Measuring range X	27…34 mm	40…58 mm	62…145 mm	70…280 mm
Resolution Z	3,3…5,2 μm	4,8…9,6 μm	9,4…49 μm	12,4…160 μm
Resolution X	22…28 μm	33…47 μm	54…123 μm	68…246 μm
Temperature drift	2 μm/K	3 μm/K	10 μm/K	20 μm/K
Linearity deviation	18 μm	30 μm	95 μm	200 μm
	0,05 %			
Service life (Tu=+25 °C)*	20.000 h			

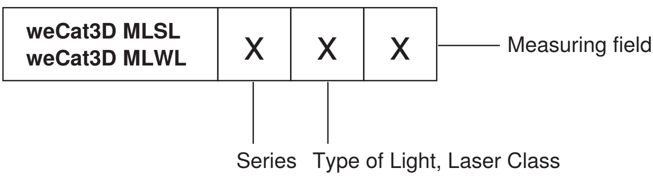
Order No. / Technical Data	MLSL2x5	MLSL2x6
Optical Data		
Working range Z	280…1280 mm	300…1500 mm
Measuring range Z	1000 mm	1200 mm
Measuring range X	200…850 mm	250…1350 mm
Resolution Z	40…570 μm	60…990 μm
Resolution X	190…760 μm	270…1170 μm
Temperature drift	50 μm/K	60 μm/K
Linearity deviation	500 μm	600 μm
	0,05 %	
Service life (ambient temp. = +25° C)*	20.000 h	

* Service life is related to the Laser. Since the Laser is not permanently switched on, the service life increases accordingly.

Technical Data / Order No.	MLWL1x1	MLWL1x2	MLWL1x3	MLWL1x4	MLWL1x5
Optical Data					
Working range Z	70…130 mm	83…213 mm	215…475mm	390…910 mm	600…1 400 mm
Measuring range Z	60 mm	130 mm	260 mm	520 mm	800 mm
Measuring range X	30…52 mm	50…110 mm	150…230 mm	285…455 mm	450…720 mm
Resolution Z	2…4,9 µm	3,2…14 µm	9,6…22 µm	17,8…43 µm	28…67 µm
Resolution X	17…26 µm	26…55 µm	79…120 µm	151…238 µm	235…361 µm
Temperature drift	3 µm/K	6 µm/K	12 µm/K	24 µm/K	37 µm/K
Linearity deviation	15 µm	32,5 µm	65 µm	130 µm	200 µm
	0,025 %				
Service life (Tu=+25 °C)*	20.000 h				

Technical Data / Order No.	MLWL2x1	MLWL2x2	MLWL2x3	MLWL2x4	MLWL2x5
Optical Data					
Working range Z	120…300 mm	120…470 mm	300…1 000 mm	600…2 000 mm	1000…2 500 mm
Measuring range Z	180 mm	350 mm	700 mm	1 400 mm	1 500 mm
Measuring range X	65…145 mm	120…395 mm	280…830 mm	440…1300 mm	850…1300 mm
Resolution Z	5,2…26 µm	8,9…76 µm	27…162 µm	39…289 µm	92…439 µm
Resolution X	36…81 µm	68…198 µm	181…446 µm	251…683 µm	505…1 095 µm
Temperature drift	10 µm/K	16 µm/K	32 µm/K	64 µm/K	70 µm/K
Linearity deviation	45 µm	87,5 µm	175 µm	350 µm	375 µm
	0,025 %				
Service life (Tu=+25 °C)*	20.000 h				

* Service life is related to the Laser. Since the Laser is not permanently switched on, the service life increases accordingly.



Where type of light and laser class are concerned, the “X” in the order number stands for the following variants:

x	Light Source	Laser Class (EN 60825-1)
0	Laser (Red 660 nm)	1M
2	Laser (Red 660 nm)	2M
3	Laser (Blue 405 nm)	2M
4	Laser (Red 660 nm)	3R*
5	Laser (Blue 405 nm)	3R*
7	Laser (Blue 450 nm)	3B*

* Only available for weCat3D MLWL and MLSL2

4.1 Measuring Fields

weCat3D MLSL

weCat3D MLSL1x1

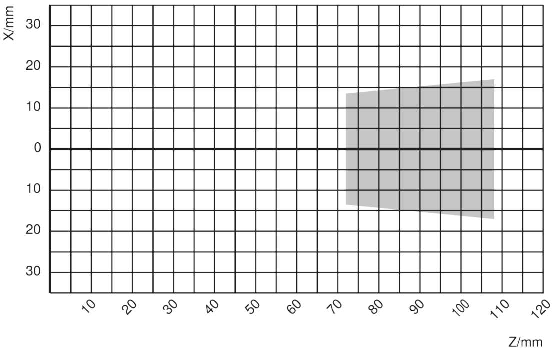


Fig. 2: Measuring field MLSL1x1

weCat3D MLSL1x2

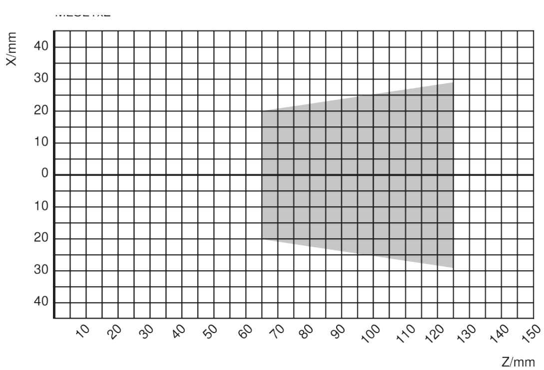


Fig. 3: Measuring field MLSL1x2

weCat3D MLSL1x3

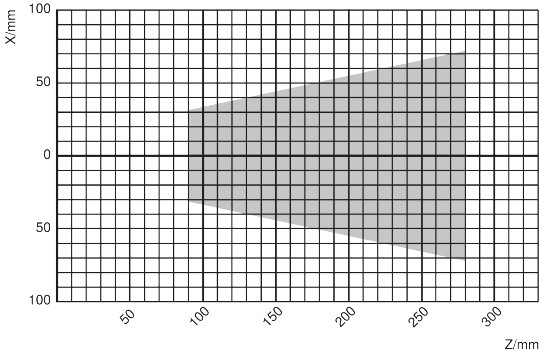


Fig. 4: Measuring field MLSL1x3

weCat3D MLSL1x4

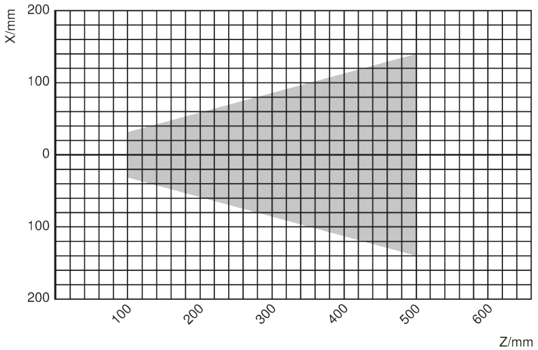


Fig. 5: Measuring field MLSL1x4

weCat3D MLSL2x5

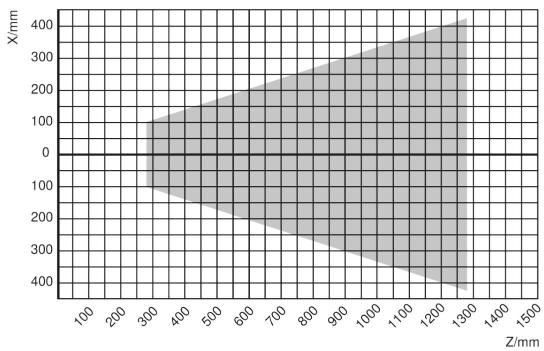


Fig. 6: Measuring field MLSL2x5

weCat3D MLSL2x6

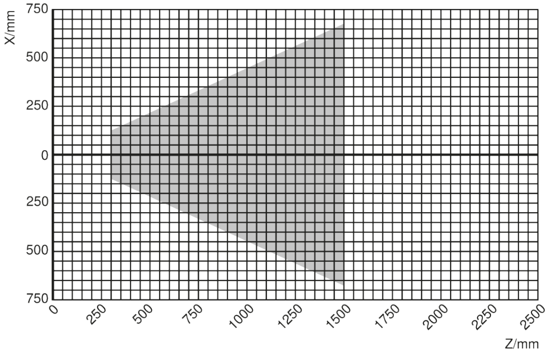


Fig. 7: Measuring field MLSL2x6

weCat3D MLWL

weCat3D MLWL1x1

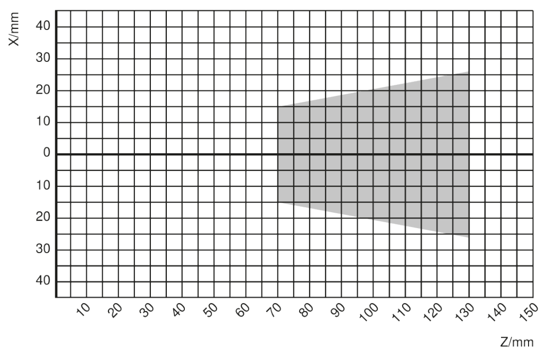


Fig. 8: Measuring field MLWL1x1

weCat3D MLWL1x2

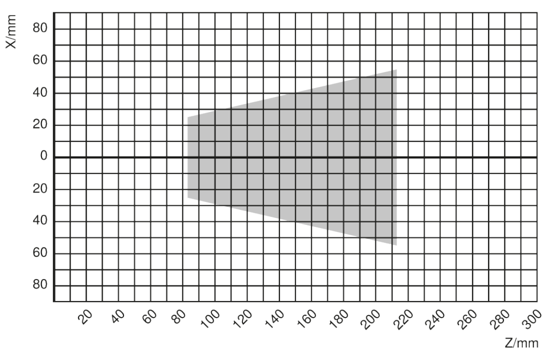


Fig. 9: Measuring field MLWL1x2

weCat3D MLWL1x3

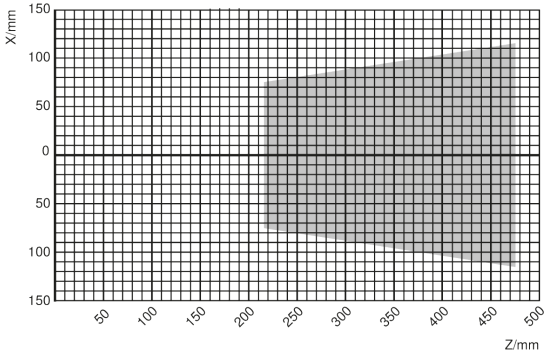


Fig. 10: Measuring field MLWL1x3

weCat3D MLWL1x4

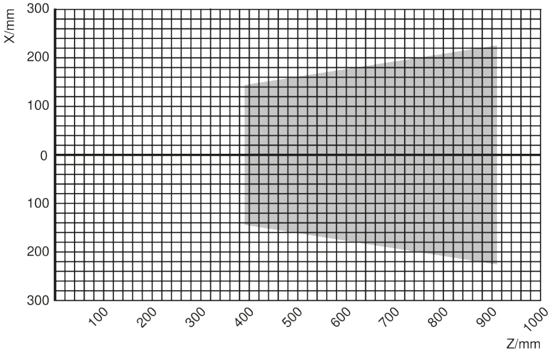


Fig. 11: Measuring field MLWL1x4

weCat3D MLWL1x5

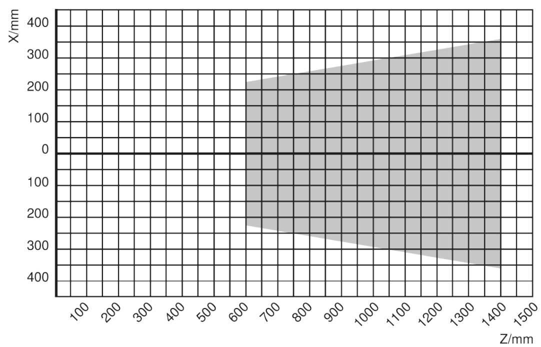


Fig. 12: Measuring field MLWL1x5

weCat3D MLWL2x1

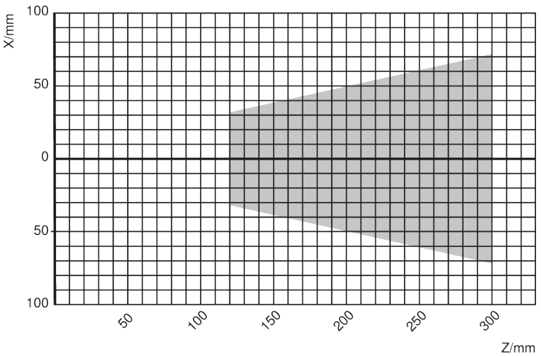


Fig. 13: Measuring field MLWL2x1

weCat3D MLWL2x2

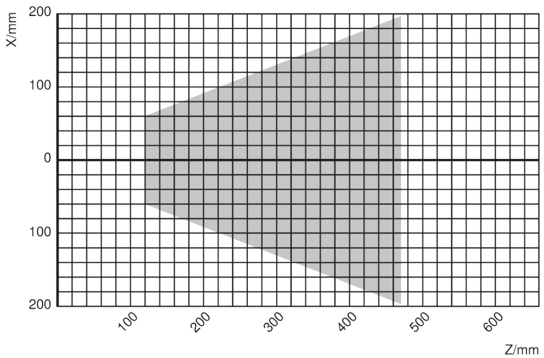


Fig. 14: Measuring field MLWL2x2

weCat3D MLWL2x3

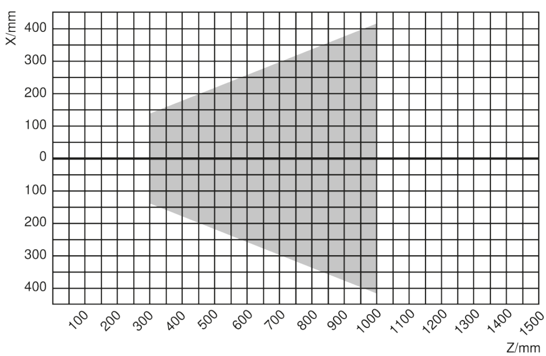


Fig. 15: Measuring field MLWL2x3

weCat3D MLWL2x4

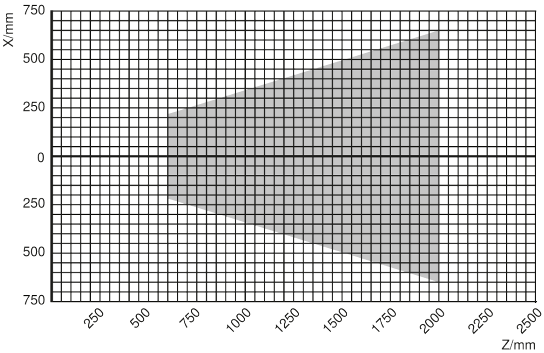


Fig. 16: Measuring field MLWL2x4

weCat3D MLWL2x5

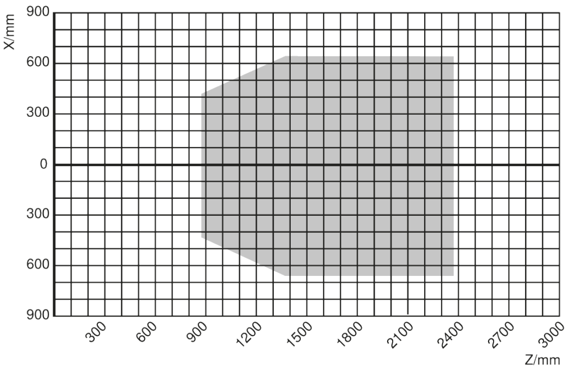


Fig. 17: Measuring field MLWL2x5

4.2 Dimensional Drawings

weCat3D MLSLxxx

weCat3D MLSL1xx

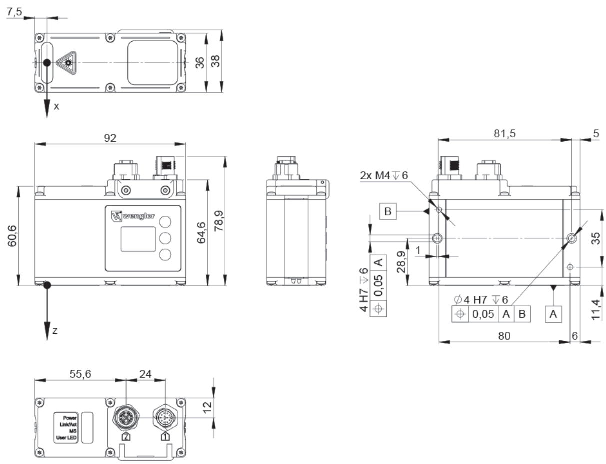
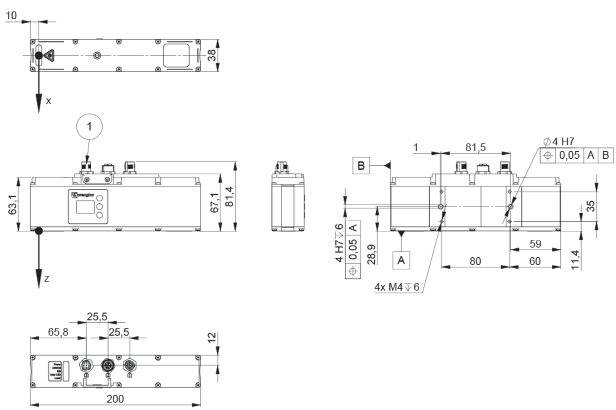


Fig. 18: Dimensional Drawings MLSL1xx

weCat3D MLSL2xx

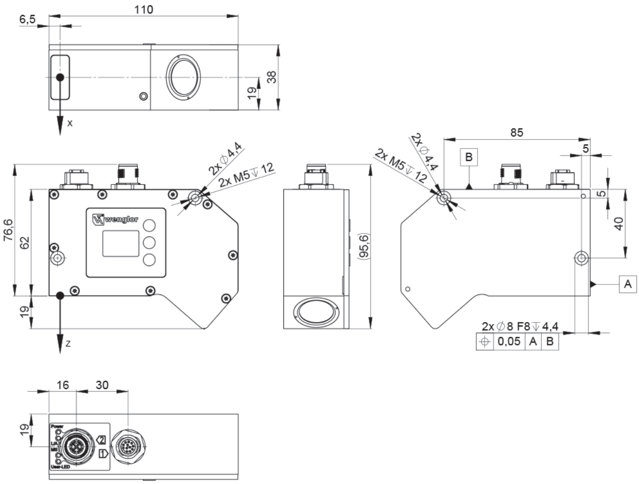


1 = only MLSL2 with laser class 3R and 3B

Fig. 19: Dimensional Drawings MLSL2xx

weCat3D MLWLxxx

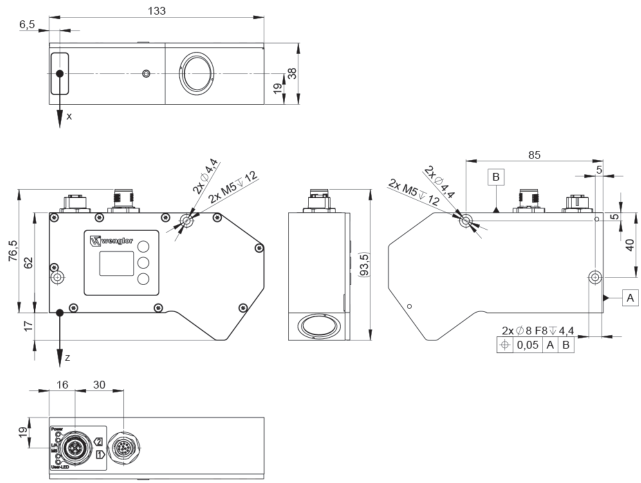
weCat3D MLWL1x1



1 = Recommended mounting position based on the sensor's center of gravity

Fig. 20: Dimensional Drawings MLWL1x1

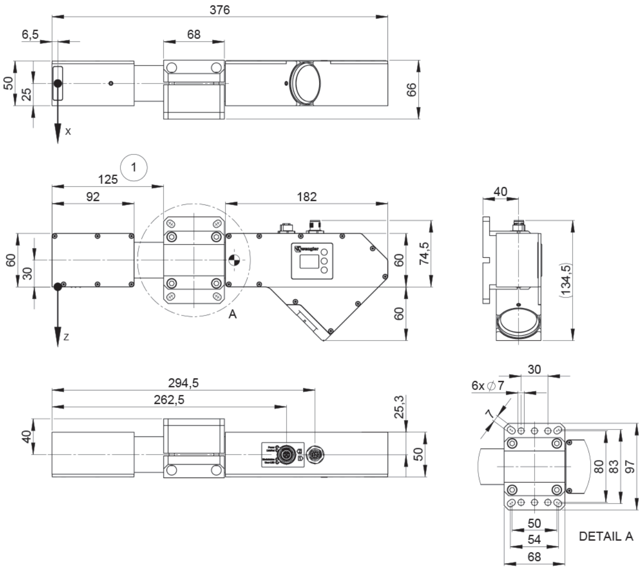
weCat3D MLWL1x2



1 = Recommended mounting position based on the sensor's center of gravity

Fig. 21: Dimensional Drawings MLWL1x2

weCat3D MLWL1x3



1 = Recommended mounting position based on the sensor's center of gravity

Fig. 22: Dimensional Drawings MLWL1x3

weCat3D MLWL1x4

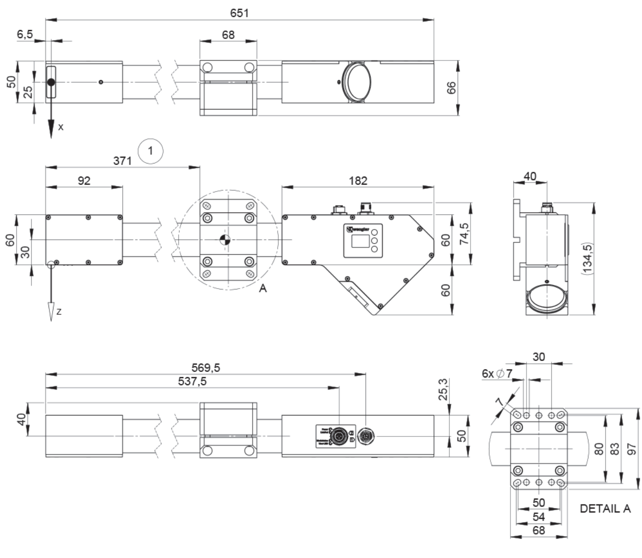


Fig. 23: Dimensional Drawings MLWL1x4

weCat3D MLWL1x5

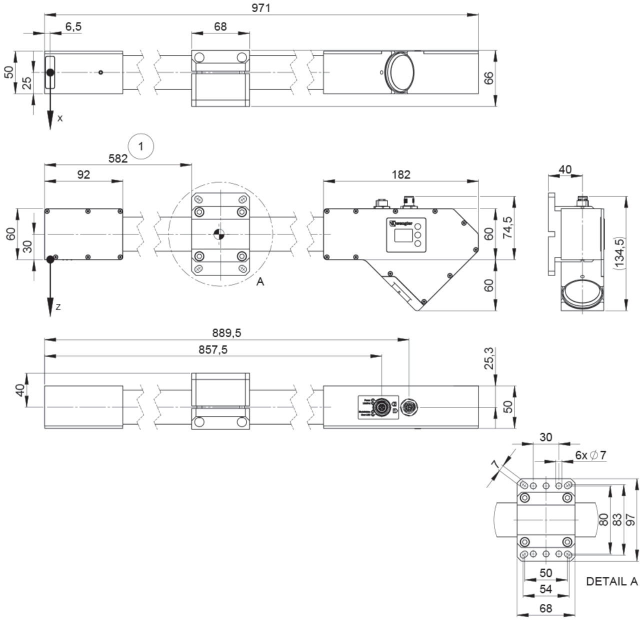


Fig. 24: Dimensional Drawings MLWL1x5

weCat3D MLWL2x1

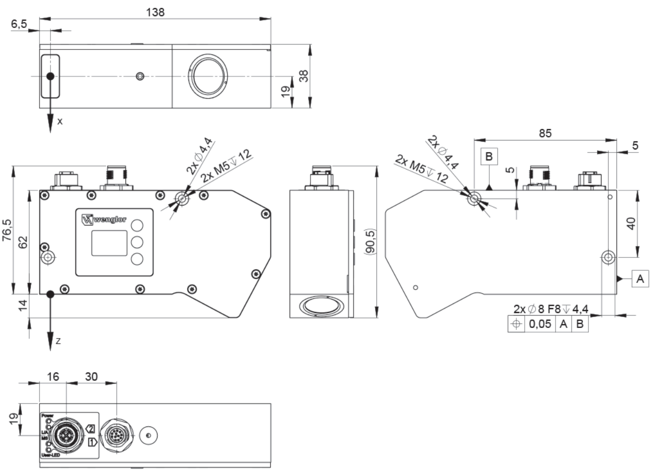


Fig. 25: Dimensional Drawings MLWL2x1

weCat3D MLWL2x2

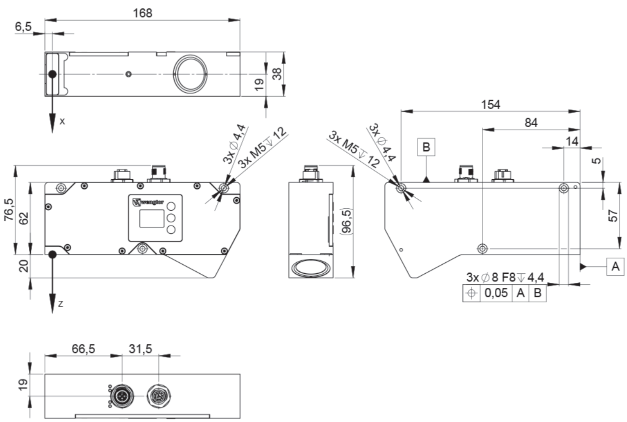


Fig. 26: Dimensional Drawings MLWL2x2

1 = Recommended mounting position based on the sensor's center of gravity

weCat3D MLWL2x3

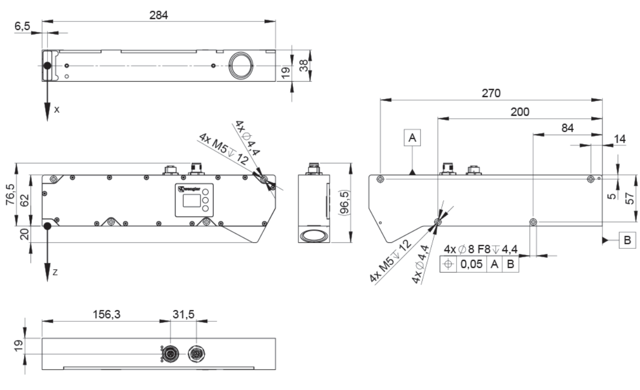


Fig. 27: Dimensional Drawings MLWL2x3

1 = Recommended mounting position based on the sensor's center of gravity

weCat3D MLWL2x4

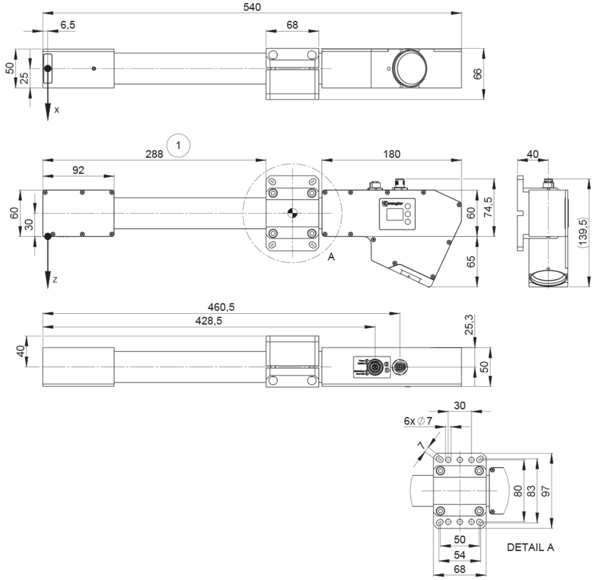


Fig. 28: Dimensional Drawings MLWL2x4

weCat3D MLWL2x5

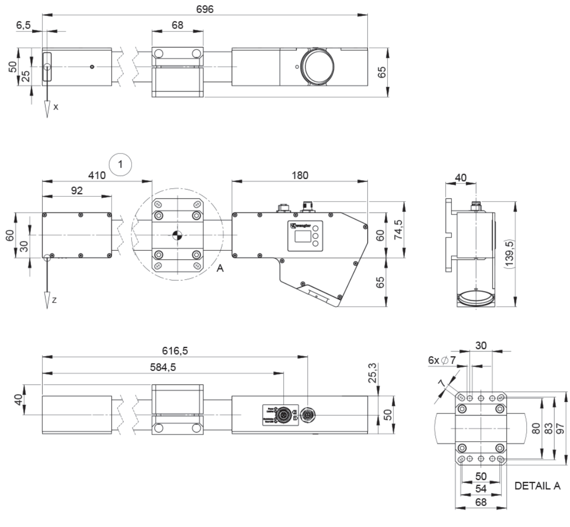
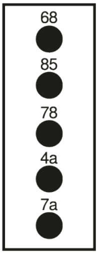


Fig. 29: Dimensional Drawings MLWL2x5

4.3 LED Display

A26



68 = Power
85 = Link/Act
78 = Modul status
4a = User-LED
7a = Laser (only MLSL2 with laser class 3R and 3B))

LED	Designation	Status	Function
68	Power	Blue	Operating voltage on
		Off	Operating voltage off
85	Link/Act	Green	Link included (1000 Mbit), no transmission
		Green blinking	Communication (1000 Mbit)
		Red	Link included (100 Mbit), no transmission
		Red blinking	Communication (100 Mbit)
		Orange	Link included (10 Mbit)
		Orange blinking	Communication (10 Mbit)
		Off	No Ethernet device connected
78	MS (Module Status)	Green	Device operative
		Red	Device error
		Off	Device doesn’t start up
4a	User LED	Green	The user can activate this LED individually.
		Red	
		Orange	
		Off	
7a	Laser	Green	Laser approval available
		Red	No laser approval
		Off	No supply voltage for laser shutdown

NOTE!
10 Mbit connection (orange LED lights up / blinks) is inadequate for error-free functioning (see also "Network Buffer" in section 7.2).

4.4 Control Panel

1. Up button
2. Enter button
3. Down button
4. Display

NOTE!
Display brightness may decrease with age. This does not result in any impairment of the sensor function.

5. Transport and Storage

5.1 Transport

Upon receipt of shipment, inspect the goods for damage in transit. In the case of damage, conditionally accept the package and notify the manufacturer of the damage. Then return the device making reference to damage in transit.

5.2 Scope of Delivery

- weCat3D Sensor
- Laser warnings
- Mounting set (included only in some devices)

5.3 Storage

The following points must be taken into condition with regard to storage:

- Do not store the product outdoors.
- Store the product in a dry, dust-free place.
- Protect the product against mechanical impacts.
- Protect the product against exposure to direct sunlight.
- Observe storage temperature.

ATTENTION:
Risk of property damage in case of improper storage!
The product may be damaged.
- Comply with storage instructions.

6. Installation and Initial Start-up

ATTENTION!
Risk of property damage in case of improper installation!
The product may be damaged.
- Comply with installation instructions.

6.1 General Installation Instructions

- Observe electrical and mechanical regulations, standards, and safety rules.
- Make sure that the sensor is mounted firmly and securely.
- The power supply should be connected directly and it should be as short as possible (max. length: 30 m).
- The sensor must be protected against mechanical influences.
- The sensor should not be subjected any vibration because this could influence measurement.
- The sensor must be installed such that the laser line is as perpendicular as possible to the surface to be measured in order to obtain accurate measurement results.
- Adequate heat dissipation must be assured for the device. This can be accomplished, for example, by means of a metallic connection between the sensor housing and the mounting base.
- A cooling unit should be used as of an ambient temperature of 45° C or in the event that the sensor is mounted in a thermally insulated manner (see section 6.4.3).
- Stable operation is achieved after a warm-up phase of 15 minutes.

6.1.1 Coordinate System of the Sensor

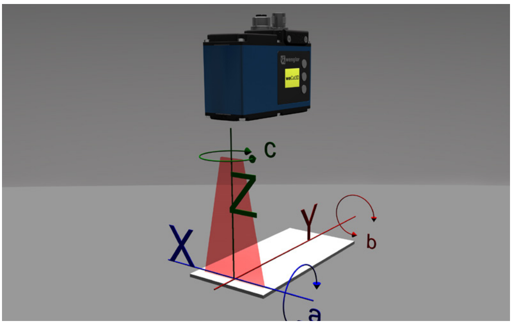


Fig. 30: Coordinate system of the sensor

The X-axis corresponds to measuring ranges X.
The a-axis corresponds to rotation around the X-axis.

The Z-axis corresponds to measuring range Z.
The c-axis corresponds to rotation around the Z-axis.

The Y-axis corresponds to advancing in the Y direction.
The b-axis corresponds to rotation around the Y-axis

6.1.2 Shadowing and Obstruction

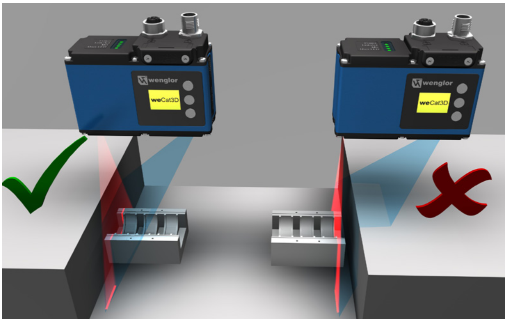


Fig. 31: Obstruction of the range of vision

The sensor's range of vision is obstructed by the object to be measured in the example shown at the right.

Measurement is possible without any obstruction in the example shown at the left.

6.1.3 Tilting around the a-Axis

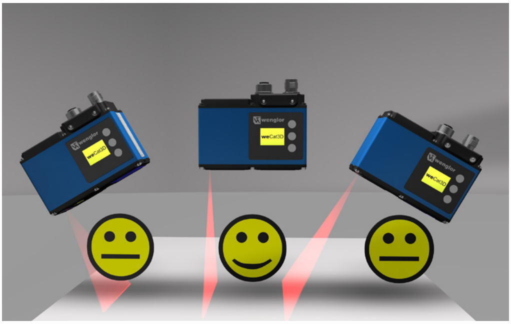


Fig. 32: Tilting around the a-axis

Tilting should be avoided in order to obtain ideal profile quality.

Nevertheless, thanks to their large dynamic range, weCat3D Sensors continue to provide measured values even in the event of tilting.

6.1.4 Tilting around the b-Axis

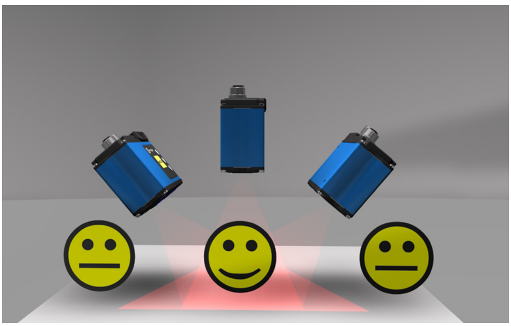


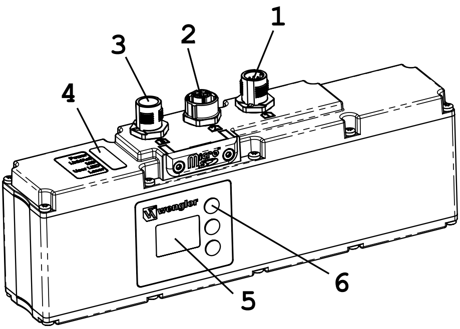
Fig. 33: Tilting around the b-axis

Tilting should be avoided in order to obtain uniform signal distribution for best possible profile quality.

Nevertheless, thanks to their large dynamic range, weCat3D Sensors continue to provide measured values even in the event of tilting.

6.2 Sensor Construction

weCat3D MLSL (example MLSL2xx)



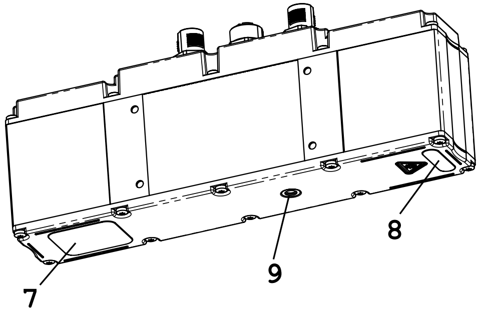
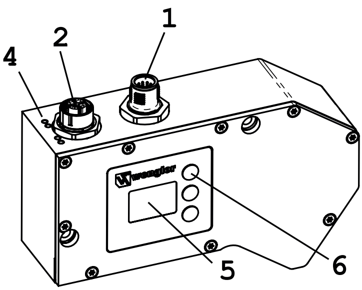


Fig. 34: Sensor construction of weCat3D MLSL2xx

weCat3D MLWL (example MLWL1x2)



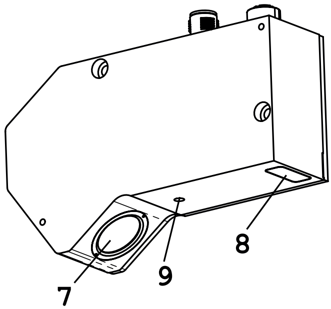


Fig. 35: Sensor construction of weCat3d MLWL1x2

① = Power supply plug connector, digital I/O
② = Connection Socket Ethernet
③ = Laser switching and safe laser shutdown (EN ISO 13849-2)
④ = LED display
⑤ = Control panel display
⑥ = Operating keys
⑦ = Receiver
⑧ = Laser exit
⑨ = Thread for fixing the screening grid retainer

ATTENTION:
The thread for fixing the screening grid retainer may not be used for fixing the sensor.

6.2.1 Tightening Torques

Tightening torques must be complied with in order to assure error-free operation. The respective values are listed in the following table.

Connection Type	Tightening torque in (Nm)
Connector cable: M12 (plug 1)	0,6
Network cable: M12 (socket 2)	0,4
Mounting: M5 (threaded)	2.5 (min. thread engagement length: 6 mm)
Mounting: M4 (threaded)	1,5 (min. thread engagement length: 4 mm)
M6 mounting (thread)	8

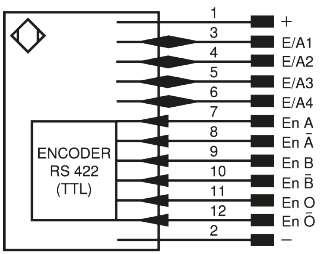
6.3 Installation

6.3.1 Electrical Connection

The number of connection plugs depends on the sensor type (see section 6.2). The sensor is supplied with 24 V operating voltage via the 12-pin connector (1). The 8-pin socket (2) is connected to the PC or switch and is used for the communication of the process and parametrization data. The external 24 V laser switching is carried out via the 8-pin connector (3), see section 6.3.2.

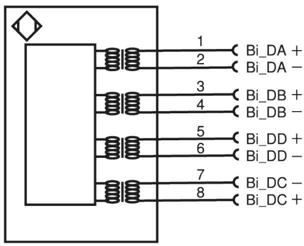
Connection diagram, power supply:

1034



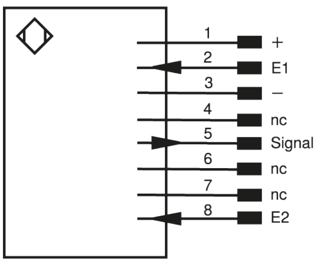
Connection diagram, Ethernet:

1022



Connection diagram for external 24 V laser shutdown:

1025



Legend

Legend					
+	Supply Voltage +	nc	Not connected	ENB RS422	Encoder B/B̄ (TTL)
–	Supply Voltage 0 V	U	Test Input	ENA	Encoder A
~	Supply Voltage (AC Voltage)	Ū	Test Input inverted	ENB	Encoder B
A	Switching Output (NO)	W	Trigger Input	AMIN	Digital output MIN
Ā	Switching Output (NC)	W–	Ground for the Trigger Input	AMAX	Digital output MAX
V	Contamination/Error Output (NO)	O	Analog Output	AOK	Digital output OK
V̄	Contamination/Error Output (NC)	O–	Ground for the Analog Output	SY In	Synchronization In
E	Input (analog or digital)	BZ	Block Discharge	SY OUT	Synchronization OUT
T	Teach Input	AMV	Valve Output	OLT	Brightness output
Z	Time Delay (activation)	a	Valve Control Output +	M	Maintenance
S	Shielding	b	Valve Control Output 0 V	rsv	Reserved
RxD	Interface Receive Path	SY	Synchronization	Wire Colors according to DIN IEC 60757	
TxD	Interface Send Path	SY–	Ground for the Synchronization	BK	Black
RDY	Ready	E+	Receiver-Line	BN	Brown
GND	Ground	S+	Emitter-Line	RD	Red
CL	Clock	⏚	Grounding	OG	Orange
E/A	Output/Input programmable	SnR	Switching Distance Reduction	YE	Yellow
	IO-Link	Rx+/–	Ethernet Receive Path	GN	Green
PoE	ower over Ethernet	Tx+/–	Ethernet Send Path	BU	Blue
IN	Safety Input	BUS	Interfaces-Bus A(+)/B(–)	VT	Violet
OSSD	Safety Output	La	Emitted Light disengageable	GY	Grey
Signal	Signal Output	Mag	Magnet activation	WH	White
BI_D+/–	Ethernet Gigabit bidirect. data line (A-D)	RES	Input confirmation	PK	Pink
EN0 RS422	Encoder 0-pulse 0/0̄ (TTL)	EDM	Contactor Monitoring	GNYE	Green/Yellow
PT	Platinum measuring resistor	ENA RS422	Encoder A/Ā (TTL)		

CAUTION!
Make sure that the cables have been correctly and securely connected in order to assure error-free operation.

NOTE!
Maximum permissible length of the power supply cable is 30 m.
The power supply cable must be equipped with an additional, suitable shield.

NOTE!
The voltage drop across the cable must be taken into account during installation.

6.3.2 Connection External 24 V Laser Shutdown

In order to be able to switch the laser on and off, series MLSL2 devices with laser class 3R and 3B are additionally equipped with a dedicated laser shutdown (see section 6.3.1). Laser shutdown must be supplied with 24 V operating voltage. Inputs E1 and E2 enable the laser when voltage is applied, and the circuit acknowledges enabling via the signal output (Laser off = 24 V; Laser on = 0 V).

NOTE!
This device is only ready for operation if external laser shutdown has been correctly connected.

6.3.3 Safe Laser Shutdown (EN ISO 13849-2)

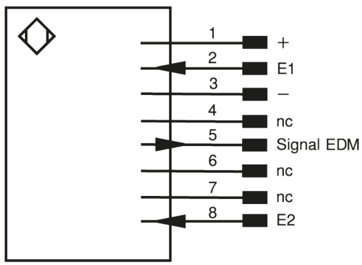
SGS TÜV SAAR
FUNKTIONALE SICHERHEIT GEPRÜFT
FUNCTIONAL SAFETY APPROVED

The following weCat3D Profile Sensors have a safe laser shutdown in accordance with EN ISO 13849-2:

Sensor
MLSL245S40
MLSL246S40
MLSL275S40
MLSL276S40

6.3.3.1 Pin Assignment

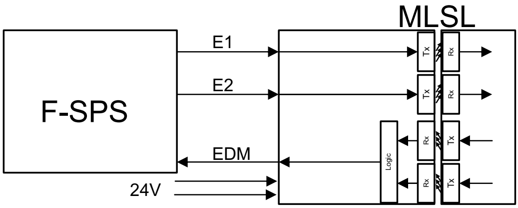
1037



The laser can be safely switched off via two external control signals (plug 3, inputs E1 and E2). The feedback signal (plug 3, EDM signal) sends a signal to the external control when the safe state (laser off) has been reached.
The feedback signal has the same switching behavior as a positively driven NC contact of a safety relais. This functionality is often called External Device Monitoring (EDM).

HINWEIS!
"Safe state" means that the laser emission is within the limits for laser class 1 in accordance with IEC 60825-1.

Block diagram with a fail-safe PLC (F-PLC) and a MLSL sensor:



Truth table:

Logic state input E1	Logic state input E2	Logic state signal EDM	Laser state
0	**0**	**1**	Laser OFF
0	1	0	Laser OFF
1	0	0	Laser OFF
1	**1**	**0**	Laser ON*

* The laser may be OFF if it is switched off on the device side.

6.3.3.2 Requirements for Integration into the Safety Circuit

In order to meet the requirements of EN ISO 13849-2, the integration of the sensor must fulfill the following criteria. If they are not met, the laser cannot be shut down safely in accordance with EN ISO 13849-2:

- The laser shutdown must be activated via two channels (inputs E1, E2).
  - » The external control signals must be generated by a system-side control whose safety level corresponds to the requirements of the risk analysis.
- The EDM feedback signal must be evaluated in the control in accordance with the truth table. If the feedback signal does not meet expectations, the control must switch to a safe state.
  - » Monitoring and the plausibility check must be performed by a system-side control whose safety level corresponds to the requirements of the risk analysis.
- The system-side control must ensure that access to the laser emission area is granted only if
  - » the laser is OFF
  - » the EDM feedback signal has been successfully checked for plausibility.
- A restart inhibit must be implemented in the system-side control.
- The system-side control and the safe laser shutdown may be connected to the same power supply.
- The connection cable between the safe laser shutdown and the system-side control must be protected against cross-circuits with measures in accordance with EN ISO 13849-2, Table D.4.

6.3.3.3 Technical Data

Electrical Data	
Power supply to plug 3	18...30 V SELV (Safety Extra Low Voltage) PELV (Protective Extra Low Voltage)
Fuse protection for the voltage supply and inputs E1, E2	Max. 4 A
Current consumption (Ub = 24 V)	≤ 100 mA
Voltage range for inputs E1, E2	-30...+30 V (SELV / PELV)
Switching thresholds for inputs E1, E2	Low: < 5 V / < 15 mA High: > 15 V / 2...30 mA
Switching current of the EDM feedback signal	≤ 45 mA
Voltage drop of the EDM feedback signal	≤ 2,3 V
Technical Safety Data	
Safety function response time	Laser ON → Laser OFF: max. 600 ms Laser OFF → Laser ON: max. 600 ms
Feedback signal response time	Laser ON → Laser OFF: max. 650 ms Laser OFF → Laser ON: max. 650 ms
Category (EN ISO 13849-1:2016)	Kat. 4
Max. achievable performance level (EN ISO 13849-1:2016)	PL e
Diagnostic Coverage (DC: EN ISO 13849-1:2016)*	Max. 99 %
MTTFd (EN ISO 13849-1:2016)	> 100 a
Mission time (TM; EN ISO 13849-1:2016)	20 a
Common Cause Factor (CCF; EN ISO 13849-1:2016)	≥ 65

* Diagnosis is performed by the system-side control

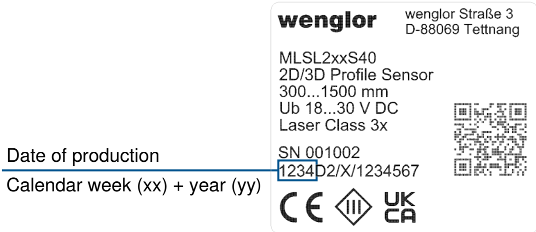
6.3.3.4 Complementary Products

ATTENTION!
The following cables must be used to follow the security level of certified laser interlock.

Connection lines

M12, 8-pin to open end	Description	Cable length
ZC9L001	Socket, straight, TPE-U (PUR)	5 m
ZC9L002	Socket, straight, TPE-U (PUR)	10 m
ZC9L003	Socket, angled, TPE-U (PUR)	5 m
ZC9L004	Socket, angled, TPE-U (PUR)	10 m

6.3.3.5 Type Label MLSL2xxS40



6.3.4 Adjusting the Sensor's Network Settings

Upon shipment from the factory, the sensor's **IP address is 192.168.100.1** and its **subnet mask is 255.255.255.0.**

In order to be able to connect the sensor to your PC, you have to make sure that the sensor and your PC are both within the same IP address range.

Address format for IP Addresses (IPv4)

	Network Part	**Device Part** (host part)
IP address	192.168.100.	**001**
Subnet mask	255.255.255.	**000**

The network part of the sensor's IP address must coincide with the network part of PC's IP address, but the device part of the address must be different for the sensor and the PC. By default, the IP address of the network adapter card is set to dynamic (automatic allocation). Change the setting to "static" (see Fig. 36 and Fig. 37).

The integrated web server can then be accessed where, amongst other things, the IP address can be changed. Further information can be found in section 7.

Alternatively, the IP address can be changed without connection to the PC directly at the control panel (OLED display). Further information can be found in section 8.

NOTE!
After change of the IP address the sensor will reboot.

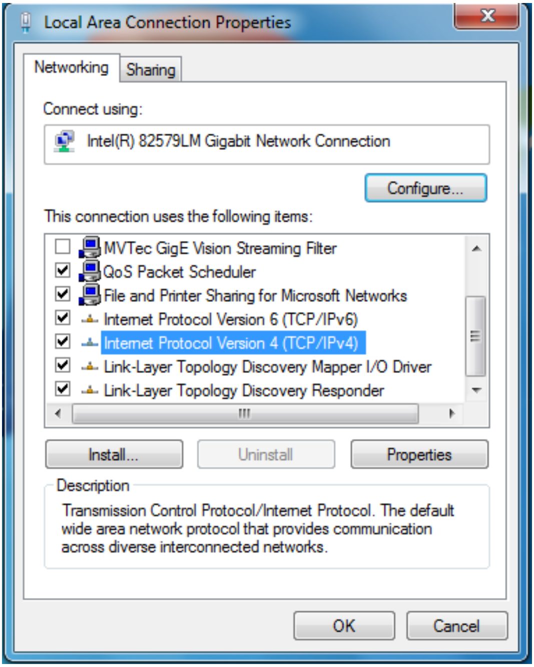


Fig. 36: Local connection properties

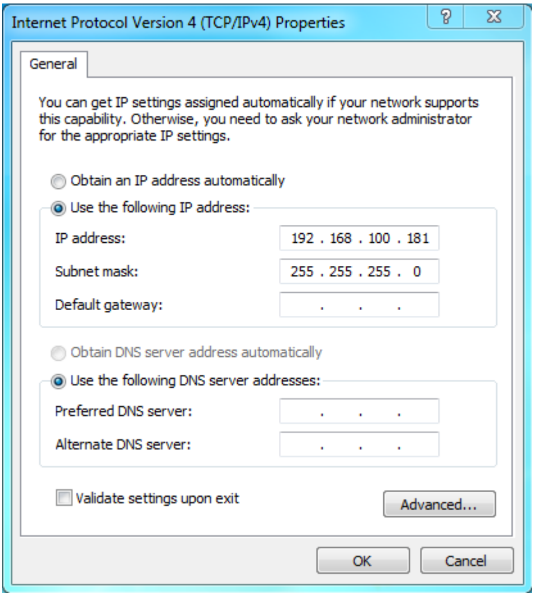


Fig. 37: Properties of internet protocol

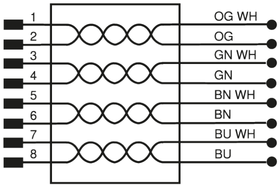
6.4 Complementary Products

wenglor offers connection technology for field wiring.

Suitable mounting technology no.	343

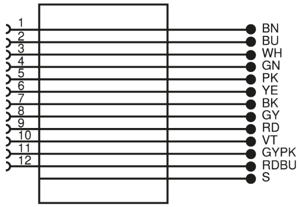
50

S80



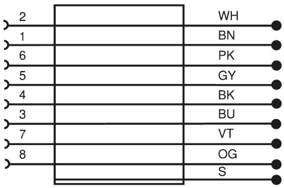
87

S89



89

S74



Switch EHSS001
Cooling unit
Screening grid retainer
Protective screen
Control Unit

A detailed list of available products can be found in the two following sections.

6.4.1 MLSLxxx

Software (optional)

DNNF012	uniVision for Linux
DNNF020	uniVision for Windows
DNNF013*	VisionApp Demo 3D
DNNP001*	VisionApp 360 for Windows
DNNP011*	Plugin VisionApp 360

Control Unit (optional)

BB1C001*	uniVision Profile
BB1C008*	uniVision Profile Extended
BB1C101*	uniVision Profile (Industrial Ethernet)
BB1C102*	uniVision Profile Extended (Industrial Ethernet)
BB1C105*	uniVision All in One (Industrial Ethernet)

Cooling Unit (optional)

ZLSK001

Screening Grid Retainer (optional)

ZLSS001	for MLSL1xx
ZLSS002	for MLSL2xx

Screening Grid Sets (optional)

ZLSE001 (Plastic)	for ZLSS001
ZLSE002 (Glass)	for ZLSS001
ZLSE005 (Plastic)	for ZLSS002
ZLSE006 (Glass)	for ZLSS002

Mounting System

ZLSZ001
ZLSZ002

microSD Card (optional spare part)

ZNNG013	included in delivery

Connection Lines (MLSL24x, MLSL25x, MLSL27x)

M12, 8-pin to open end		
ZAS89R201**	straight	2 m
ZAS89R501**	straight	5 m
ZAS89R601**	straight	10 m
ZAS89R701**	straight	20 m
ZAS89R202**	angled	2 m
ZAS89R502**	angled	5 m
ZAS89R602**	angled	10 m

Connection Lines

M12, 12-pin to open end		
ZDCL001**	straight	2 m
ZDCL002**	straight	5 m
ZDCL003**	straight	10 m
ZDCL007**	straight	30 m
ZDCL004**	angled	2 m
ZDCL005**	angled	5 m
ZDCL006**	angled	10 m

Connection Cables

M12, 8-pin to RJ45		
ZC1V001	straight	2 m
ZAV50R502	straight	5 m
ZC1V002	straight	10 m
ZC1V013	straight	30 m
ZC1V009	angled	2 m
ZC1V010	angled	5 m
ZC1V011	angled	10 m
ZC1V003**	straight	5 m
ZC1V014**	straight	10 m
ZC1V015**	angled	10 m

Connection Cables

M12, 12-pin to M12, 12-pin	
ZDCV001**	2 m
ZDCV002**	5 m
ZDCV003**	10 m

* Profile mode only (see section 7.2.2)

** Drag chain suitable

NOTE!

The cable outlet direction of angled plugs can be found in section 6.4.5).

6.4.2 MLWLxxx

Software (optional)

DNNF012	uniVision for Linux
DNNF020	uniVision for Windows
DNNF013*	VisionApp Demo 3D
DNNP001*	VisionApp 360 for Windows
DNNP011*	Plugin VisionApp 360

Control Unit (optional)

BB1C001*	uniVision Profile
BB1C008*	uniVision Profile Extended
BB1C101*	uniVision Profile (Industrial Ethernet)
BB1C102*	uniVision Profile Extended (Industrial Ethernet)
BB1C105*	uniVision All in One (Industrial Ethernet)

Cooling Unit (optional)

ZLWK001	for MLWL1x1
ZLWK002	for MLWL1x2
ZLWK003	for MLWL1x3 / MLWL1x4 / MLWL1x5 / MLWL2x4 / MLWL2x5
ZLWK004	for MLWL2x1
ZLWK005	for MLWL2x2
ZLWK006	for MLWL2x3

Screening Grid Retainer (optional)

ZLWS001	for MLWL1x1
ZLWS002	for MLWL1x2
ZLWS003	for MLWL1x3 / MLWL1x4 /MLWL1x5 / MLWL2x4 / MLWL2x5
ZLWS004	for MLWL2x1
ZLWS005	for MLWL2x2
ZLWS006	for MLWL2x3

Screening Grid Sets (optional)

ZLWE007 (Plastic)	for ZLWS001/ ZLWS002 / ZLWS004
ZLWE004 (Glass)	for ZLWS001/ ZLWS002 / ZLWS004
ZLWE008 (Plastic)	for ZLWS003
ZLWE005 (Glass)	for ZLWS003
ZLWE003 (Plastic)	for ZLWS005 / ZLWS006
ZLWE006 (Glass)	for ZLWS005 / ZLWS006

Mounting System

ZLSZ001
ZLSZ002

Connection Lines

M12, 12-pin to open end		
ZDCL001**	straight	2 m
ZDCL002**	straight	5 m
ZDCL003**	straight	10 m
ZDCL007**	straight	30 m
ZDCL004**	angled	2 m
ZDCL005**	angled	5 m
ZDCL006**	angled	10 m

Connection Cables

M12, 8-pin to RJ45		
ZC1V001	straight	2 m
ZAV50R502	straight	5 m
ZC1V002	straight	10 m
ZC1V013	straight	30 m
ZC1V009	angled	2 m
ZC1V010	angled	5 m
ZC1V011	angled	10 m
ZC1V003**	straight	5 m
ZC1V014**	straight	10 m
ZC1V015**	angled	10 m

Connection Cables

M12, 12-pin to M12, 12-pin	
ZDCV001**	2 m
ZDCV002**	5 m
ZDCV003**	10 m

* Profile mode only (see section 7.2.2)

** Drag chain suitable

NOTE!

The cable outlet direction of angled plugs can be found in section 6.4.5).

6.4.3 Assembly of the Cooling Unit

For each sensor type there is a suitable cooling unit optionally available (see sections 6.4.1 and 6.4.2).

weCat3D MLSL

The cooling unit for MLSL1xx and MLSL2xx ist equipped with two captive screws (2.5 mm internal hex).
A heat conducting foil (ZNNE004) has to be placed between sensor housing and cooling element (see Fig. 38).

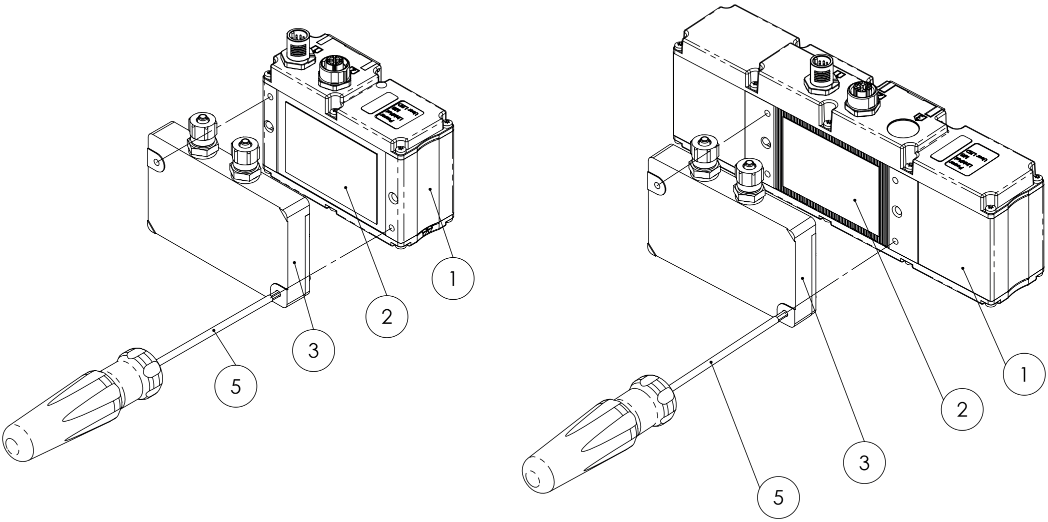


Fig. 38: Assembly of the cooling unit of weCat3D MLSL1xx (left) and MLSL2xx (right)

① = weCat3D Profile Sensor
② = Heat conduction foil
③ = Cooling unit
④ ---
⑤ = Hexagon screwdriver

NOTE!
When dismantling the cooling unit, first loosen the captive screws. If the cooling unit still cannot be removed, slide a sheet of metal or something similar between the sensor housing and the cooling unit and carefully remove the self-adhesive heat conducting foil.
Any adhesive residues can be removed with isopropanol.

weCat3D MLWL

The cooling units of the MLWLxxx sensors consist of one or two elements, depending on the design. A heat conduction foil is not required. The cooling unit is mounted with the supplied fastening screws (see Fig. 39).

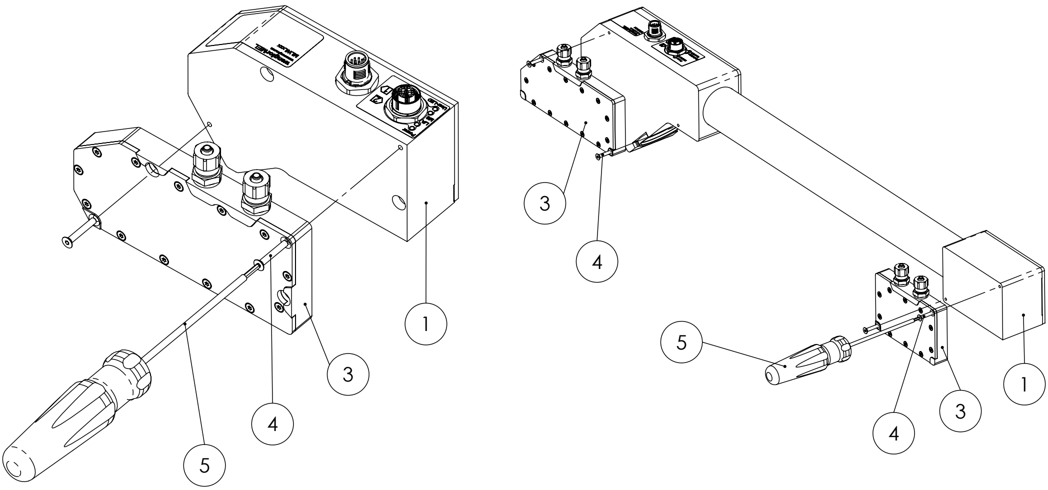


Fig. 39: Assembly of the cooling unit of weCat3D MLWLxxx

① = weCat3D profile sensor
② ---
③ = Cooling unit
④ = Countersunk screw M3 x 20 mm
⑤ = Hexagon screwdriver

6.4.4 Assembly of Screening Grid Retainer

For each sensor type there is a suitable screening grid retainer including screening grids optionally available (see sections 6.4.1 and 6.4.2).

For the sensors MLSL2xx and MLWLxxx, the assembly is carried out with the captive screw fixed at the screening grid retainer.

The screening grid retainer of MLSL1xx is equipped with two clamps which must be snapped into the housing.

ATTENTION:
Before mounting the screening grid retainer at the sensor the M5 grub screw has to be removed from the mounting hole with a hexagon screwdriver.

6.4.4.1 Inserting the Screening Grids

Each set of screening grids consists of two separate screens. Depending on sensor type there are two different methods of mounting them into the screening grid retainer (see Fig. 40 ... Fig. 42)

Method 1 (screwed screening grid retainer)**:**

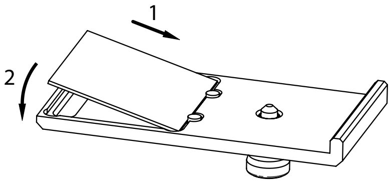


Fig. 40: Inserting the screening grid with screwed screening grid retainer (Method 1)

The screening grid is pressed diagonally against the two rubber clamps (1) and then put into the gap of the screening grid retainer (2).

Method 1 (clamped screening grid retainer)**:**

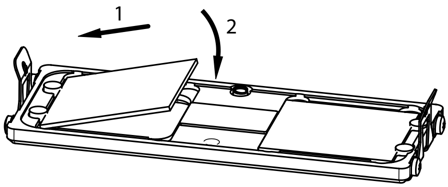


Fig. 41: Inserting the screening grid with clamped screening grid retainer (Method 1)

The assembly follows the same principle as with the screwed screening grid retainer (see above).

Method 2 (only for screwed screening grid retainers)**:**

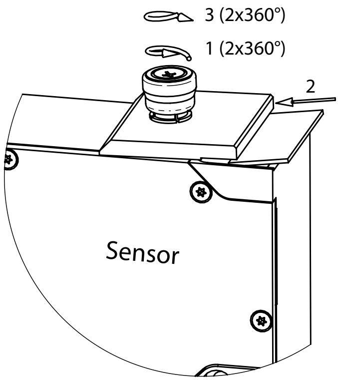
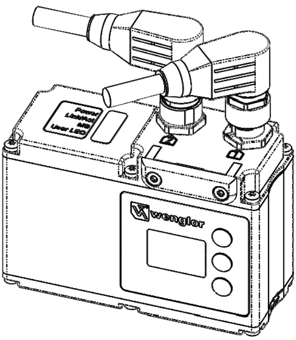


Fig. 42: Inserting the screening grid (Method 2)

First, the screening grid retainer is fixed to the sensor housing with one turn (1). Then the screening grid is pushed sideways into the gap (2) and the screening grid retainer is screwed tight with two more turns (3).

6.4.5 Direction of the cable outlet with angled plugs

The direction of the outgoing cable with angled plugs varies depending on the sensor type (see Fig. 43 and Fig. 44).



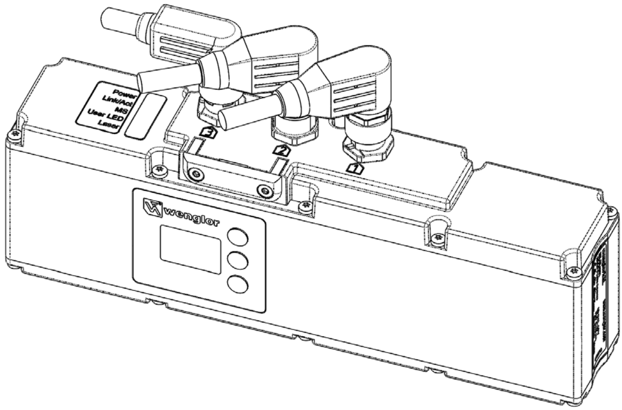
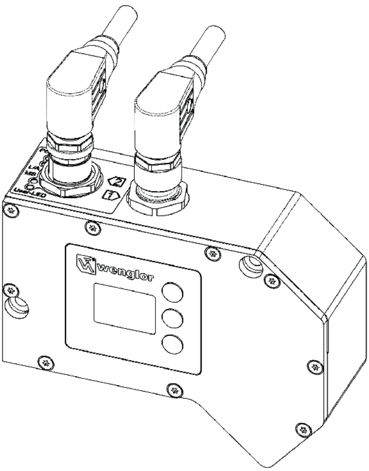


Fig. 43: Cable outlet weCat3D MLSL1xx (left) and weCat3D MLSL2xx (right)



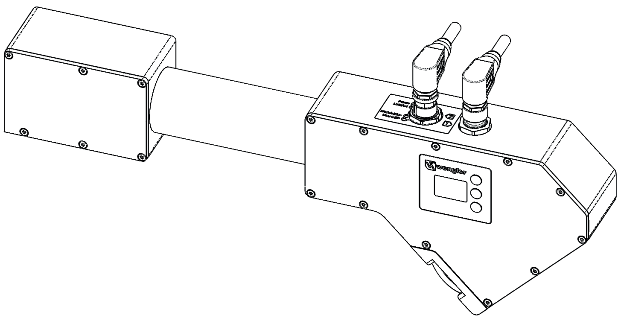


Fig. 44: Cable outlet weCat3D MLWLxxx compact (left) and weCat3D MLWL with carbon rod (right)

NOTE!
You will find an overview of the availble angled cables in the sections 6.4.1 and 6.4.2.

6.5 Default Settings

			weCat3D MLSL	weCat3D MLWL
Pin function	E/A1		Encoder E1+E2	Encoder E1+E2
	E/A2		Encoder E1+E2	Encoder E1+E2
	E/A3		Sync Out	Sync Out
	E/A4		Sync In	Sync In
I/O settings	E/A1		Operating voltage active	Operating voltage active
	E/A2		Operating voltage active	Operating voltage active
	E/A3		Push-Pull	Push-Pull
	E/A4		Operating voltage active	Operating voltage active
Encoder	Direction of rotation		Independent of direction	Independent of direction
	Encoder divider		0	0
Display	Intensity		Screensaver	Screensaver
	Mode		Analysis	Analysis
Operating mode			Profile Sensor	Profile Sensor
Web password			Disabled	Disabled
Profile	Measuring rate (Hz)		200	175
	Signal selection		Intensity	Intensity
	Exposure time (μs)*		150	150
	Measuring field (pix)	Offset X	0	0
		Width X	1280	2048
		Offset Z	0	0
		Height Z	1024	2048
Interface	IP address		192.168.100.1	192.168.100.1
	Subnet mask		255.255.255.0	255.255.255.0
	TCP port		32001	32001
	Std. gateway		192.168.100.254	192.168.100.254
	MAC address		See OLED display, interface section 8.7.2	See OLED display, interface section 8.7.2
	UDP address		0.0.0.0	0.0.0.0
	UDP port		32001	32001
Language			English	English
Password OLED	Deactivate/ activate		Deactivated	Deactivated
	Change		„0000“	„0000“
Network password			admin	admin

* A standard exposure time is set upon shipment from the factory. With some surfaces, exposure time has to be matched to ambient conditions, i.e. increased or reduced.

7. Integrated Web Server

The integrated web server makes it possible to enter settings for the sensor and save them directly at the PC (see Fig. 45).

NOTE!
The website can be opened with any standard browser (e. g. Edge, Chrome, Firefox).

7.1 Accessing the Integrated Website

Start your web browser and enter the preset IP address (192.168.100.1) to the browser's address line.

NOTE!
If the actual IP address differs from the preset address and you don't know the actual address, you can view it at the OLED display after selecting the "Interface" menu item (see section 8.7).

7.2 Structure of Device Website

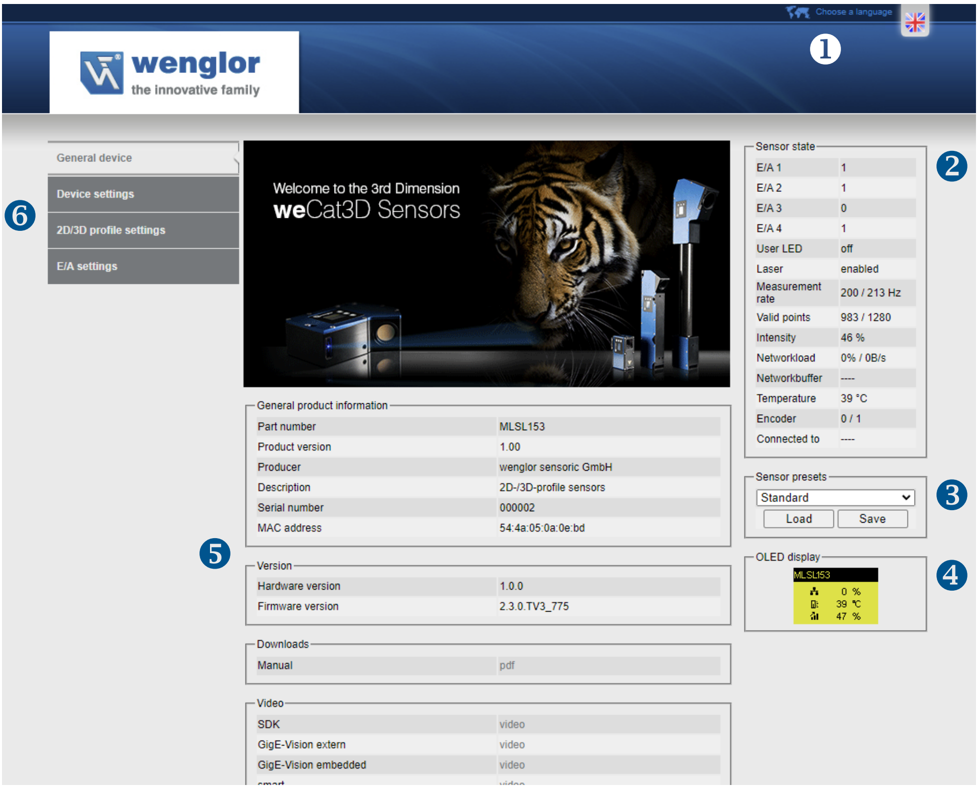


Fig. 45: Integrated webserver of weCat3D Profile Sensor

The integrated website is subdivided into the following areas:

① **Language Selection**
The website can be changed from English to German with the language selection function.

② **Sensor Status**

I/O1…I/O4	Indicates the current switching status of the respective input or output. (1/0)
User LED	Indicates the color in which the user LED currently lights up (off, green, red, orange).
Laser	Indicates the current status of the laser (enabled = laser on / disabled = laser off).
Measuring rate	Left: current measuring rate Right: maximum possible measuring rate with selected measuring range and exposure time (calculated value may deviate slightly)
Valid points	Left: number of valid measuring points within the measuring range. Right: maximum number of measuring points within the selected measuring range.
Intensity	Indicates the intensity reffered to the average of all valid points within the measuring range. In typical applications, an intensity within a range of 10 to 90 % results in an ideal profile. The intensity is influenced by sensor installation and the exposure time setting.
Network load	Indicates the network's current transmission load (at the sensor side). Continuous utilization of close to 100 % should be avoided because overrun might otherwise occur at the sensor's network buffer. Utilization can be influenced by reducing the measuring rate or by changing the content of the transmission protocol (see section 7.2.3).
Network buffer	Indicates internal network buffer occupancy as a percentage. Momentary increases in occupancy are no problem. However, if occupancy increases continuously, network utilization must be reduced (see description of "network utilization"), in order to prevent the loss of profile data. **NOTE!** Bandwidth may be low. Check the network settings at the PC and/or the LED display. If the orange LED lights up, transmission speed is only 10 Mbit (see section 4.3).
Temperature	Displays current temperature inside the sensor housing. Depending on how the sensor is mounted, this temperature is 15 to 25° C above ambient temperature. In order to avoid damage and a reduced service life, use the sensor within the specified temperature range only. **NOTE!** As of an internal temperature of 60° C, the sensor is in its critical temperature range.
Encoder	Left: HTL encoder counter (rotary encoder) Right: RS422 TTL encoder counter (rotary encoder)
Connected with	Displays the IP address of the PC or the control unit with which the sensor is connected.

③ **Sensor presets**
This field provides the user with the opportunity of saving all of the selected settings to the sensor and retrieving them later. The values saved under "Standard" are loaded automatically when the sensor is started up (see section 8.6).

④ **OLED Display**
This field shows the current content of the OLED display. It’s refreshed approximately once per second.

⑤ **Dynamic Page Content**
On access to the web page following information are provided:

- **General product information**
  Device-specific information
- **Version**
  Firmware and hardware version of the sensor
- **Downloads**
  Download of the weCat3D manual
- **Video**
  Download of Getting Started Videos for SDK, GigE Vision (external / embedded) and smart sensor.

⑥ **Category Selection**
The integrated website offers four different categories:

- **General device**
  Overview page with general information regarding the sensor.
- **Device settings**
  Network and display settings can be changed, and reset commands as well as a sensor restart can be triggered.
- **2D/3D profile settings**
  Profile display with option for parameter settings.
- **E/A settings**
  Function and performance of the four configurable I/Os can be selected.
- **Visualization**
  Is shown if the operation mode is smart weCat3D (see "General Settings"/"Operating mode).

7.2.1 General Device

This is the sensor’s initial page which displays all relevant information concerning the device such as order number, product version, manufacturer, description, serial number and MAC address, as well as hardware and firmware version

7.2.2 Device Settings

Content is subdivided into 4 categories (see Fig. 46 ... Fig. 49):

Network settings

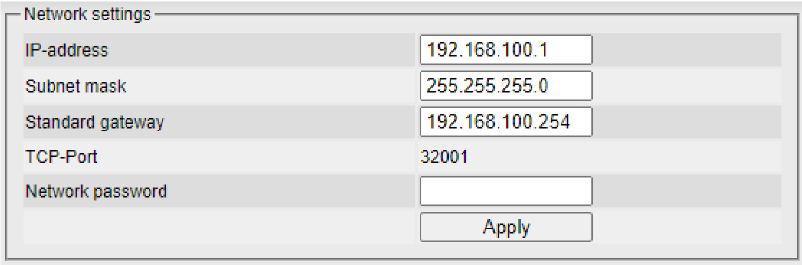


Fig. 46: Network settings

The desired address ranges can be entered to the **“IP Address”**, **“Subnet Mask** and **“Standard Gateway”** fields. These addresses permit operation, as well as communication between the sensor and your network (PC).

CAUTION!

- If you don’t have access to information concerning available address ranges within your network, contact your IT department first.
- Incorrect entries may result in network conflicts.
- The sensor’s IP address must differ from the IP address of the PC.

After the desired changes have been made, enter the Network password (**admin**) to the field and click **“OK”**. The changes are activated without restarting the sensor. In order to return to the integrated website, enter the new IP address to your web browser’s address line.

NOTE!
If no password or an incorrect password is entered, a corresponding note will be displayed.

NOTE!
After change of the IP address the sensor will reboot.

UDP Socket settings

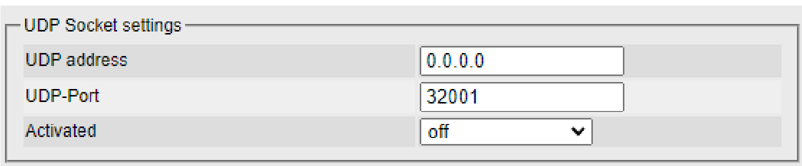
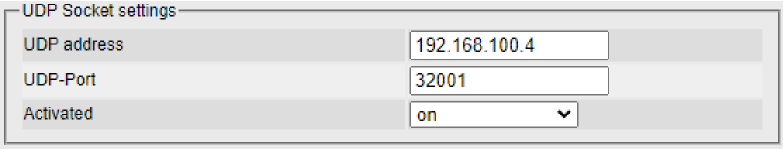


Fig. 47: UDP Socket settings

UDP-address	Host IP address, to which the sensor sends data
UDP-Port	Host port number to which the sensor sends data
Activated	Activating/deactivating UDP socket connection

Example:



NOTE!
For data transmission via UDP, it must be ensured that the receive port on the host PC is not occupied.

NOTE!
Changes of UDP data via web interface are only applied after the sensor has been restarted.

Display settings

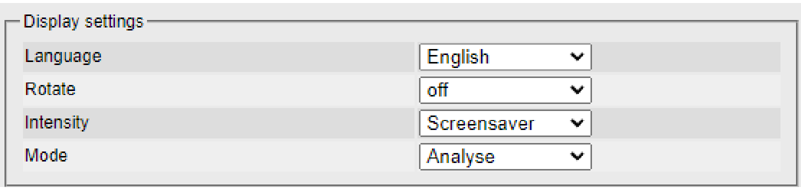


Fig. 48: Display settings

Language	Sets the language for the display (German, English).
Rotate	The display is rotated 180° (on/off).
Intensity	Adjusts display performance. • Normal: Display intensity is set to the default value. • Energy saving: If no keys are pressed for a period of one minute the display is switched off, and is switched back on as soon as a key is activated. • Screensaver: If no keys are pressed for a period of 30 seconds, the display is switched to the run mode and returns to the last used menu as soon as a key is activated. The colors are inverted every 30 seconds in order to protect the display.
Mode	Selection of various display modes for the run mode • Network: IP address, subnet mask and MAC address are displayed. • I/O status: Display of input and output states • Analysis: Displays network utilization as a percentage, internal temperature in °C and intensity as a percentage. • Live Image: Displays the current profile image

General Settings

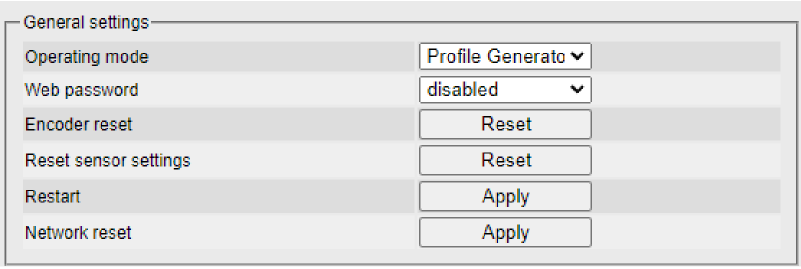


Fig. 49: General settings

Operating mode	• Profile Generator: The sensor functions as a 2D Profile Sensor and sends the measured profile to the PC or the control unit. • GigE Vision: If selected an embedded GigE Vision server is available and the sensor can be connected directly with image processing software supporting GigE Vision (see section 12.1). • Smart weCat3d: The sensor works in conjunction with the uniVision software (see Operating Instructions DNNF012/DNNF020) and enables thus an user-specific evaluation in the sensor. **NOTE!** In GigE Vision and smart weCat3D mode using VisionApp Demo 3D, VisionApp 360 or SDK is not possible.
Web password	Activating/deactivating the web password "admin". The setting is retained after a restart. **NOTE!** It can happen that a password prompt appears without being activated. In order to eliminate this it is necessary to clear the cache of the browser. This can be done using the key combination Ctrl+F5.
Encoder reset	Resets both encoder counters (rotary encoders) in the sensor to zero.
Reset sensor settings	Returns all settings to their default values. Exception: Network settings. The settings in Set0, Set1 and Set2 are deleted.
Restart	Restarting of the sensor can be forced by pressing “Restart”.
Network reset	Returns the network settings to their default values (see section 6.5). **NOTE!** After a network reset the sensor will be restarted.

7.2.3 2D/3D Profile Settings

This page contains the display of the measured profile and several setting options (see Fig. 50).
If the sensor is in Camera mode camera image of the sensor is shown. Using the camera image it is easier to identify sources of artifacts as reflections.

NOTE!
It is not recommended to use the profile display when the sensor is in the "Smart weCat3D" mode (see section 7.2.5). The speed in the "Smart weCat3D" mode could be reduced.

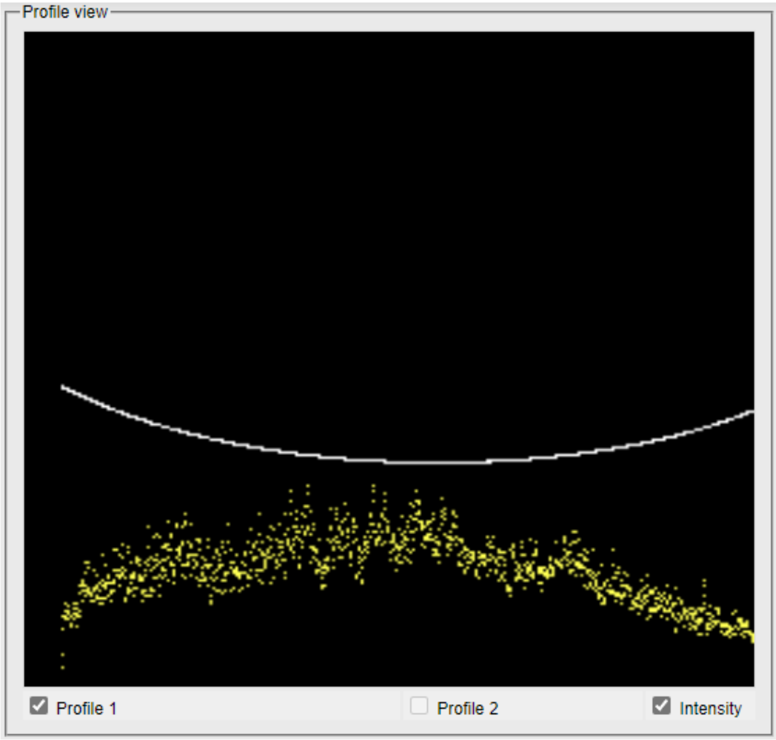


Fig. 50: Profile view and settings

Profile 1/2	The sensor enables the output of two profiles, which can be displayed or hidden by selecting or unselecting the checkboxes (this affects the display only and doesn't influence the parameter settings). The individual colors have the following meanings: White: Profile 1 Red: Profile 2 (only visible with corresponding setting) Yellow: Intensity, profile 1 Orange: Intensity, profile 2 (only visible with corresponding setting)
Intensity	Provides information about how much light is received at each point along the laser line.

ROI Settings

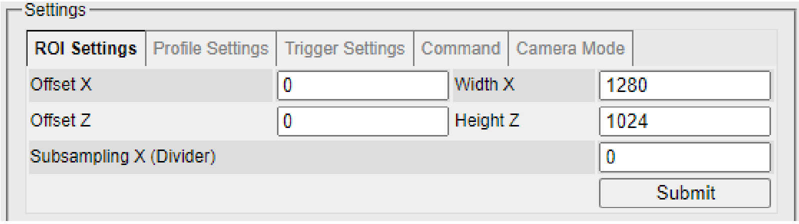


Fig. 51: ROI settings

The region of interest (ROI) or the sensor's active range consists of the four following variables: Offset X, Width X, Offset Z and Height Z (see Fig. 51). These variables are specified in pixels and can be changed as desired within their respective limits. This information makes it possible to reduce the active range so that only the actually required range is read out.
The selected ROI should be as large as necessary and as small as possible. The smaller the range, the faster the evaluation and thus the higher the measuring rate. The measuring rate of the individual sensor can be increased in this way.
Differentiation must be made in this respect between the weCat3D MLSL, for which a reduction in X and Z influences the measuring rate, and the weCat3D MLWL, for which only a reduction in Z affects the measuring rate. Limiting X only reduces network load in this case.

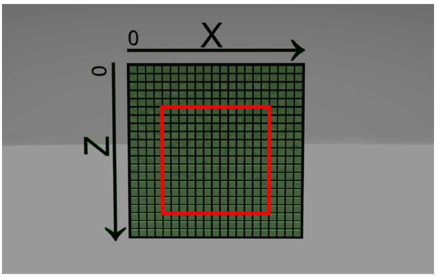


Fig. 52: Image of the integrated camera

Starting point "0" can be seen at the top left.
X designates the measuring range X of the columns.
Z designates the measuring range in rows.
The red rectangle shows the selected ROI.

Example in Fig. 52: Offset X = 4 Width X = 13
Offset Z = 5 Height Z = 13

Offset X	The integrated camera in the device has 1280 columns (MLSL) or 2048 columns (MLWL) in the X direction, which are read out continuously. As a default setting, offset X has a value of 0. If this value is increased, not all of the columns are read out, but rather only those as of the new starting point (see section 10.5.40.2). **NOTE!** In this case, width X must be manually adjusted because it's not self-adjusting.
Width X	Specifies the total number of columns which will be read out in the X direction (see "Offset X"). As a default setting, all 1280 or 2048 columns are read out (see section 10.5.40.1). **NOTE!** This setting reduces or increases measuring range X.
Offset Z	The integrated camera in the device has 1024 columns (MLSL) or 2048 rows (MLWL) in the Z direction, which are read out continuously. As a default setting, offset Z has a value of 0. If this value is increased, not all of the rows are read out, but rather only those as of the new starting point (see section 10.5.40.5). **NOTE!** In this case, height Z must be manually adjusted because it's not self-adjusting.
Height Z	Specifies the total number of rows which will be read out in the Z direction. As a default setting, all 1024 or 2048 rows are read out (see section 10.5.40.4). **NOTE!** This setting reduces or increases measuring range Z.
Subsampling X	Sets the number of measured values in X, which will be read out. This setting reduces resolution in X but has no influence on the maximum profile rate – only network load is reduced (see section 10.5.40.3).

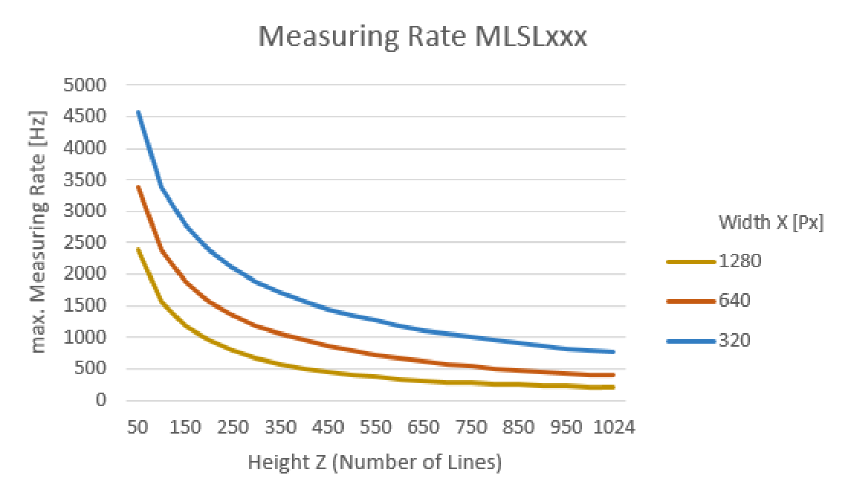


Fig. 53: MLSL: Measuring rate depends on height Z (number of lines) and on the number of columns (width X)

NOTE!
Reducing the width X below 320 pixels has no effect on the measuring rate.

NOTE!
For how to reduce the number of read out pixels see 10.5.40.

For the weCat3D MLSL profile sensor we have two deviating parameters which have influence on the calculation respectively the measuring rate: Height Z and Width X. When reducing the number of read out pixels in X and in Z, the measuring rate increases (see Fig. 53, measuring rate depending on height Z and width X at 1280 pixels (full frame), 640 pixels and 320 pixels).

The formula to calculate the approximate reachable measuring rate is:

$$f(a) = 0{,}454 * \ln(a)^3 + 178{,}357 * \ln(a)^2 - 5422{,}450 * \ln(a) + 39936{,}771$$
$$(a = \text{height } Z * \text{width } X)$$

NOTE!
The formula just gives an approximation, if the range of the height Z is 34...1024 and the range of the width X is 32...1280.
The measuring rate also depends on exposure time and the resources connected to the computer (see also AcquisitionLineRate in section 12.3.2).

NOTE!
The values of MLSL are also true for MLZL and M2SL.

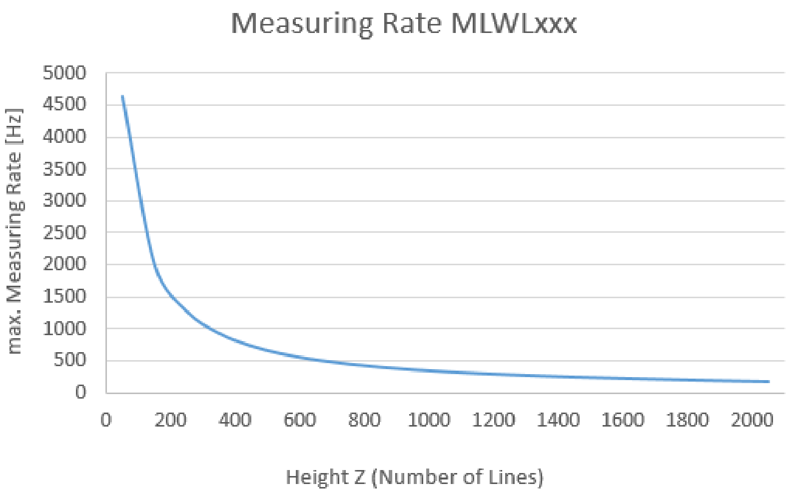


Fig. 54: MLWL: Measuring rate depends on height Z (number of lines)

Formula to calculate the measuring rate of the weCat3D MLWL Profile Sensor (see also Fig. 54):

*f(a) = -72,748 * ln(a)*^{3} *+ 1662,436 * ln(a)*^{2} *-12864,209 * ln(a) + 33862,688*
(a = height Z)

NOTE!
The formula just gives an approximation, if the range of the Height Z is 32...2048.
The measuring rate also depends on the exposure time and the resources connected to the computer.

NOTE!
For further explanations see the feature ExposureTime (see section 10.5.2) which also influences the measuring rate.

NOTE!
The maximum value range in AcquisitionLineTime feature (see section 10.5.3) will be updated dynamically according to the given ROI size.

NOTE!
The values of MLWL are also true for OPT3013, OPT3042 and MLWL033.

Profile Settings

Settings for exposure time, laser and measuring rate can be made via the "Profile Settings" tab (see Fig. 55).

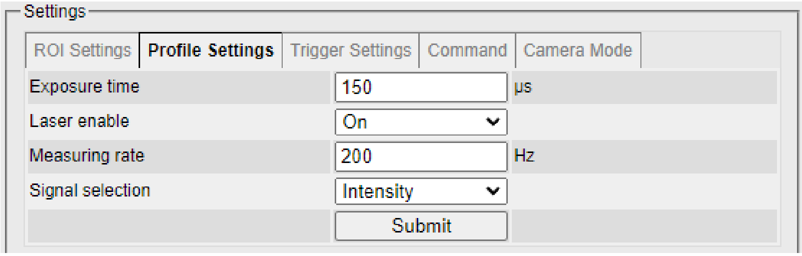


Fig. 55: Profile settings

Exposure time	Exposure time determines how much time the profile sensor takes for exposure at the internal camera. This parameter also controls laser on-time. The value is specified in microseconds. **NOTE!** At a intensity of less than 10%, exposure time should be increased for an ideal profile. At a intensity of greater than 90%, exposure time should be reduced for an ideal profile.
Laser	This function makes it possible to switch the laser on and off manually. **NOTE!** The exposure time can change by switching on and off the laser via the web interface.
Measuring rate	The measuring rate can be set when the “internal” sync mode is selected. The highest possible measuring rate, depending on the selected ROI, can be found in the “ROI Settings” section. **NOTE!** The measuring rate of the connected sensor is read out in the VisionApp Demo 3D software (DNNF013; can be downloaded for free at www.wenglor.com in the product area) depending on the selected ROI.

<table>
<tr>
<td>Signal selection</td>
<td>
All of the internal camera’s columns are searched for peaks. A peak is the signal that is created by the detected laser line in the column of the camera. If two or more peaks are detected in a single column, the user can specify the order in which the peaks are read out as distance value Z (see Fig. 56).

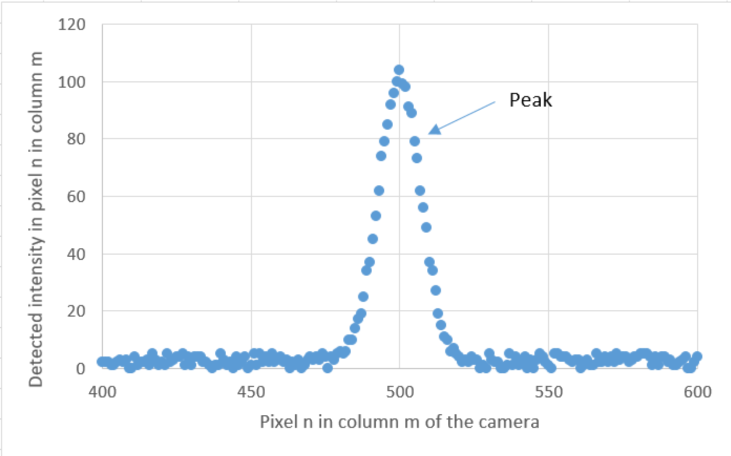


Fig. 56: Intensity curve in pixel in column m of the camera

NOTE!
This setting can be used to reduce reflections and other failures of the detected laser line.

The following selection criteria are available:

- **Intensity:** The brightest signal is read out as profile.
- **Width:** The widest signal is read out as profile. A wider signal occurs when the laser light penetrates more deeply into the object’s surface. Signal width is not shown in the profile display.
- **Peak 1:** The first peak is read out as profile.
- **Peak 2:** The second peak is read out as profile..
</td>
</tr>
</table>

Trigger Settings

Various settings for the profile acquisition can be made via the "Trigger settings" tab (see Fig. 57).

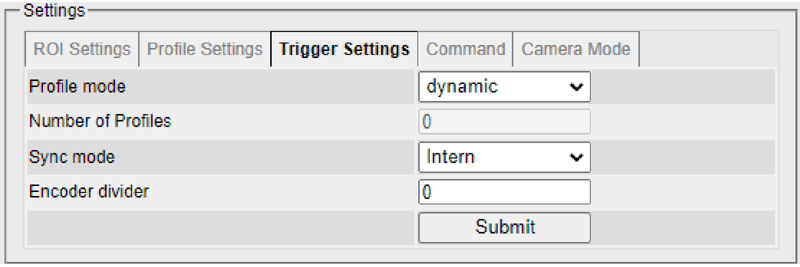


Fig. 57: Trigger settings

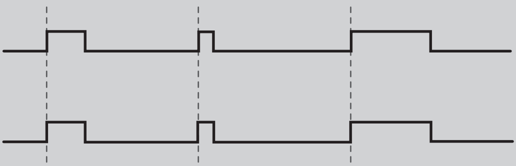
<table>
<tr><td>Profile mode</td><td>• Dynamic: The profile is recorded for as long as the trigger signal is applied (see Fig. 58).


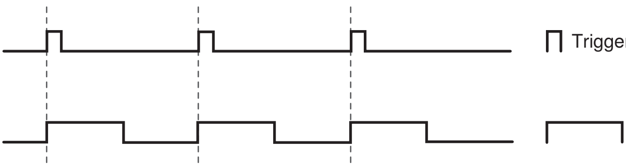
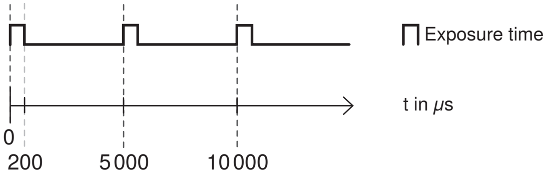
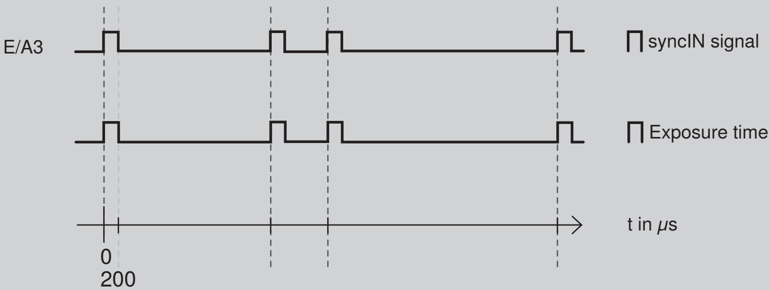
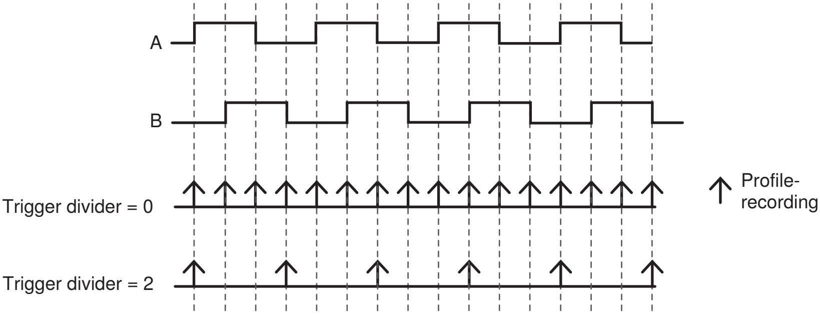
Fig. 58: Profile mode „Dynamic“</td></tr>
<tr><td>Profile mode</td><td>• Fix: After the number of profiles has been selected (see section 10.5.30), precisely this number of profiles is recorded after the start event occurs (the trigger source is selected in the Sync mode setting), and profile recording is stopped automatically until the next start event occurs (see also “profile enable” pin function and Fig. 59).


Fig. 59: Profile mode „Fix“</td></tr>
<tr><td>Number of profiles</td><td>Specifies how many profiles will be recorded before the sensor stops and can only be restarted by a command or an input signal. Can be combined with the “Internal”, “syncIN” and “Encoder” modes.</td></tr>
</table>

Sync mode	• Internal: The sensor is triggered internally via the set measuring rate. <u>Sample settings:</u> » Measuring rate: 200 Hz (i. e. one measurement every 5000 μs, see Fig. 60) » Exposure time: 200 μs  Fig. 60: Internal trigger
Sync mode	• syncIN: The sensor is triggered externally via the Sync In pin function, e. g. by another sensor or by an input signal. Only one profile per input signal is recorded and transmitted. <u>Example settings</u> (see Fig. 61): » Sync In pin function: E/A3 » Input function Ub active: Rising edge » Exposure time: 200 μs  Fig. 61: Trigger via syncIN **NOTE!** In the case of time-critical applications and strict timing requirements for triggering, the trigger signal via an E/A pin directly at the sensor must be used. **NOTE!** The applied input signal must be interference-free in order to assure flawless functioning of the sensor.

Sync mode	• Encoder: The sensor is triggered via encoder (HTL or RS422 TTL). Example settings: » Trigger source: Encoder (HTL or TTL) » Trigger divider: 0 and 2 (see Fig. 62)  Fig. 62: Trigger divider **NOTE!** Switching Back and Forth Between HTL and TTL Encoder (see “Encoder I1+I2” pin function). Any of the following encoder modes can be selected: • Motion (default): Sensor records profile independent of direction. • Position: Sensor records profile in a given direction. If the direction of motion is reversed, the last position is saved. New profiles are not recorded until the saved position has been passed. • Direction: Sensor records profile in one direction of motion only.
Sync mode	• Software: The sensor is triggered by a software command. The corresponding interface commands can be found in the documentation for the SDK.
Trigger divider	As a standard feature, the sensor is triggered by each pulse. This value can be used to specify how many pulses will be counted before profile recording is triggered. Possible values for trigger divider see section 10.5.25. Example: If "149" is entered under "Trigger divider", the sensor records a profile for pulse 150, 300, 450 etc. (see also Fig. 62). The maximum allowed input frequency at the I/O is 1 MHz.

Commands
Permits direct transmission of interface commands to the sensor (see Fig. 63). For further details please see the interface description in section 10).

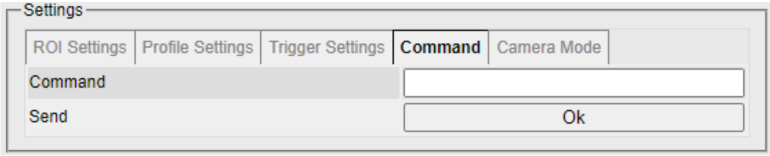


Fig. 63: Input field for interface commands

Camera Mode
In camera mode you can select profile or camera image (see Fig. 64).

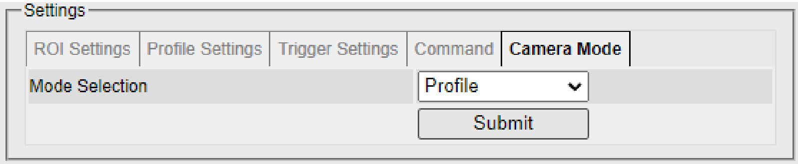


Fig. 64: Camera Mode

Mode selection	• Profile: The profile is shown/transfered via the interface and used software. • Camera image: The camera image is shown (transfer only via SDK).

7.2.4 I/O Settings

Various pin functions can be selected for the 4 configurable inputs/outputs.
Depending on the selected setting, context menus offer corresponding selection options (see Fig. 65).

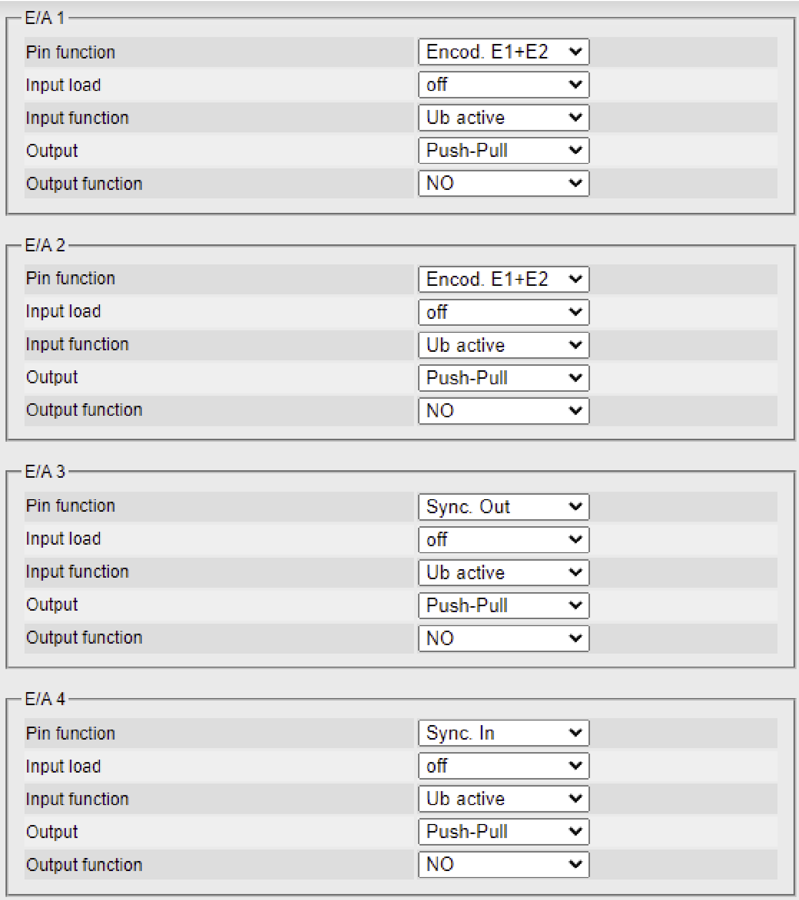


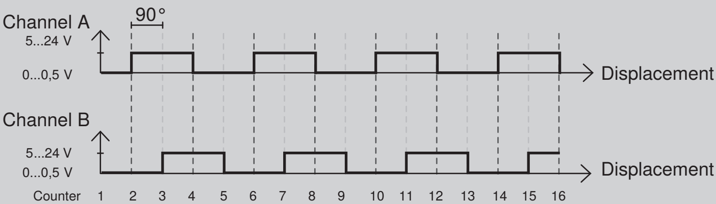
Fig. 65: I/O settings

<table>
<tr><td>Pin function</td><td>• Sync. In: Input function for synchronizing several sensors with each other, or for recording individual profiles with the help of pulses.

ATTENTION!
Exceeding the sensor's maximum measuring rate must be avoided (see section 10.5.40 and 7.2, Sensor status.</td></tr>
<tr><td>Pin function</td><td>• Sync. Out: Output function for synchronizing additional sensors. The sync out pin is connected with the sync in pin of other sensors.</td></tr>
<tr><td>Pin function</td><td>• User Input: Input function for querying the switching status of the selected input at the device via the software interface.</td></tr>
<tr><td>Pin function</td><td>• User Output: Output function for setting the output at the device via the software interface.</td></tr>
<tr><td>Pin function</td><td>

• Encod. E1+E2: Input function for connecting a HTL (5-24 V A/B channel) rotary encoder. This function must be set for E/A1 and E/A2 at the same time.
This function is only available for E/A1 and E/A2 (see Fig. 66).

HTL Encoder:



TTL Encoder:

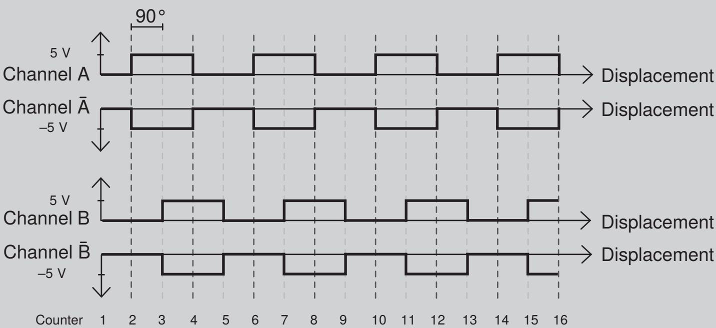


Fig. 66: TTL and HTL Encoder

NOTE!
TTL is active when pin functions E/A1 and E/A2 are not set to encoder. All four cables have to be connected (A; Ā; B; B̄), see section 6.3.1.

</td></tr>
</table>

Pin function	• Laser off: Input function for switching off the laser externally with the help of a 24 V signal. **ATTENTION!** This function does not constitute a safe laser shutdown.
Pin function	• Profile enable: Input function which enables profile recording for as long as the signal is applied. **NOTE!** In combination with the “fixed” mode, the “profile active” pin function is used to start the specified number of profiles. If the signal remains continuously active, the sensor transmits a multiple of the defined number of profiles (see also “Number of Profiles” under “Trigger Settings” in the “Profile Display” menu).
Pin function	• Encod. reset: Input function for resetting the internal encoder counter to “0”.
Internal load	Connects an internal resistor to the input (pull-down). Internal load of 2 mA (on/off)..
Input function	Determines whether the input responds to supply voltage or 0 V. This makes it possible to invert any pin function.
Output	Sets the output’s polarity (push-pull, PNP, NPN).
Output function	The output can be configured as normally open (NO) or normally closed (NC).

NOTE!
The above listed functions can be set individually for each of the 4 configurable I/Os (with the exception of “Encod. E1+E2” which is limited to I/O1 in combination with I/O2.

7.2.5 Visualization

In the "Smart WeCat3D" operating mode, a link to uniVision web is displayed on the website. This website can be configured by the user and is used to display the results of the profile analysis. For further information please see the operating instructions for the uniVision software DNNF012/DNNF020.

7.3 Firmware Update

ATTENTION:
The firmware of the sensor should be 1.2.0 or higher before updating the sensor to firmware version 2.0.0. If the firmware is lower than 1.2.0 update to the latest 1.2.x (see download area of the product page on wenglor.com). How to update firmware see the update note included in the firmware package. The current firmware version can be found on the sensor webpage (see operating instructions for details).

ATTENTION:
Updating a sensor with firmware version 1.x.x to 2.0.0 can also lead to an update of the linearization table due to changes in the format of the linearization table.

The update process can take several minutes!

Before you update the firmware please close all software applications connected to the sensor. We recommend to switch off the sensor and restart it again. As soon as the sensor is online you can start the update procedure:

Accessing the integrated website:
Start your web browser and enter the preset IP address (192.168.100.1) into the browser.
Add „administration.html" after the IP address and press Enter (see Fig. 67).

Fig. 67: Web address for update process

If the actual IP address differs from the preset address and you don't know the current address, you can view it at the OLED display after selecting the „Interface" menu item.

Click „Choose a file" (1) and select the file „Firmware_weCat3D_xxx.run". The file can be downloaded at www.wenglor.com from the product's download area.
Then acknowledge by clicking „Update" (2) and wait until updating has been completed (see Fig. 68 ... Fig. 70).

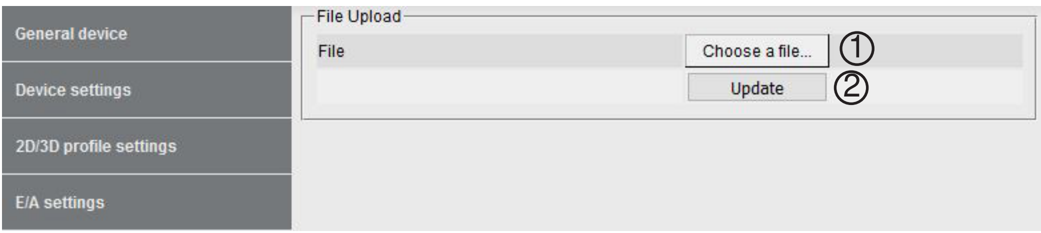


Fig. 68: Start window update process

ATTENTION:
Supply power may not be interrupted and the system may not be restartet during the update procedure. The device may otherwise be destroyed.

During the update procedure the actual status is displayed in the Update area:

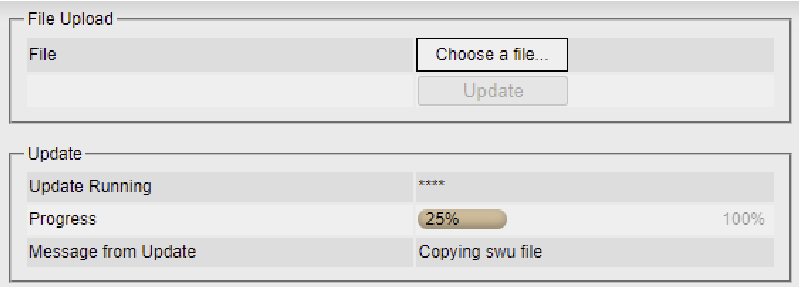


Fig. 69: Status display of update process

When the update is completed the following display appears:

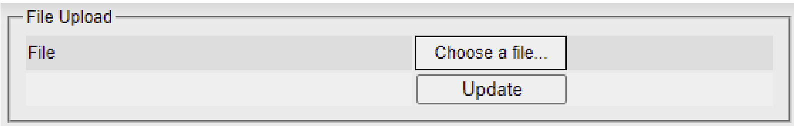


Fig. 70: Update process finished

After the successful update the current firmware version is shown on the "General device" page.

In the case that the linearization table must be updated a message is shown on the web page (see Fig. 71):

Fig. 71: Message for invalid linearization table

Use the Firmware_weCat3DAddon_1.0.0.run in the firmware package and update the sensor with this package again. The process is equal to the described update process above.
After the update the message on the sensor web page should be disappeared. If not please contact the technical support of wenglor.

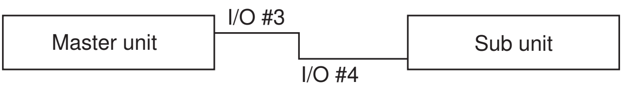
7.4 Using more than one Sensor (Sychronization)

The synchronization of several 2D / 3D profile sensors is necessary when the laser lines of the sensors are in the same field of view and thus influence each other.

NOTE!
A 2D/3D Profile Sensor with red laser light and a 2D/3D Profile Sensor with blue laser light do not influence each other.

Synchronization of two 2D/3D Profile Sensors:
Wire the two profile sensors together so that an I/O pin of the first sensor (master unit) is connected to an I/O pin of the second sensor (sub unit).

Example: I/O #3 of the master unit is connected with I/O #4 of the sub unit:



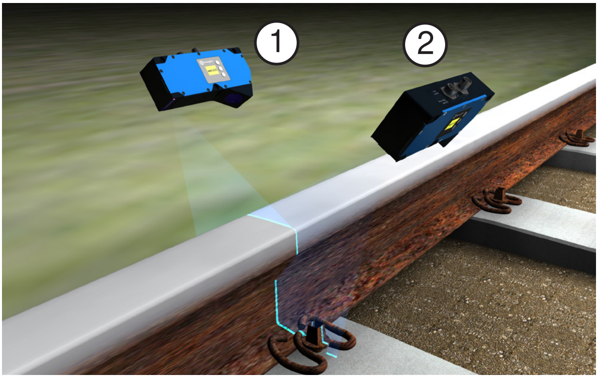
NOTE!
The master unit must be set in sync mode „Encoder“, „Intern“ or „Software“, the sub unit must be set in sync mode „syncIN“.

Set an I/O pin of the master unit with a time delay as an output. The delay should at least correspond to the exposure time of the master unit. The length of the output signal must not exceed the exposure time of the sub unit.

Pin assignment:

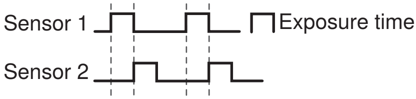
Pin	Input/Output	Function	Color
5	I/O3	sync out	pink
6	I/O4	sync in	yellow

Example 1:
Time-shifted measurement for extending the measuring range while avoiding reciprocal influence of the sensors despite overlapping of the laser lines.



Sample application: Rail head measurement

Sample configuration:
Master unit, SyncOut (default setting E/A3) connected to sub unit, SyncIn (default setting E/A4).

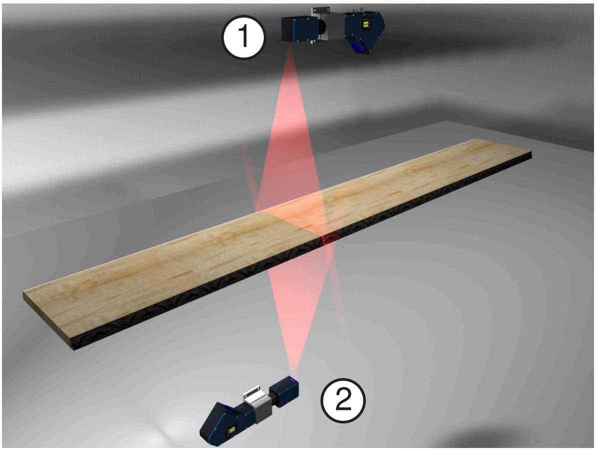


Sensor 1:
Sync mode (Intern/Encoder/Software)
Exposure time 200 μs
SyncOut delay = 200 μs*

Sensor 2:
Sync mode (syncIN)
Exposure time 200 μs

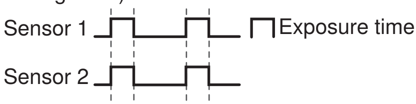
*(see section 10.5.32, the command can be entered via website).

Example 2:
Simultaneous measurement:



Sample application: Thickness measurement of wooden floorboards

Sample configuration:
Master unit, SyncOut (default setting E/A3) connected to sub unit, SyncIn (default setting E/A4)



Sensor 1:
Sync mode (Intern/Encoder/Software)
Exposure time 200 μs
SyncOutDelay = 0 μs

Sensor 2:
Sync mode (syncIN)
Exposure time 200 μs

NOTE!
The signal width must be at least half the period of the measuring rate.

8. OLED Display

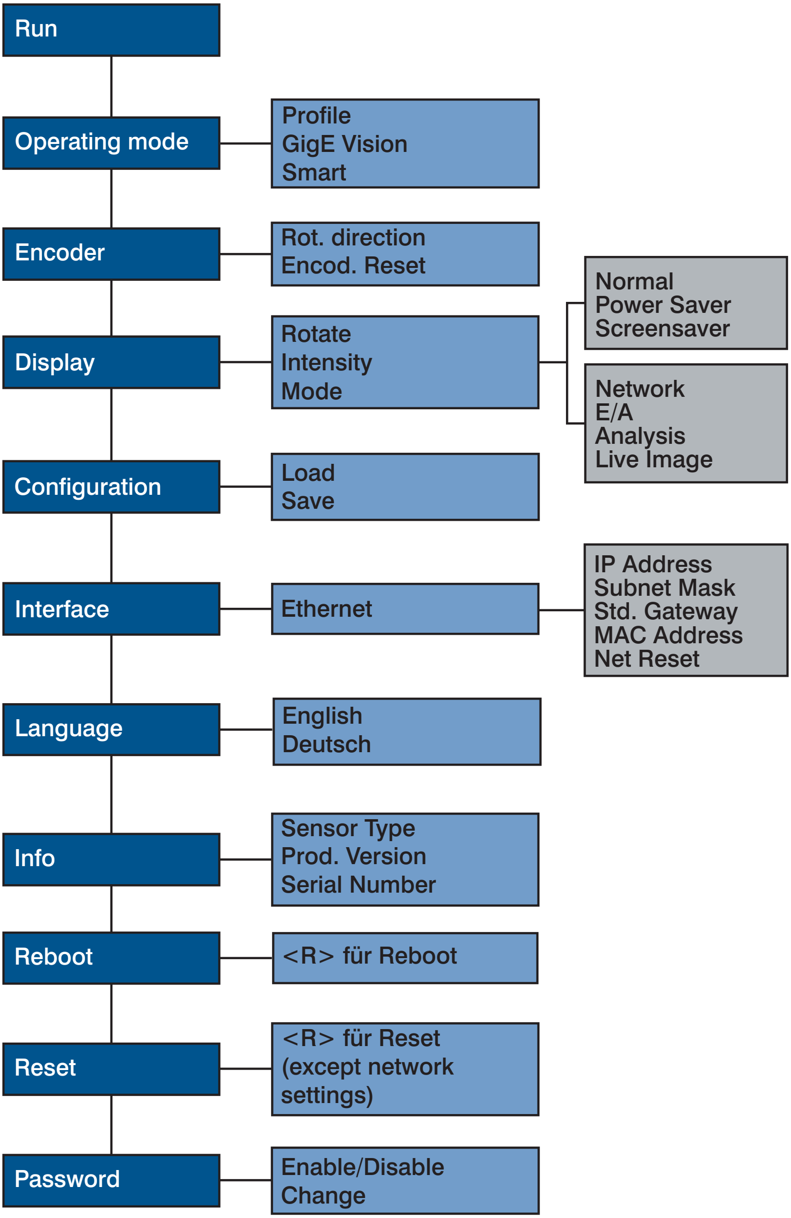
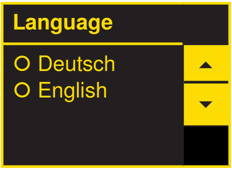


Fig. 72: Structure of OLED display

8.1 Settings

A language can be selected for the OLED display from the language selection. This has no effect on the internal website and is saved to the sensor automatically.



Navigation with the keys:

▲ : Navigate up

▼ : Navigate down.

↵ : Selection is acknowledged with the enter key.

Meanings of menu items:

◂ Back: Move up one level within the menu.

◂◂ Run: Switch to the display mode.

You can switch to the configuration menu by pressing any key.

NOTE!
If no settings are adjusted in the configuration menu for a period of 30 seconds, the sensor is automatically returned to the display mode. The sensor accesses the last used menu view when a key is once again activated. If a setting is configured, it becomes active when the configuration menu is exited.

CAUTION!
Do not use any sharp objects to press the keys when configuring settings, because they might otherwise be damaged.

8.2 Run

The sensor is switched to the display mode. Further information is available in section 8.5.3. The display mode can be changed to the network mode, the I/O display or the Analysis mode in the display menu.

8.3 Operating Mode

Operating mode	Setting the operating mode of the sensor	
O Profile	**Profile:**	Sensor works in operating mode "Profile".
O GigE Vision	**GigE Vision:**	Sensor works with embedded GigE Vision interface.
O Smart	**Smart:**	Sensor works in conjunction with the uniVision software.
◂ Back		(Notes in section 7.2.2 must be considered!)
◂◂ Run		

8.4 Encoder

Encoder	Setting the encoder's direction of rotation	
O Rot. Direction O Encod. Reset ◂ Back ◂◂ Run	**Rot. direction:**	Rising: the encoder's direction of counting is ascending. Falling: the encoder's direction of counting is descending.
	Encoder reset:	Encoder settings are reset.

8.5 Display

Various changes can be made to the settings at the display in order to simplify operation of the sensor.

8.5.1 Rotate

Rotate	180° rotation of the display

8.5.2 Intensity

Display intensity can be adjusted, for example to assure that the display is still easily legible even in bright environments.

Display	Display setting	
O Normal O Power Saver O Screensaver ◂ Back ◂◂ Run	**Normal:**	Display intensity is set to the middle value.
	Power saver:	If no keys are pressed for a period of one minute, the display is switched off, and is switched back on as soon as a key is activated.
	Screen saver:	If no keys are pressed for a period of 30 seconds, the display is switched to the display mode and returns to the last used menu as soon as a key is activated. The colors are inverted every 30 seconds in order to protect the display..

8.5.3 Mode

The sensor is equipped with various display modes which appear in the run display.

Mode	Selection of the display for the "Run" mode	
O Network O E/A O Analysis O Live Image ◂ Back ◂◂ Run	**Network:**	The IP address, MAC adress and the subnet mask are displayed.
	E/A:	Display of input and output states.
	Analysis:	Displays network utilization as a percentage, internal tempera-ture in °C and intensity as a percentage.
	Live Image:	Displays the current profile image.

8.6 Configuration

Configuration	Sensor configuration management	
O Load	**Load:**	Stored sensor settings are loaded.
O Save	**Save:**	Sensor settings are saved.
◂ Back		
◂◂ Run		

8.6.1 Load

Load	Loading the sensor configuration	
O Standard	**Standard:**	The values saved under standard are loaded automatically when the sensor is started up.
O Set 1	**Set 1:**	The values saved to “Set 1” are loaded.
O Set 2	**Set 2:**	The values saved to “Set 2” are loaded.
◂ Back		
◂◂ Run		

8.6.2 Save

Save	Saving the sensor configuration	
O Standard	**Standard:**	Sensor settings are saved under “Standard”.
O Set 1	**Set 1:**	Sensor settings are saved under “Set 1”.
O Set 2	**Set 2:**	Sensor settings are saved under “Set 2”.
◂ Back		
◂◂ Run		

8.7 Interface

Ethernet	Settings for the Ethernet connection	
IP Address	**IP Address**	Display of the set IP address
Subnet Mask	**Subnet Mask**	Display of the set subnet mask
Std. Gateway	**Std. Gateway**	Display of the set standard gateway
MAC Address	**MAC Address**	Display of the set and unchangeable Mac address
Net Reset	**Net Reset**	Reset of the network settings to factory default
◂ Back		
◂◂ Run		

NOTE!
Changes do not become effective until after the sensor has been restarted.

8.7.1 IP Address

IP Address		Setting the IP address
192.168.100.001	+ ↵ –	The IP address can be set by pressing the “+” and “–” keys.

IP Address		Checking the IP address for correctness
192.168.100.001 Input correct?	Y N ◀	Correctness of the entered IP address is confirmed by pressing the “**Y**” key and the address is transferred to the sensor. If necessary, the IP address can be reentered after pressing the “**N**” key. After pressing the ◀ key, the display is returned to the Ethernet network menu without saving the entered IP address.

The subnet mask, the standard gateway and the TCP/IP port can be changed using the same procedure as for the IP address.

NOTE!
After change of the IP address the sensor will reboot.

8.7.2 MAC Address

MAC Address		Displaying the MAC address
54:4a:05:00:08:04	◀ ↵ –	The sensor’s unchangeable MAC address is displayed. After pressing the ◀ key, the display is returned to the Ethernet network menu.

8.7.3 Network Reset

Network Reset		Resetting the network configuration
Press <R> for Reset	R ◀	The network configuration can be reset by pressing “**R**”. After pressing the ◀ key, the display is returned to the Ethernet network menu.

See default settings in section 6.5.

8.8 INFO

Info	Display of sensor information
Sensor type MLSL123 Product version 1.0.0 Serial number 123456789	Sensor type, product version, serial number and status are displayed in the information menu. These entries play an important role in the event that technical problems should occur, and when contacting Technical Support with questions.

8.9 Restart

Restart		Restart of the sensor
Press <R> for restart	R ◄	Restarting of the sensor can be forced by pressing “R”. After pressing the ◄ key, the display is returned to the main menu..

8.10 Reset

Sensor settings (except for network settings) can be returned to their default values in the “Reset” menu (see section 6.5).

Reset		Restoring default settings
Press <R> for reset	R ◄	All of the selected sensor settings are returned to their default values by pressing the “R” key, except network settings. The settings in Set0, Set1 and Set2 are deleted. After pressing the ◄ key, the display is returned to the main menu.

8.11 Password

Password protection prevents inadvertent changes to selected settings.

Password	Activating the password function	
Enable/disable Change ◂ Back ◂◂ Run	**Enable/ Disable:**	Enable or disable password protection. When password protection is enabled, the sensor is automatically disabled after pressing any key in the “Run” mode.
	Change:	Change the password.

NOTE!

- If the password function has been activated, the password must be entered each time supply power is interrupted. After pressing any key, the menu is automatically switched to the password entry mode.
- After the password has been correctly entered, the entire menu is enabled and the sensor can be operated. The password function is deactivated upon shipment from the factory.
  Passwords can be selected within a range of **0000 to 9999.**
- It must be assured that the selected password is noted before any changes occur.
  If the password is forgotten, it has to be overwritten with a master password.
  The master password can be requested by e-mail from **support@wenglor.com**.

9. Other Devices

9.1 Application Dedicated Series

9.1.1 MLZL

The sensors of the MLZLxxx series are weCat3D sensors optimized for welding applications. Further information can be found in the operating instructions "Optical Seam Tracking Solutions" (see product page on wenglor.com).

9.1.1.1 Technical Data

	MLZL121	MLZL131	MLZL141	MLZL151	MLZL171
Optical Data					
Working range Z	74...158 mm				
Measuring range Z	84 mm				
Measuring range X	38...62 mm				
Linearity deviation	65 μm				
Resolution Z	8,3...32,5 μm				
Resolution X	32...64 μm				
Light source	Laser (red)	Laser (blue)	Laser (red)	Laser (blue)	Laser (blau)
Wavelength	690 nm	450 nm	690 nm	450 nm	450 nm
Laser class (EN 60825-1)	2M	2M	3R	3R	3B
Environmental conditions					
Ambient temperature	0...45 °C				
Storage temperature	-20...70 °C				
EMC	DIN EN 61000-6-2; 61000-6-4				
Shock resistance (DIN IEC 68-2-27)	30 g / 11 ms				
Vibration resistance (DIN IEC 60068-2-6)	6 g (10...55 Hz)				
Atmospheric humidity	5...95 %, non-condensing				
Cooling system*					
Medium	Water, air (oil free)				
Temperature of medium	Water: 15 °C Air: 20 °C				
Flow rate	Water: 0,55 l/min (max. 1 l/min) Air: 30 l/min				
Pressure resistance	max. 2 bar				
Process connection	Push-in fitting (4 mm)				
Purge					
Medium	Air (oil free)				
Pressure resistance	max. 6 bar				
Process connection	Push-in fitting (4 mm)				
Electrical Data					
Supply voltage	18...30 V DC				
Current consumption (Ub = 24 V)	300 mA	330 mA	300 mA	330 mA	1000 mA
Measuring rate	200...4000/s				

	MLZL121	MLZL131	MLZL141	MLZL151	MLZL171
Measuring rate (subsampling)	800...4000/s				
Inputs/Outputs	4				
Switching output voltage drop	< 1,5 V				
Switching output/switching current	100 mA				
Switchable to NC/NO	yes				
Configurable as PNP/NPN/Push-Pull	yes				
Short-circuit protection	yes				
Reverse polarity protection	yes				
Overload protection	yes				
Interface	Ethernet TCP/IP				
Baud rate	100/1000 Mbit/s				
Protection class	III				
Mechanical Data					
Housing material	Aluminium, anodised				
Optic cover	Plastic, PMMA				
Degree of protection	IP67				
Connection	M12×1; 12-pin				
Connection Ethernet	M12×1; 8-pin, X-coded				
Weight	560 g				
General Data					
Webserver	yes				

* see section 9.1.1.2 for detailed information

9.1.1.2 Cooling Parameters

Water cooling:

Temperature of medium (°C)	Max. ambient temperature (°C)
25	100
20	110
15	120
10	130
5	140

The indicated values refer to following conditions:

- Closed system
- Flow rate: 0,55 l/min
- Pressure: 1,6 bar

Air cooling:

Temperature of medium (°C)	Max. ambient temperature (°C)
approx. 20 °C	70

The indicated value refers to following conditions:

- Open system
- Flow rate: 30 l/min

9.1.1.3 Complementary Products

Part number	Description
ZLSE010	Screening grid, 10 pcs., plastic
ZLSE011	Screening grid, 10 pcs., glass
ZLSE012	Screening grid retainer
ZLSE013	Guard plate

9.1.1.4 Measuring Field X, Z

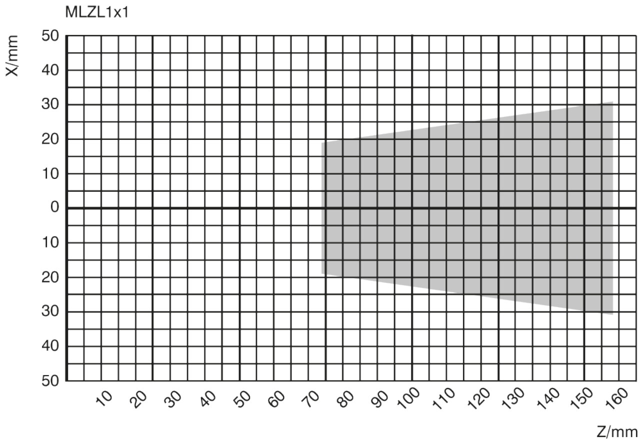


Fig. 73: Measuring field MLZL1x1

9.1.1.5 Dimensional Drawings

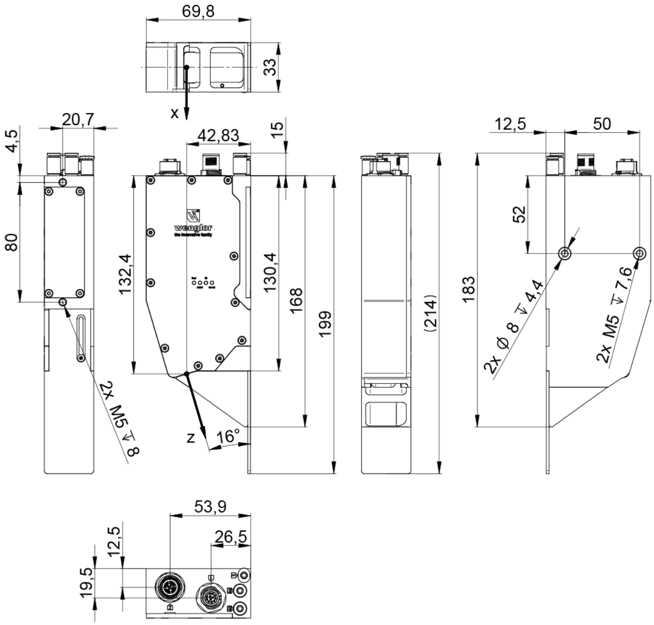
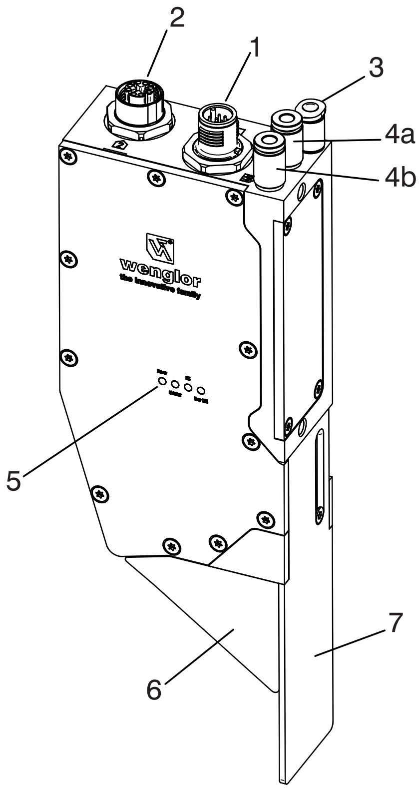


Fig. 74: Dimensional drawings MLZL1x1

9.1.1.6 Sensor Construction



1	= Power supply plug connector, digital I/O
2	= Connection socket Ethernet
3	= Air purging
4a	= Air/water cooling IN
4b	= Air/water cooling OUT
5	= LED display
6	= Screening grid retainer with screening grid
7	= Guard plate

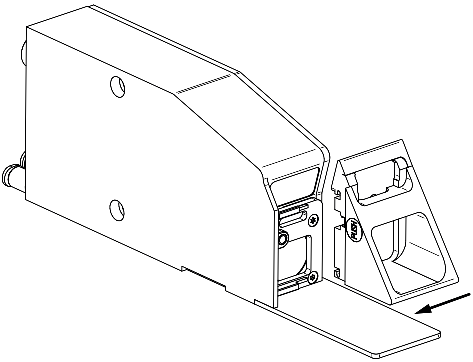
Fig. 75: Sensor construction of MLZL1x1

9.1.1.7 Pin Assignment

The pin assignment corresponds to the standard assignment of the weCat3D sensors, see section 6.3.1. The connections of cooling and purging are designed for tubes with an outer diameter of 4 mm.

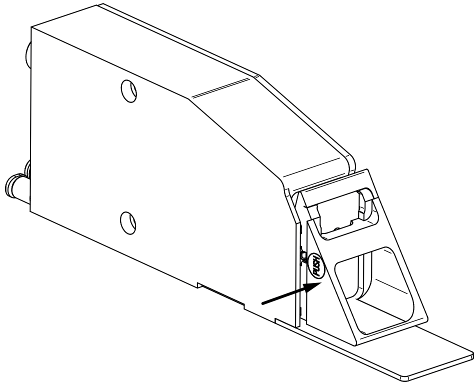
9.1.1.8 Installation of Screening Grid Retainer and Screening Grid

Assembly of the screening grid retainer:



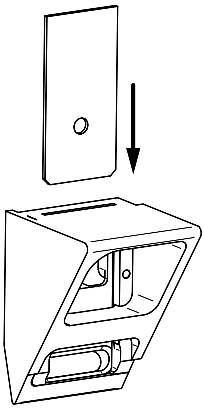
Insert the screening grid retainer into the guide bar of the sensor (see picture above) and slide it to the left until it snaps in.

Disassembly of the screening grid retainer:



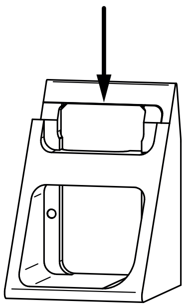
By pressing the push button, the screening grid retainer is removed from the fixation and can be slided out to the right (see picture above).

Inserting the screening grid:



Slide the screening grid into the relevant slot, bevelled edges first (see picture above).

Removing the screening grid:



Place a pointed object (e. g. fingernail) on the upper edge of the screening grid and slide it out of the screening grid retainer (see picture above).

NOTE!
Before inserting the plastic screening grid (ZLSE010) please remove the protective foil on both sides.

9.1.2 M2SL

The sensors of the M2SL series have a stainless steel housing and they are weCat3D sensors adapted to the food and pharmaceutical sectors.

The sensors meet the requirements of

- ECOLAB
- IP67/IP69K

ATTENTION:
When using high-pressure cleaners, make sure that the cables are not damaged by the water jet.

9.1.2.1 Product Overview

Below is an overview of all available M2SLxxx sensors and their corresponding standard products.

M2SLxxx	Standard Product MLSLxxx
M2SL225	MLSL225
M2SL235	MLSL235
M2SL226	MLSL226
M2SL236	MLSL236

9.1.2.2 Technical Data

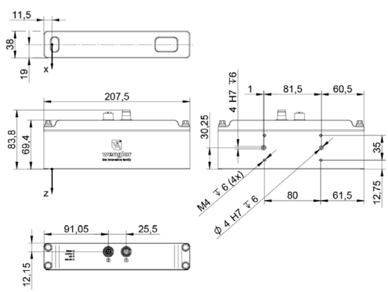
	M2SL225	M2SL235	M2SL226	M2SL236
Optical Data				
Working range Z	280...1280 mm	280...1280 mm	300...1500 mm	300...1500 mm
Measuring range Z	1000 mm	1000 mm	1200 mm	1200 mm
Measuring range X	200...850 mm	200...850 mm	250...1350 mm	250...1350 mm
Linearity deviation	500 μm	500 μm	600 μm	600 μm
Resolution Z	40...570 μm	40...570 μm	60...990 μm	60...990 μm
Resolution X	190...760 μm	190...760 μm	270...1170 μm	190...1170 μm
Light source	Laser (red)	Laser (blue)	Laser (red)	Laser (blue)
Wave length	660 nm	405 nm	660 nm	405 nm
Laser class (EN 60825-1)	2M	2M	2M	2M
Electrical Data				
FDA Accession Number	2111542-000	1610468-004	1710959-001	1610468-004
Supply voltage	18 ... 30 V DC			
Current consumption (Ub = 24 V)	1000 mA			
Measuring rate	200 ... 4000/s			
Measuring rate (subsampling)	800 ... 4000/s			
Inputs/Outputs	4			
Switching output voltage drop	< 1,5 V			
Switching output/switching current	100 mA			
Switchable to NC/NO	yes			
Configurable as PNP/NPN/Push-Pull	yes			
Short-circuit protection	yes			
Reverse polarity protection	yes			

	M2SL225	M2SL235	M2SL226	M2SL236
Overload protection	yes			
Interface	Ethernet TCP/IP			
Baud rate	100 / 1000 Mbit/s			
Protection class	III			
Environmental conditions				
Ambient temperature	0 ... 45 °C			
Storage temperature	-20...70 °C			
EMC	DIN EN 61000-6-2; 61000-6-4			
Shock resistance DIN IEC 60068-2-27	30 g / 11 ms			
Vibration resistance DIN IEC 60068-2-6	6 g (10...55 Hz)			
Atmospheric humidity	5...95 %, non-condensing			
Mechanical Data				
Housing material	Stainless steel, V4A (1.4404 / 316L)			
Optic cover	Plastic, PMMA			
Degree of protection	IP67 / IP69K			
Connection	M12 × 1; 12-pin			
Connection Ethernet	M12 × 1; 8-pin, x-coded			
Weight	2100 g			
General Data				
Webserver	yes			

9.1.2.3 Measuring Fields X,Z

The measuring fields correspond to those of the standard products, see section 9.1.2.1 and 4.1.

9.1.2.4 Dimensional Drawings



9.1.2.5 Pin Assignment

The pin assignment corresponds to the standard assignment of the weCat3D Sensors, see section 6.3.1.

9.1.3 MLSL123S50

NOTE!
Unless otherwise stated below, the information given in these operating instructions applies (reference device: MLSL123).

The MLSL123S50 is already preset to Smart Sensor operating mode ex works and contains sample projects from the press brake control manufacturer Delem (Esautomotion and Cybelec coming soon).

With the help of the sample projects, the sensors can communicate with the respective press brake control systems without changing the configuration of the devices.

Only the IP addresses need to be adapted before initial start-up.

To be able to connect the sensor to your press brake control, you must ensure that the sensor and the control are in the same IP address range. In addition, the device parts of the IP addresses for the two sensors must be different. You can change the IP address of the sensor via the integrated web server (section 7) or alternatively directly via the control panel (OLED display) on the device without a connection to a PC (section 8).

NOTE!
Technical data and pin asignment correspond to the specifications for the 2D-/3D profile sensor MLSL123.

NOTE!
For information on the connection to the control system, please refer to the operating instructions of the respective press brake controller.

ATTENTION:
The installed firmware must not be replaced or updated with a different version. Otherwise the functionality of the sensor is no longer guaranteed.

9.2 Special Devices

9.2.1 OPT3013

NOTE!
Unless otherwise stated below, the information given in these operating instructions applies (reference device: MLWL1x2).

9.2.1.1 Use for Intended Purpose

The OPT3013 may only be used for the measurement of materials. It's not suitable for measuring living beings because the skin may be endangered if the system comes to a standstill. The sensor must move at a speed of at least 0.3 mm per second in order to assure safe operation. This prevents inadvertent irradiation of the skin at any given point. Use of the product for other than its intended purpose may be hazardous to the skin. Use of the product for other than its intended purpose is deemed in improper. Any liability on the part of the manufacturer is excluded in this case.

9.2.1.2 Minimum coverage of the visual field width

The visual field always has 2048 points over its entire width. A point is defined by an X/Z coordinate and an intensity value. The points are classified as valid or invalid:

Invalid points:	*X-value = 0*	*Z-value = 0*	*intensity = 0*
Valid points:	*X-value ≠ 0*	*Z-value ≠ 0*	*intensity ≠ 0*

When using the OPT3013, 5% of the valid points within the visual field width must be detected in order to enable measurement at full speed, i.e. the sensor switches from flash mode to measuring mode after at least 105 points have been detected.

9.2.1.3 Technical Data

Light source	Laser (UV) / laser (red)
Wavelength	375 nm / 660 nm
Laser class UV/red (EN 60825-1:2014)	1/2
Current consumption (Ub = 24 V)	1500 mA
Measurement enable *	EA3: 5…24 V DC
Trigger *	EA1 + EA2: encoder signal TTL or HTL **OR** EA4: Frequency proportional to movement speed
Weight	600 g

* see flowchart "Standard Operation" below

9.2.1.4 OPT3013 Safety Clearances

In accordance with laser standard EN 60825-1:2014, NOHD (distance as of which laser class 1 is achieved) amounts to 3.4 meters.

In accordance with TROS, which takes skin safety into consideration in addition to eye safety, safety clearance is 15 meters. Country-specific safety clearances can be calculated from the irradiance values listed in the table shown below.

Abbreviations:

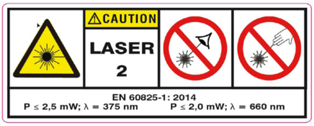
NOHD: Nominal ocular hazard distance

TROS: Technical rules of the German occupational safety regulation concerning artificial optical radiation

9.2.1.5 Irradiance of UV Light

Distance [m]	0.1	1.0	1.5	2.0	2.5	3.0	3.5	4.0	4.5	5.0
W/m^2	2673.8	67.5	29.9	16.9	10.8	7.6	5.6	4.3	3.4	2.8

Distance [m]	6.0	7.0	8.0	9.0	10.0	11.0	12.0	13.0	14.0	15.0
W/m^2	1.9	1.4	1.1	0.9	0.7	0.6	0.5	0.4	0.4	0.3

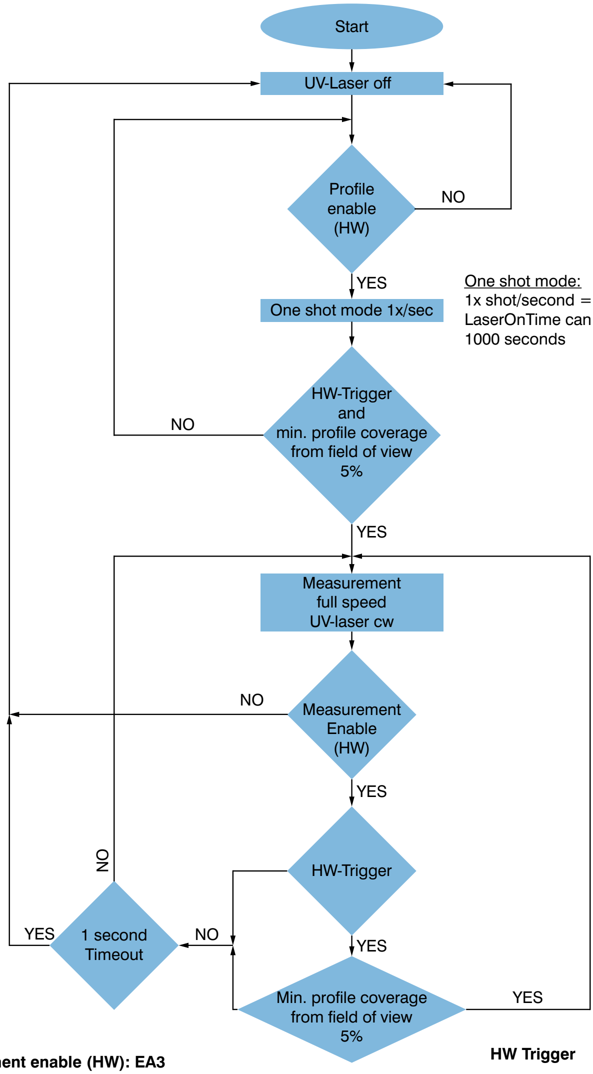


LASER ENERGY - EXPOSURE IN CLOSE PROXOMITY TO THE APERTURE MAY CAUSE INJURY TO THE SKIN!

CAUTION!
In accordance with TROS Laser 2015 (Germany), the skin may not be statically exposed to UV radiation for longer than 13 seconds at a distance of 100 mm. We recommend wearing gloves during use in the setup mode. Red laser light (660 nm ± 10 nm) may not be obstructed!

NOTE!
Adequate heat dissipation must be assured. If the sensor is operated with the default exposure time set at the factory, a metallic connection between the sensor housing and the mounting surface is sufficient to this end. As of an internal temperature of 56 °C, the sensor must be cooled with the help of the suitable cooling element (see MLWL1x2 standard device). If internal temperature rises to greater than 61 °C, the sensor's laser diodes (red and UV) are shut down automatically in order to protect them. If temperature drops to below 59 °C, automatic shutdown is once again deactivated.

9.2.1.6 Standard Operation



One shot mode:
1x shot/second = 10 ms LaserOnTime
LaserOnTime can be max. 13.1 seconds in 1000 seconds

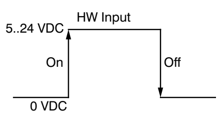
Measurement enable (HW): EA3

• Signal edge controlling:

Low → High edge = Loop On
(must stay high)

High → Low edge = Loop Off
(must stay low)

Condition: New pulse to turn on regardless of reason of switching off



HW Trigger

• Movement control signal:

1. Encoder signal on both inputs (TTL or HTL / EA1+EA2)

OR

2. Frequency on I/O pin proportional to movement speed (EA4)

Condition: Signal must have a minimum frequency of 1 Hz

Fig. 76: Flow chart of OPT3013 standard operation

9.2.2 OPT3042

NOTE!
Unless otherwise stated below, the information given in these operating instructions applies (reference device: MLWL225).

9.2.2.1 Technical Data

Working range Z	1450...2050 mm
Measuring range Z	600 mm
Measuring range X	200...280 mm
Linearity deviation	150 μm
Resolution Z	25...49 μm
Resolution X	105...146 μm

9.2.2.2 Measuring Field X, Z

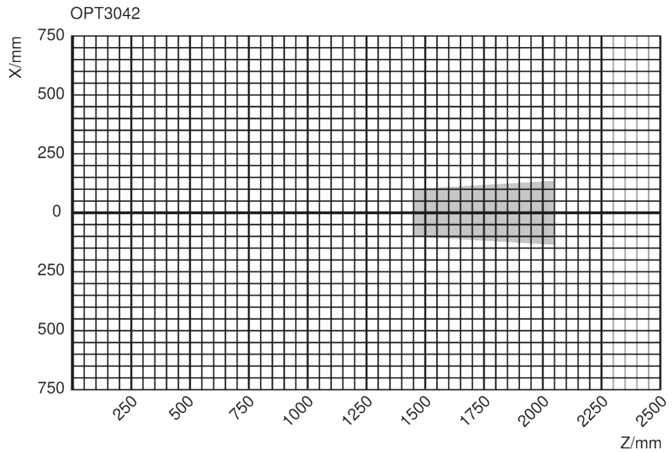


Fig. 77:Measuring field OPT3042

9.2.3 MLWL033

NOTE!
Unless otherwise stated below, the information given in these operating instructions applies (reference device: MLWL233).

9.2.3.1 Technical Data

Degree of protection	IP69K
Housing material	Stainless steel V2A (1.4305 / 303)
Optic cover	Plastic, PMMA
Cable jacket material	Plastic, TPE
OLED display	no
Cover seal	Silicon blue
Bending radius	87 mm
Cable length	5 m

ATTENTION:
When using high-pressure cleaners, make sure that the cables are not damaged by the water jet.

9.2.3.2 Pin Assignment

The connection cables are firmly connected with the sensor housing.
The assignment of the open ends is as follows:

Connection supply:

Color	Description
brown	+24 V
blue	0 V
white	E/A1 - HTL signal
green	E/A2 - HTL signal
pink	E/A3
yellow	E/A4
black	En A
grey	En $\bar{A}$
red	En B - TTL signal
violet	En $\bar{B}$ - TTL signal
grey/pink	En O - TTL signal
red/blue	En $\bar{O}$ - TTL signal

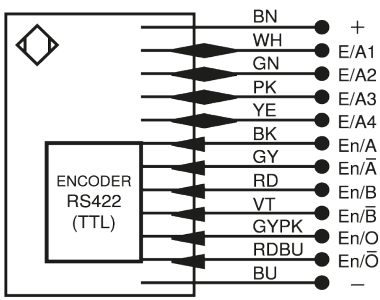


Fig. 78: Pin assignment of connection supply

Connection Ethernet:

Color	Description
white/orange	Bi_DA +
orange	Bi_DA –
white/green	Bi_DB +
green	Bi_DB –
white/brown	Bi_DD +
brown	Bi_DD –
blue	Bi_DC +
white/blue	Bi_DC –

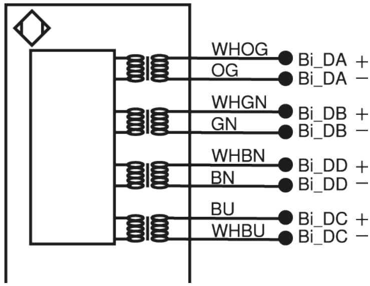


Fig. 79: Pin assignment of Ethernet connection

9.2.3.3 Dimensional Drawings

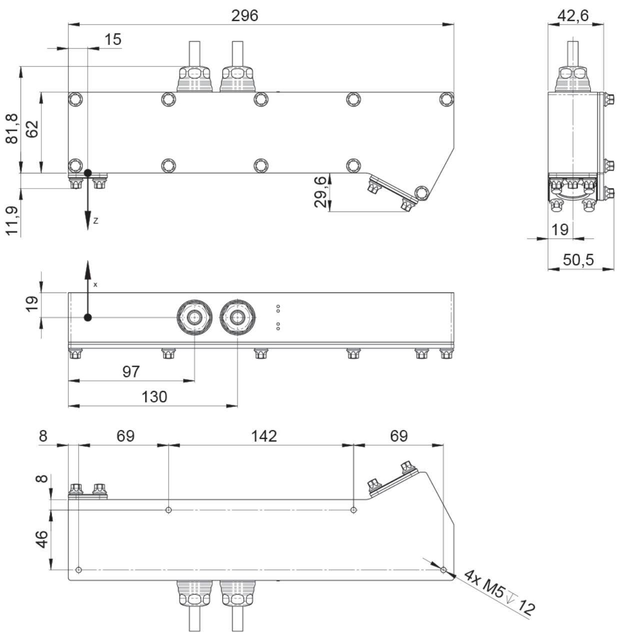


Fig. 80: Dimensional drawings MLWL033

10. Software Development Kit (SDK)

10.1 Introduction

This section describes the functions and the commands using the library provided by the Software Development Kit (SDK) to realize custom application development for the weCat3D product series.

10.2 System Requirements

Applications development with the DLL/shared library requires a Microsoft operating system (WIN7 x64, WIN10 x64)/ Linux (Ubuntu x64,14.04 or higher).
The weCat3D product series requires a 1 Gigabit network interface card.

The SDKs are available for download at www.wenglor.com in the product's separate download area.
The SDK is distributed through different packages, each package provides an example project on how to use the SDK functions.

10.3 Application Example

Each SDK has a demo project with the source code. The demo application is given as a mean to demonstrate the data transmission from the profile sensor to the application using the SDK functions (see Fig. 81, screenshot of the SDK_Windows_QT_C++).

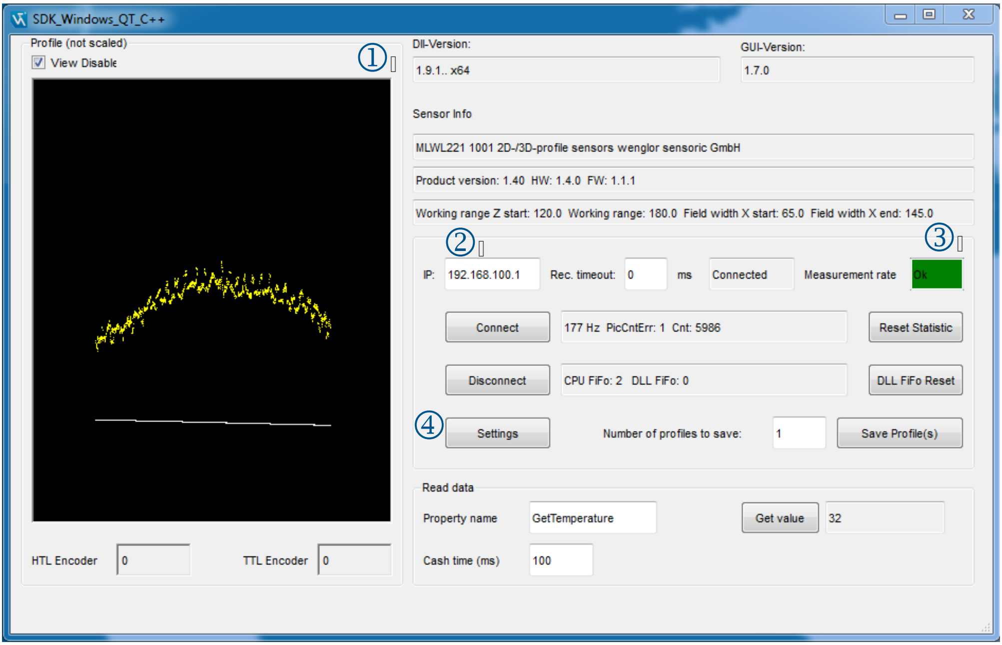


Fig. 81: Initial page of SDK demo project

① The demo project builds a connection to the profile sensor and shows a 2D representation of the scanned profile.
The white points in the representation show the scanned profile, while the yellow dots display the intensity (signal strength) of each point.

② To build up a connection with the sensor, the sensor IP must be entered in the IP field and the "Connect" button must be clicked.

③ The main window in the demo project shows also the state of the measurement rate.
If the measurement rate is within the allowed limits, the display field will show “Ok” (green background).
If the measurement rate is too fast, the display field will show “too fast” (red background).

④ Click on the button “Settings” in order to check the ROI settings and the corresponding max scan request value.
If the demo project fails to build a connection to the profile sensor, it will display the error message “EthernetScanner_Connect: Error in connection”.

NOTE!
Please check the IP address of your profile sensor and your network settings (see section 6.3.3).

NOTE!
You can check the connection state of the profile sensor through the web interface. Just type the IP address of the profile sensor in a web browser and look at the “Connected to” field on the right side of the web interface (see section 7).

The advance settings window (opens only if the connection to the profile sensor is established) allows to setup the profile sensor and read the values of basic properties. It allows sending raw ASCII commands as well, see Fig. 82).

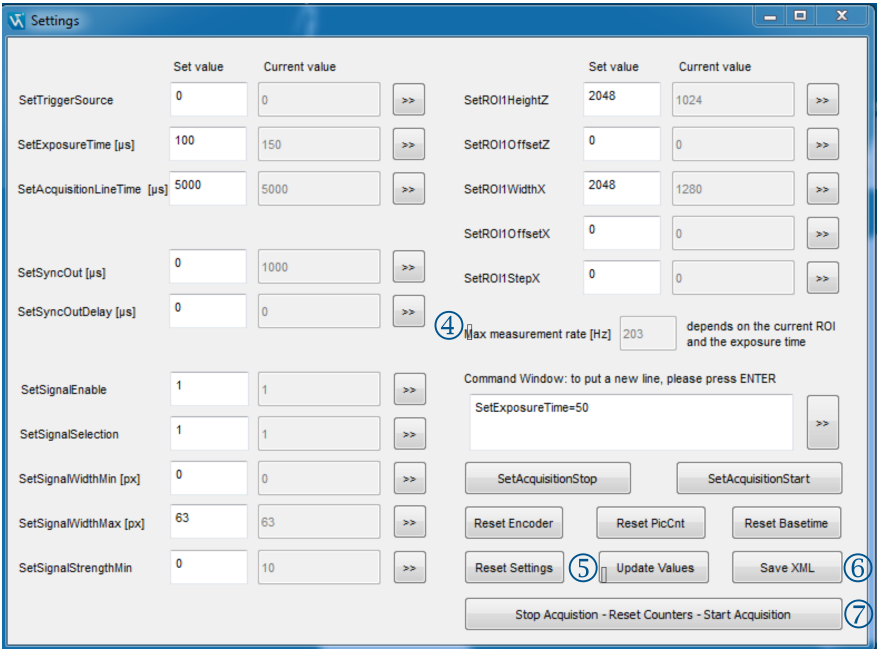


Fig. 82: Settings SDK demo project

④ The “Max measurement rate” field computes the maximum measurement rate for triggering the profile sensor from the current ROI settings. The equation for computing the maximum value is available in the source code or in section 7.2.

NOTE!
The computed max. measurement rate value is only an approximate value.

⑤ “Update Values” button updates the values of some basic properties by calling and parsing the XML data description from the profile sensor.

⑥ “Save XML” saves the XML descriptor as XML file.

⑦ “Stop Acquisition - Reset Counters - Start Acquisition” button shows an example of the best behaviour to reset the profile sensor counters (like picture counter and system time counter) after stopping the acquisition.

In order to get scanned profiles from the profile sensor in a reliable way, the host application should send the following commands in the given sequence to the profile sensor to build a connection:

1. Build a connection to the profile sensor (EthernetScanner_Connect).
2. Check the connection status (EthernetScanner_GetConnectStatus).
3. (Optional) Set up the profile sensor according to application needs through ASCII commands (EthernetScanner_WriteData).
4. (Optional) Read the property values from the profile sensor (EthernetScanner_ReadData).
5. Read the scanned profiles from the profile sensor (EthernetScanner_GetXZI) and process the data accordingly.
6. Disconnect from sensor before ending the application (EthernetScanner_Disconnect).

NOTE!
In DLL version 1.7.0 or higher, there is no need to send the ASCII command “SetInitializeAcquisition” to the profile sensor after each connection. The DLL sends this command internally. If your program sends this command, the DLL (1.7.0 or higher) will ignore it. Sending the command “SetInitializeAcquisition” from the DLL has brought a lot of performance improvements to the DLL.

NOTE!
In DLL version 1.9.0 or higher there is no need to make sure that the DLL is initialized through calling the function “EthernetScanner_GetInfo” (see "Obsolete Functions" in section 10.9). The function “EthernetScanner_GetConnectStatus” (step 2) will return ETHERNETSCANNER_TCPSCANNERCONNECTED after building a valid connection to the profile sensor AND initializing the DLL.

10.4 SDK Functions

All the SDK functions are based on C function standard calls (_stdcall) and are compatible with all compilers that support C programming language. In fact, since the functions are based on C standard call, they can be deployed in a wide range of "Integrated Development Environments" (IDEs) (QT, Visual studio C++, Visual Basic, C#, Delphi, Matlab, Labview, Embarcadero, etc.)

NOTE!
In DLL version 1.9.0 or higher all the SDK functions are thread safe.

NOTE!
All header definitions mentioned below are available in the header file "EthernetScannerSDKDefine.h" provided with the SDK.

10.4.1 Connecting weCat3D Profile Sensor

Command	void* EthernetScanner_Connect(char *chIP, char *chPort, int iTimeOut)
Parameter 1	char *chIP: IP address of the profile sensor: „192.168.100.1“ with \0 at the end
Parameter 2	char *chPort: port number of the profile sensor: „32001“ with \0 at the end
Parameter 3	int iTimeOut: Timeout in [ms] for the receive-function to close the connection, if no data is received. It is recommended to keep the timeout 0.
Response	void* a handle to the profile sensor. A NULL pointer is returned in case of failure
Description	This function will create a connection to the weCat3D sensor. The function will return a handle to the profile sensor, which will be used by other functions.

NOTE!
For checking the connection status with the profile sensor enter “EthernetScanner_GetConnectStatus”, see section 10.4.4.

10.4.2 UDP Connection

Command	void* EthernetScanner_ConnectUDP(char* chDestIP, char* chDestPort, char* chSrcIP, char* chSrcPort, char* chMode)
Parameter 1	char *chDestIP: IP address of the profile sensor: „192.168.100.1“ \0 terminated
Parameter 2	char *chDestPort: Portnumber of the profile sensor: „32001“ \0 terminated
Parameter 3	char *chSrcIP: IP address of the networks interface card to which the sensor is connected \0 terminated
Parameter 4	char *chSrcPort: Free Port which is used to receive the sensor data \0 terminated
Parameter 5	not used
Response	void* a handle to the profile sensor. A NULL pointer is returned in case of failure
Description	This function will create a UDP connection to the weCat3D sensor. The function will return a handle to the profile sensor, which will be used by ohter functions.

NOTE!
As UDP is a connectionless protocol a disconnect won't be detected by the SDK automatically. Furthermore it can't be guaranteed that data sent from the sensor won't get lost during transmission on high network loads.

10.4.3 Closing Connection

Command	void*_ stdcall EthernetScanner_Disconnect(void *pEthernetScanner)
Parameter 1	void*: the handle of the profile sensor, returned by the function "EthernetScanner_Connect", to be disconnected
Response	void*: a handle to the profile sensor. In case of a successful disconnect operation, the function will return a null pointer, else it will return the profile sensor handle itself.
Description	Close the connection between the DLL and the weCat3D sensor.

10.4.4 Check Connection

Command	void EthernetScanner_GetConnectStatus(void *pEthernetScanner, int *uiConnectStatus)
Parameter 1	void* : a handle to the profile sensor returned by the function "EthernetScanner_Connect"
Parameter 2	int * : a pointer to an integer variable, through which the connection status is returned.
Response	---
Description	This function checks the connection status to the profile sensor. The function is a non-blocking function. There are two states for the connection: • ETHERNETSCANNER_TCPSCANNERCONNECTED (3) meaning that the profile sensor is successfully connected to the DLL and the given IP and PORT in the function „EthernetScanner_Connect" are valid. In DLL version 1.9.0 or higher this state also means that the DLL is initialized (the DLL has received the linearization table from the profile sensor and all the internal parameters in the DLL have been set accordingly). From now on a valid profile can be polled using the function "EthernetScanner_GetXZIExtended". • ETHERNETSCANNER_TCPSCANNERDISCONNECTED (0) meaning that the profile sensor is disconnected or the given IP und PORT in the function "EthernetScanner_Connect" are not valid.

10.4.5 Get Measured Profile

Command	int EthernetScanner_GetXZIExtended(void *pEthernetScanner, double *pdoX, double *pdoZ, int *piIntensity, int *piSignalWidth, int iBuffer, unsigned int *puiEncoder, unsigned int *pucUSRIO, int dwTimeOut, unsigned char *ucBufferRaw, int iBufferRaw, int *iPicCnt)
Parameter 1	void *: a handle to the profile sensor returned by the function "EthernetScanner_Connect"
Parameter 2	double* : pointer to a raw buffer (Typ "double") used by the function to write in the X coordinates [in mm] of the measured profile. Pass NULL, if the buffer is not used.
Parameter 3	double*: pointer to a raw buffer (Typ "double") used by the function to write in the Z coordinates [in mm] of the measured profile. Pass NULL, if the buffer is not used.
Parameter 4	int* a pointer to a raw buffer (of type int) used by the function to write the intensity [10 bit] of the measured profile. Pass NULL, if this buffer is not used.

Parameter 5	int* a pointer to a raw buffer (Typ “int”) used by the function to write the peak width [in pixel < 32 pxl] . Pass NULL, if this buffer is not used.
Parameter 6	int: the length of the raw buffers passed in parameter 2 to 5. The length of the raw buffers should be larger than the number of measured points returned by the profile sensor. You can use the header definition ETHERNETSCANNER_BUFFERSIZEMAX provided in "EthernetScannerSDKDefine.h" to define the length of the raw buffers in the parameters 2 to 5.
Parameter 7	int*: a pointer to a variable (Typ “int”) which returns the encoder value of the current measured profile
Parameter 8	int*: a pointer to a variable (Typ “int”) which returns the IO status of the current measured profile. The IO status is decoded as follows: bit0: EA1 bit1: EA2 bit2: EA3 bit3: EA4 bit4: TTL Encoder A bit5: TTL Encoder B bit6: TTL Encoder C Pass NULL, if this value is not used
Parameter 9	The value of the blocking time to wait for a new measured profile, until the function times out. The value 0 makes the function non-blocking (timeout in ms).
Parameter 10	Deprecated. Pass NULL
Parameter 11	Deprecated. Pass NULL
Parameter 12	int* : a pointer to variable (Typ “int”) which returns the picture counter of current measured profile. This value is used to control the sequence of the received profiles.
Response	ETHERNETSCANNER_INVALIDHANDLE (-1000) if the sensor handle (parameter 1) is NULL or invalid. In the case of a success call, the function will return the total number of points of the measured profile written to the raw buffer (in parameter 2 to 5). The function will return ETHERNETSCANNER_GETXZINONEWSCAN (-1) if no new profile is available, ETHERNETSCANNER_GETXZIINVALIDBUFFER (-3) if the length of the buffer given in parameter 1 to 5 is shorter than the data to be written, ETHERNETSCANNER_GETXZIINVALIDLINDATA (-2) if the DLL is not initialized.
Description	The function calls up a profile from the internal FiFo in the DLL, if a new profile is available. The DLL saves all the measured profiles received from the profile sensor in an internal FiFo buffer. The programmer is responsible to pull the scanned profiles using this function as fast as possible to prevent overflow of the FiFo. If the program can not pull the scanned profiles fast enough, then it is recommended to decrease the output rate of the profile sensor. The function could be set to be blocking or non-blocking depending on the value of parameter 9. Set the function to be blocking (parameter 9 > 0) if you call the function from a secondary thread in your application. Set the function to be non blocking (parameter 9 = 0) if you call the function from the main thread in your application. To check the status of the FiFo, see section 10.4.8. To know how to set up the output rate of the profile sensor see the ASCII commands SetAcquisitionLineTime in in section 10.5.3.

10.4.6 Get Range Image

Command	int EthernetScanner_GetRangeImage(void* pEthernetScanner, unsigned short* imageBuffer, int iBuffer, int iTimeOutPerScan, bool* bFrameLost=nullptr , int* picCntBuffer=nullptr, int* encoderBuffer=nullptr, unsigned int* timeStampBuffer = nullptr);
Parameter 1	void *: a handle to the profile sensor returned by the function "EthernetScanner_Connect"
Parameter 2	unsigned short* : pointer to a raw buffer (Type "unsigned short") used by the function to write in the 16 Bit Grayscale Range Image.
Parameter 3	int: the length of the raw buffer passed in parameter 2. The length of the raw buffer should be larger than the number of measured points returned by the profile sensor. The number of elements of the buffer has to be at least sensors maximum xResolution* number of profiles per image.
Parameter 4	int: the value of the blocking time to wait for a new measured profile, until the function times out. The value 0 makes the function non-blocking (timeout in ms).
Parameter 5	bool*: (optional) pointer to a bool variable with (true) indicating a frame was lost during acquisition.
Parameter 6	int*: (optional) pointer to array of type int with the same size set by nrProfilesPerScan returning the picture count of each profile.
Parameter 7	int*: (optional) pointer to array of type int with the same size set by nrProfilesPerScan returning the encoder value of each profile.
Parameter 8	unsigned int*: (optional) pointer to array of type unsigned int with the same size set by nrProfilesPerScan returning the timestamp value of each profile
Response	ETHERNETSCANNER_INVALIDHANDLE (-1000) if the sensor handle (parameter 1) is NULL or invalid. In the case of a success call, the function will return ETHERNETSCANNER_OK (0). The function will return ETHERNETSCANNER_GETXZINONEWSCAN (-1) if no new profile is available, ETHERNETSCANNER_GETXZIINVALIDBUFFER (-3) if the length of the buffer is shorter than the data to be written, ETHERNETSCANNER_ GETXZIINVALIDLINDATA (-2) if the DLL is not initialized.
Description	With the weCat3D Range Image feature you can obtain multiple scans bundled as one ordered 16Bit 2D image, scaling the original z values as the images intensity. To parametrize use the corresponding WriteData functions in section 10.5.7.

10.4.7 Get Image

Command	int EthernetScanner_GetImage(void* pEthernetScanner, char *cBuffer, int iBuffer, unsigned int *puiWidth, unsigned int *puiHeight, unsigned int *puiOffsetX, unsigned int *puiOffsetZ, unsigned int *puiStepX, unsigned int *puiStepZ, unsigned int iTimeOut)
Parameter 1	void *: a handle to the profile sensor returned by the function "EthernetScanner_Connect"
Parameter 2	char *: pointer to a 8Bit raw buffer (e.g. unsigned char) used by the function to pass the pixelwise Intensities of the Camera Image. The Buffer should at least have the size of the nr of Pixels read e.g. RoiX_Width *RoiZ_Height.
Parameter 3	int: the length of the raw buffer passed in
Parameter 4	unsigned int*: pointer to variable of type unsigned int to read width of image in pixel
Parameter 5	unsigned int*: pointer to variable of type unsigned int to read height of image in pixel
Parameter 6	unsigned int*: pointer to variable of type unsigned int to read X offset in pixel
Parameter 7	unsigned int*: pointer to variable of type unsigned int to read Z offset in pixel
Parameter 8	unsigned int*: pointer to variable of type unsigned int to read subsampling in X
Parameter 9	unsigned int*: pointer to variable of type unsigned int to read subsampling in Z
Parameter 10	unsigned int: the value of the blocking time to wait for a new measured profile, until the function times out. The value 0 makes the function non-blocking (timeout in ms).
Response	ETHERNETSCANNER_INVALIDHANDLE (-1000) if the sensor handle (parameter 1) is NULL or invalid. In the case of a success call, the function will return the size of the data written to the raw buffer. The function will return ETHERNETSCANNER_GETXZINONEWSCAN (-1) if no new profile is available, ETHERNETSCANNER_GETXZIINVALIDBUFFER (-3) if the length of the buffer given in parameter 1 to 5 is shorter than the data to be written.
Description	The function is used to read out the camera image of the sensor in the currently defined ROI. Reading Camera Image is only possible in camera mode 1 (=camera images), see section 10.5.4.

10.4.8 Check DLL FiFo State

Command	int EthernetScanner_GetDllFiFoState(void *pEthernetScanner)
Parameter 1	void* : the handle to the profile sensor returned by the function "EthernetScanner_Connect"
Response	int: the status of the FiFo in the DLL in % (0 – 100) ETHERNETSCANNER_INVALIDHANDLE (-1000) if the sensor handle (parameter 1) is NULL or invalid.
Description	The function is used to check the status of the internal FiFo in the DLL to prevent the overflow and hence, to prevent the loss of unpolled scanned profiles.

10.4.9 Reset DLL FiFo

Command	int EthernetScanner_ResetDllFiFo(void *pEthernetScanner)
Parameter 1	void * the handle to the profile sensor returned by the function “EthernetScanner_Connect”
Response	The function returns ETHERNETSCANNER_OK (0) if the calling was successful. ETHERNETSCANNER_INVALIDHANDLE (-1000) if the sensor handle (parameter 1) is NULL or invalid.
Description	The function is used to reset the internal FiFo in the DLL. However, that could lead to the loss of unpolled scanned profiles. This function is useful, if the application can not poll the scanned profiles fast enough and the programmer wants to process the latest scanned profile. In that case, it is recommended to call this function just before calling the function “EthernetScanner_GetXZIExtended”.

10.4.10 Setup Profile Sensor

Command	int EthernetScanner_WriteData(void *pEthernetScanner, char *ucBuffer, int uiBuffer)
Parameter 1	void * the handle to the profile sensor returned by the function “EthernetScanner_Connect”
Parameter 2	char*: a pointer to a raw buffer (Typ “char”) which contains the ASCII command to be sent to the profile sensor
Parameter 3	int: the length of the raw buffer passed in parameter 2
Response	The function returns the number of bytes sent to the profile sensor. Normally, it should be the same length as the ASCII comand. ETHERNETSCANNER_INVALIDHANDLE (-1000) if the scanner handle (parameter 1) is NULL or invalid.
Description	The function is used to send ASCII commands to setup the profile sensor. The supported ASCII commands can be found in section 10.7.

10.4.11 Read DLL Version

Command	int EthernetScanner_GetVersion(unsigned char *ucBuffer, int uiBuffer)
Parameter 1	char*: a pointer to a raw buffer (Typ “char”) used by the function to write in the DLL version.
Parameter 2	int: the length of the raw buffer used in parameter 1. You can use a length of 1024 to create the buffer passed in parameter1.
Response	The function returns the total length (in bytes) of the written data in the raw buffer. If the length of DLL version to be written in the raw buffer is larger than the length of the raw buffer given in parameter 2, the function returns ETHERNETSCANNER_ERROR (-1).
Description	The function is used to check the current version of the DLL.

10.4.12 Read Property Values

Command	int EthernetScanner_ReadData(void *pEthernetScanner, char *chPropertyName, char *chRetBuf, int iRetBuf, int iCacheTime)
Parameter 1	void * the handle to the profile sensor returned by the function "EthernetScanner_Connect"
Parameter 2	char * buffer with the ASCII command (ending with char \0)
Parameter 3	char * return buffer for the result of the ASCII command
Parameter 4	int: the length of the return buffer. You can use the header definition ETHERNETSCANNER_BUFFERSIZEMAX provided in "EthernetScannerSDKDefine.h" to define the length of the raw buffer in parameter 3.
Parameter 5	int: the cache time in ms; the value in this parameter defines the function mode (XML mode or scan mode). See the description below for details.
Response	The function returns • ETHERNETSCANNER_READDATAOK (0) in case of success operation, • ETHERNETSCANNER_READDATASMALLBUFFER (-1) if the return buffer passed in parameter 3 is shorter than the length of the data available to be written in the buffer, • ETHERNETSCANNER_READDATANOTSUPPORTEDMODE (-2) in the case where the given ASCII command is not supported in the current read mode (like PictureCounter in XML mode), • ETHERNETSCANNER_READDATAFEATURENOTDEFINED (-3) if the ASCII command is not supported, • ETHERNETSCANNER_READDATANOSCAN (-4) if the function is called in scan mode and no profile is yet polled using the function „EthernetScanner_GetXZIExtended“, • ETHERNETSCANNER_READDATAFAILED (-5) if the function failed to read data from XML data or from profile data. • ETHERNETSCANNER_INVALIDHANDLE (-1000) if the sensor handle (parameter 1) is NULL or invalid.

Description	Starting from DLL version 1.9.0 or higher, the function “EhernetScanner_ReadData” is being introduced as a standard function in the SDK. The function reads the property values from the profile sensor. These values are cached in the DLL and the iCacheTime (parameter 5) defines how old the property value should be before writing it in the return buffer (parameter 3). The function and the supported ASCII command does not depend on specific firmware of the profile sensor. The function is implemented in the DLL as a comfort function to make it easy for the programmer to read property values from the profile sensor. There are two operating function modes: XML mode and scan mode: • XML mode is defined when the iCacheTime >=0. In this mode, the data are fetched from the XML descriptor received from the profile sensor and cached in internal structure in the DLL. If the data cache is older than the given iCacheTime value, the DLL will call a new XML file from the profile sensor, parse it and cache the data in the internal structure and then write the property value in the return buffer. **NOTE!** Setting a low value for iCacheTime in XML mode (d. h. iCacheTime = 0) will decrease the performance of the DLL since the DLL is then forced to read the full properties from the profile sensor and parse it each time the EthernetScanner_ReadData function is called. This would be evident if the DLL is working on low resource system or if the profile sensor works in range of kHz. • Scan mode is defined when the iCacheTime = -1. The DLL in this mode reads the property value from the data delivered with the current scan (pulled using the function „EthernetScanner_GetXZIExtended“). The property value in this mode will hold until the next successful call of the function „EthernetScanner_GetXZIExtended“. An example on how to use the new function can be found in the example code in the SDK. **NOTE!** Supported ASCII commands can be found in section 10.7. Not all properties are supported on both reading modes, see section 10.7 for more details.

10.5 Setup Profile Sensor

Below are the ASCII commands that are used to set up the profile sensor using the function "EthernetScanner_WriteData".

10.5.1 Initiate Reboot

Command	SetReboot\r
Description	Reboot the system

10.5.2 Exposure Time

10.5.2.1 Fixed Exposure Time

Command	SetExposureTime=x\r		
Parameter	Values of x: 0 … 100 000	**Default:**	150
Description	Exposure time is set in μs. If HDR mode is set (see section 10.5.6.1), SetExposureTime is the exposure time of the first profile. SetExposureTime2 is the exposure time of the second profile (see section 10.5.6.2).		

10.5.2.2 Auto Exposure Time

Command	SetAutoExposureMode=x\r		
Parameter	Values of x: 0: disabled 1: enabled	**Default:**	0
Description	Enables/disables the automatic control of the exposure time.		

NOTE!
Auto exposure time is available from firmware version 1.2.0 or higher.

Set the Minimum of Auto Exposure Time

Command	SetAutoExposureTimeMin=x\r		
Parameter	Values of x: 10...100 000	**Default:**	10
Description	Adjustment of the minimum exposure time in AutoExposureMode. The value is set in μs.		

Set the Maximum of Auto Exposure Time

Command	SetAutoExposureTimeMax=x\r		
Parameter	Values of x: 10...100 000	**Default:**	1000
Description	Adjustment of the maximum exposure time in AutoExposureMode. The value is set in μs.		

Set the Minimum of Intensity Range

Command	SetAutoExposureIntensityRangeMin=x\r		
Parameter	Values of x: 0...1024	**Default:**	450
Description	Sets the lower limit of the intensity range.		

Set the Maximum of Intensity Range

Command	SetAutoExposureIntensityRangeMax=x\r		
Parameter	Values of x: 0...1024	**Default:**	500
Description	Sets the upper limit of the intensity range.		

NOTE!
The intensity range should contain the area of the highest intensity. The exposure time is adjusted according to the average intensity of the selected range.

Set Minimum of Range X

Command	SetAutoExposureRangeXMin=x\r		
Parameter	Values of x: MLSL: 0...1279 MLWL: 0...2047	**Default:**	MLSL: 64 MLWL: 64
Description	Sets the starting point of range X.		

Set Maximum of Range X

Command	SetAutoExposureRangeXMax=x\r		
Parameter	Values of x: MLSL: 0...1279 MLWL: 0...2047	**Default:**	MLSL: 1215 MLWL: 1983
Description	Sets the ending point of range X.		

NOTE!
Range X defines the area where the control of the exposure time is applied.

10.5.3 Setup Acquisition Line Time

Command	SetAcquisitionLineTime=x\r		
Parameter	Values of x: 166 … 1 000 000	**Default:**	MLWL: 5714 MLSL: 5000
Description	Time between two consecutive profiles in µs. This function is only effective in internal trigger mode. 166 µs = 6000 Hz Explanation: MLWL: 5714 µs = 175 Hz MLSL: 5000 µs = 200 Hz		

The allowed values for AquisitionLineRate and ExposureTime depend on each other. The allowed value for AcquisitionLineRate and ExposureTime should hold for the following equation:

1000000 x (1 / AcquisitionLineRate) ≥ ExposureTime + 45 (µs)

NOTE!
It is necessary to reduce the ROI settings and the scan contents in the profile sensor to get a higher LineTimeRate (see SetROI1HeightZ, section 10.5.40.4, SetROI1WidthX, section 10.5.40.1, SetSignalContentWidth, section 10.5.12 and SetSignalContentReserved, section 10.5.13).

NOTE!
The profile sensor can transmit data through the network up to 30 MByte/s. Thus it is necessary to disable some signal contents to get a higher LineTimeRate (bis zu 6 kHz (166 µs) bei MLWL und 4 kHz (250 µs) bei MLSL).

10.5.4 Camera Mode

Command	SetCameraMode=x\r		
Parameter	Values of x: 0: Profile 1: Camera images	**Default:**	0
Description	If camera mode is selected the image of the camera is shown/transfered via the interface. The external GigE Vision interface, uniVision, VisionApp Demo 3D and VisionApp 360 do not support the camera mode. Thus it should be not selected if these programs are used.		

10.5.5 UDP Connection

Command	SetUDPSocketPort=x\r
Parameter	Values of x: 1024 ... 65000
Description	Input of the host port number to which the sensors sends the data (reserved 32001/32002).

Command	SetUDPSocketIP=x\r
Description	Input of the host IP address to which the sensors sends the data (format aaa.bbb.ccc.ddd).

Command	SetUDPSocketStart=x\r	
Parameter	Values of x: 0: End of UDP data transmission 1: Start UDP data transmission	
Description	Activating/deactivating UDP data transmission.	

Here a short example how to configure and activate UDP via commands:

```
SetAcquisitionStop
SetUDPSocketIP=192.168.100.181    // IP address of the host (IPC)
SetUDPSocketPort=32003            // Port used by the host (IPC)
SetUDPSocketStart=1
SetInitializeAcquisition
SetAcquisitionStart
```

10.5.6 HDR Mode

High Dynamic Range Imaging (HDR) is used to record objects with a very high intensity contrast. With firmware version 1.1.3 and higher HDR is implemented in the weCat3D sensors using the method of recording two profiles with different exposure times. The generation of the HDR profile based on the two profiles must be done by the user.

10.5.6.1 Set HDR

(available from FW version 1.1.3)

Command	SetHDR=x\r		
Parameter	Values of x: 0: HDR disabled 1: HDR enabled	**Default:**	0
Description	Enables/disables HDR mode.		

10.5.6.2 Setup ExposureTime2

(available from FW version 1.1.3)

Command	SetExposureTime2=x\r		
Parameter	Values of x: 0 … 100 000	**Default:**	150
Description	ExposureTime2 is set in μs. If HDR mode is set (see section 10.5.6.1), SetExposureTime2 is the exposure time of the second profile. SetExposureTime is the exposure time of the first profile (see section 10.5.2.1).		

10.5.7 Range Image

10.5.7.1 Set Number of Profiles

Command	SetRangeImageNrProfiles=x\r		
Parameter	Values of x: 1 ... 1000	**Default:**	1
Description	Sets the number of profiles per image implicitly defining the height of the image.		

10.5.7.2 Offsetting and Scaling (optional)

Command	SetRangeImageXScale=x\r		
Parameter	Values of x: Floating point value 0..1	**Default:**	xRangeMax/2
Description	Scale for mapping the x coordinates to the RangeImage pixels.		

Command	SetRangeImageXOffset=x\r		
Parameter	Values of x: - xRangeMax/2 ... xRangeMax/2	**Default:**	xRangeMax/xResolutionSensor
Description	Offset for mapping the x coordinates to the RangeImage pixels.		

Command	SetRangeImageZScale=x\r		
Parameter	Values of x: Floating point value 0..1	**Default:**	ZStart
Description	Scale mapping the z coordinates to the RangeImage intensity.		

Command	SetRangeImageZOffset=x\r		
Parameter	Values of x: 0 ... ZStart + ZRange	**Default:**	ZRange/65535
Description	Offset for mapping the z coordinates to the RangeImage pixels.		

NOTE!
To either modify the coordinate range which is pictured in the range image or recalculate 3D coordinates from the acquired range image an offset value and scale factor are applied both in X and Z, similar to the GigeVision coordinate scale and offset features. These values describe the mapping between 3D coordinates and 2D range image pixels according to:

$$X(i)_{Coord}=x(i)_{RangeImage}{}^{*}XScale+XOffset$$

$$Z(i)_{Coord}=I(i)_{RangeImage}{}^{*}ZScale+ZOffset$$

with X(i): pixelCoord of point i in 2D Image
and I(i): 16Bit Intensity value of point i in 2D Image

(please see Fig. 83 for visualization the range parameters)

*Scales and Offsets can be read out by their corresponding ReadData commands (see section 10.7).
By default ScaleFactor and offset are initialized so that the maximum measurement range in X and the Sensors Z range are displayed completely in the Range Image.

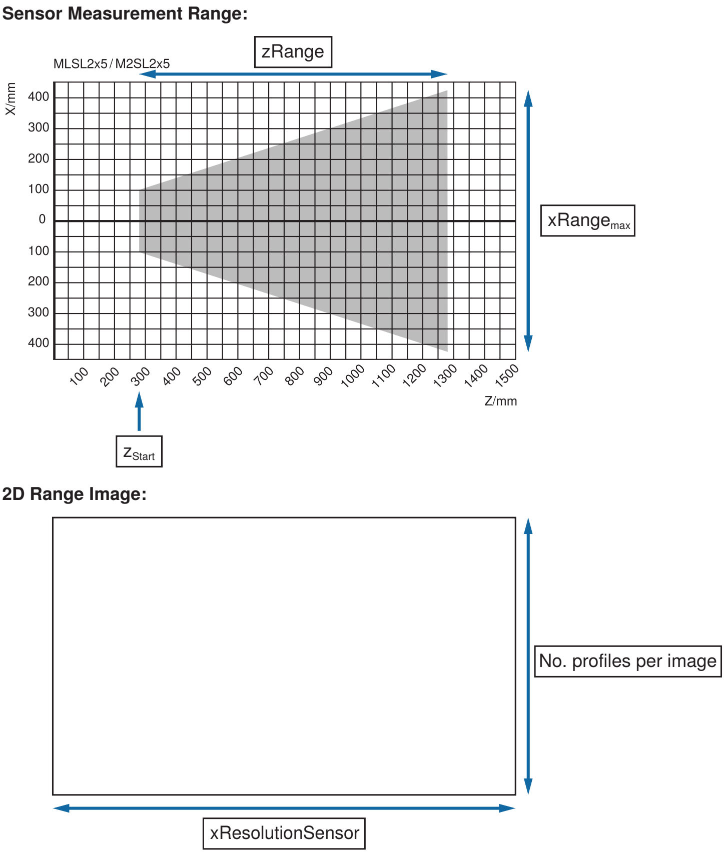


Fig. 83: Relation between sensor measurement range and 2D Range Image

10.5.8 Deactivate Laser

Command	SetLaserDeactivated=x\r		
Parameter	Values of x: 0: Laser on 1: Laser off	**Default:**	0
Description	Software command to control the laser as a global function. If this function is set to 1(enabled), then all other enabled signals on the E/A do not have any effects.		

10.5.9 Set User LED

Command	SetUserLED=x\r		
Parameter	Values of x: 0: off 1: red 2: green 3: orange	**Default:**	0
Description	The command controls the user LED for optical display of the application status directly at the weCat3D sensor.		

10.5.10 Enable Signal (Z)

Command	SetSignalContentZ=x\r		
Parameter	Values of x: 0: disabled 1: enabled	**Default:**	1
Description	By default, the data sent from the profile sensor contains Z (the depth), X (the width), I (the intensity / signal strength) and the peak width. This command will disable sending the Z signal value to save the bandwith of the network.		

10.5.11 Enable Signal (Strength)

Command	SetSignalContentStrength=x\r		
Parameter	Values of x: 0: disabled 1: enabled	**Default:**	1
Description	By default, the data sent from the profile sensor contains Z (the depth), X (the width), I (the intensity / signal strength) and the peak width. This command will disable sending the I signal value to save the bandwidth of the network.		

10.5.12 Enable Signal (Width)

Command	SetSignalContentWidth=x\r		
Parameter	Values of x: 0: disabled 1: enabled	**Default:**	1
Description	By default, the data sent from the profile sensor contains Z (the depth), X (the width), I (the intensity / signal strength) and the peak width. This command will disable sending the peak width value to save the bandwidth of the network.		

10.5.13 Reserved Data

Command	SetSignalContentReserved=x\r		
Parameter	Values of x: 0: disabled 1: enabled	**Default:**	1
Description	By default, the data sent from the profile sensor contains Z (the depth), X (the width), I (the intensity / signal strength), the peak width and the debug data. This command will disable sending the debug data to save the bandwidth of the network. This command only has an effect if the sensor is not linearized internally (SetLinearization-Mode = 0, see section 10.5.38).		

10.5.14 Setup Socket Connection Timeout

Command	SetSocketConnectionTimeout=x\r		
Parameter	Values of x: 0 ... 60 000ms	**Default:**	0
Description	Profile sensor Ethernet connection: rx-tx-timeout in ms. 0: Connection will not be closed, if no Ethernet data has been transferred (rx/tx).		

10.5.15 Setup Heartbeat Signal

Command	SetHeartBeat=x\r		
Parameter	Values of x: 0 ... 10000 ms	Default:	0
Description	The command will activate the heartbeat signal in the profile sensor. If the profile sensor does not send/receive any data, it sends every x ms a heartbeat signal (XML file). x = 0 deactivates the heartbeat signal.		

NOTE!
It is recommended to activate the heartbeat signal in the profile sensor, the heartbeat signal will enable the profile sensor to detect a physical (electrical) connection drop (like in the case where the network cable is unplugged). Thus, the profile sensor closes the connection to the host and allowing the host to build a new connection to the profile sensor. The recommended value is 1000 ms.

10.5.16 Initialize Acquisition

Command	SetInitializeAcquisition\r
Description	Sends matrix.bin to the host by request and enables sending data (profiles) used for SDK.

10.5.17 Start Acquisition

Command	SetAcquisitionStart\r
Description	After opening the socket connection this command is active and the profile data will be sent to the host (default).

10.5.18 Stop Acquisition

Command	SetAcquisitionStop\r
Description	The profile data will stop sending to the host.

NOTE!
Continue reading data from the sensor until no data arrives to be sure that no data remains in the FiFo's.

NOTE!
After using the ASCII command SetAcquistionStop do all settings or reset counters then run command SetAcquisitionStart. Please look to the commands SetAcquisitionStop and SetAcquisitionStart.

10.5.19 Reset Settings

Command	SetResetSettings\r
Description	Resets the sensor to the settings stored in Set0. The IP address of the profile sensor is retained.

NOTE!
A sleep time (1000 ms) should be added after executing the command SetResetSettings.

NOTE!
SetResetSettings command does not load the profile sensor factory settings. Only "Reset Sensor Settings" button in the profile sensor web interface resets the sensor settings to factory settings.

10.5.20 Reset Encoder

Command	SetResetEncoder\r
Description	Sets both encoder counters (HTL and TTL) to 0.

10.5.21 Reset Picture Counter

Command	SetResetPictureCounter\r
Description	Set the value of the picture counter to 0.

10.5.22 Reset Base Time Counter

Command	SetResetBaseTimeCounter\r
Description	Set the basetime counter of the sensor to 0.

10.5.23 Save/Load Settings

Command	SetSettingsSave=x\r SetSettingsLoad=x\r		
Parameter	Values of x: 0, 1, 2	**Default:**	0
Description	0: SetSettingsSave: Settings are used as default values at a restart . 0: SetSettingsLoad: 1: Set1 2: Set2 **NOTE!** If the settings are changed regularly, the same profile selection (see section 10.5.29) must be used in all settings.		

10.5.24 Setup Trigger Source

Command	SetTriggerSource=x\r		
Parameter	Values of x: -1: Sensor is in fix trigger mode (see section 7.2.3) If it is in dynamic trigger mode following settings are possible: 0: Internal trigger mode 1: Hardware trigger mode over SynIn function on E/A1...E/A4 2: Encoder trigger mode over HTL/TTL encoder 3: Software trigger mode	**Default:**	0
Description	The function is used to activate the trigger source of the profile sensor in dynamic trigger mode (see section 7.2.3 “Trigger Settings”) Set 0 for internal triggering. Set 1 to trigger the profile sensor through hardware signal (useful for synchronizing multiple profile sensors in an application). Set 2 to trigger the profile sensor through the encoder signal (if E/A 1 and E/A 2 are defined as encoder function, the E/A encoder will be used as the trigger source, otherwise the TTL-RS422 is used). Set 3 to trigger the signal through the software command “SetTriggerSoftware”. The function returns “-1” if the dynamic trigger mode in the profile sensor is switched off (the profile sensor is working in fixed trigger mode). The value “-1” can not be used in this function. If you want to switch off the dynamic trigger mode, please use the command SetTriggerAmountProfilesY.		

NOTE!
If the trigger source in the profile sensor is setup to encoder, hardware or software, and the profile sensor did not receive a trigger signal within the time defined in the iTimeOut input parameter (parameter 3) in "EthernetScanner_Connect"; the DLL will close the connection to the profile sensor and will build a new connection to it. To avoid this behaviour, you have either to set the iTimeOut value in "EthernetScanner_Connect" to 0 (see section 10.4.1) or setup the heartbeat signal to, for example, 1000 (see section 10.5.15).

10.5.25 Setup Trigger Divider

Command	SetTriggerEncoderStep=x\r		
Parameter	Values of x: 0 ... 65535	**Default:**	0
Description	Set a trigger divider for both hardware trigger source (SyncIn input) and encoder trigger source (Encoder HTL or TTL). The profile sensor will be triggered at the x+1 signal. This property is useful, if we have a high frequency external trigger source (either Encoder or SyncIn signal). The maximum input frequency of the trigger signal is 1 MHz.		

10.5.26 Setup Trigger Delay

Command	SetTriggerDelay=x\r		
Parameter	Values of x: 0 … 1 000 000	**Default:**	0
Description	Trigger delay is usually used in the sub unit in multi-sensor setup. Trigger delay is set in μs.		

NOTE!
Trigger delay + exposure time in sub unit should be smaller than the AcqusitionLineTime in the master unit.

10.5.27 Software Trigger Command

Command	SetTriggerSoftware\r
Parameter	---
Description	Triggers the profile sensor to scan a profile over a software command. The profile sensor should be in software trigger mode.

10.5.28 Setup Encoder Trigger Function

Command	SetEncoderTriggerFunction=x\r		
Parameter	Values of x: 0: DirectionUp 1: DirectionDown 2: Motion 3: PositionUp 4: PositionDown	**Default:**	2
Description	DirectionUp: The encoder will trigger the profile sensor only in one direction (counting up) DirectionDown: The encoder will trigger the profile sensor only in one direction (counting down). Motion: The encoder will trigger the profile sensor in both directions (counting up and down) PositionUp: The encoder will trigger the profile sensor in one direction (couting up) and only if the encoder position (counter value) is larger than the latest position. PositonDown: The encoder will trigger the profile sensor in one direction (couting down) and only if the encoder position (counter value) is smaller than the latest position.		

10.5.29 Enable Fixed Frame mode

Command	SetTriggerAmountProfilesY=x\r		
Parameter	Values of x: -1: Sensor is in dynamic trigger mode (see section 7.2.3) If sensor is in fix trigger mode following settings are possible: 0: Internal trigger mode 1: Hardware (SyncIn) trigger mode 2: Encoder trigger mode 3: Software trigger mode	**Default:**	-1
Description	This command is used to activate the fixed trigger mode in the profile sensor. In fixed trigger mode the profile sensor sends a certain number of profiles which are defined by the user (see section 10.5.30) to the host and then stops until the profile sensor receives a new SetAcquisitionStart command or hardware signal on ProfileEnabel pin (if defined). -1 means that the fixed frame mode in the profile sensor is switched off (the profile sensor is working in dynamic trigger mode). The value -1 can not be used in this function. If you want to switch off the fixed frame mode, please use the command SetTriggerSource=x.		

NOTE!
If the trigger source in the profile sensor is setup to encoder, hardware or software, and the profile sensor did not receive a trigger signal within the time defined in the iTimeOut input parameter (parameter 3) in "EthernetScanner_Connect"; the DLL will close the connection to the profile sensor and will build a new connection to it. To avoid this behaviour, you have either to set the iTimeOut value in "EthernetScanner_Connect" to 0 (see section 10.4.1) or setup the heartbeat signal to, for example, 1000 (see section 10.5.15).

10.5.30 Setup Number of Profiles in Fixed Frame Mode

Command	SetAmountProfilesY=x\r		
Parameter	Values of x: 0...10 000	**Default:**	0
Description	The command sets up the number of profiles to be sent to the host in the fixed frame mode (see section 10.5.29). Profiles are output only if x > 0.		

10.5.31 Setup Sync Out

Command	SetSyncOut=x\r		
Parameter	Values of x: 10...100 000	**Default:**	1000
Description	Defines the signal width (duration in μs) of the SyncOut signal (high) for the E/A SyncOut. The value of SyncOut signal width and the SyncOutDelay time (see section 10.5.32) combined should be less than the AcquisitionLineTime value (see section 10.5.3). This is important in order to prevent having one long SyncOut signal during the acquisition mode.		

NOTE!
The signal width must be at least half the period of the measuring rate.

10.5.32 Setup Delay of Sync Out

Command	SetSyncOutDelay=x\r		
Parameter	Values of x: 0...100 000	**Default:**	0
Description	Defines the value of (switching) delay (in μs) of the SyncOut trigger signal (high) for the E/A SyncOut.		

10.5.33 Profile Selection

Command	SetSignalEnable=x\r		
Parameter	Values of x: 1: Profile 1 2: Profile 2 3: Profile 1 + Profile 2	**Default:**	1
Description	The command sets the number of profiles which are sent with each scan (see section 10.5.37).		

10.5.34 Setup Minimum Peak Width

Command	SetSignalWidthMin=x\r		
Parameter	Values of x: 0...63	**Default:**	0
Description	Peak width filter: This function is a filter to define the minimum peak width in pixels. Usual values: 2 or 3		

10.5.35 Setup Maximum Peak Width

Command	SetSignalWidthMax=x\r		
Parameter	Values of x: 0...63	**Default:**	63
Description	Peak width filter: This function is a filter to define the maximum peak width in pixels. Usual values: 12		

10.5.36 Setup Minimum Signal Strength

Command	SetSignalStrengthMin=x\r		
Parameter	Values of x: 0...1023	**Default:**	0
Description	Defines the minimum signal strength for signal evaluation.		

10.5.37 Signal Selection

Command	SetSignalSelection=x\r		
Parameter	Values of x: 0: Peak 1 1: Intensity 2: Width 3: Peak 2	**Default:**	1
Description	Defines the peak which is to be used for the profile output. The sensor acquires internally two peaks. Based on this selection the sensor provides the corrensponding profile.		

10.5.38 Internal Profile Calculation

Command	SetLinearizationMode=x\r		
Parameter	Values of x: 0: disabled 1: enabled	**Default:**	0
Description	The weCat3D profile sensors have the possibility to calculate the profile internally or externally using the SDK of the weCat3D sensor. If the profile is calculated internally, the calculated profiles are submitted via a TCP/IP protocol. If set to 1 the internal calculation is enabled. Before switching between internal or external calculation it must be ensured that no data are still transmitted. The program flow is: SetAcquisitionStop\r //wait until no data are received by host SetInitializeAcquisition\r SetLinearizationMode=1\r SetAcquisitionStart\r		

NOTE!
The activation of the internal profile calculation decreases the CPU load on the host.

NOTE!
SetLinearizationMode command is available in firmware version 1.2.0 or higher.

10.5.39 Setup Encoder Count Direction

Command	SetEncoderCountDirection=x\r		
Parameter	Values of x: 0: normal 1: inverted	**Default:**	0
Description	The count direction of the encoder values can be inverted.		

10.5.40 Region of Interest (ROI)

10.5.40.1 Setup ROI Width in X

Command	SetROI1WidthX=x\r		
Parameter	Values of x: MLSL: 32...1280 MLWL: 32...2048	**Default:**	MLSL: 1280 MLWL: 2048
Description	Amount of camera rows to readout: MLWL: no effect on the measurement rate, effect on the ethernet bandwidth MLSL: in steps of 16, effect on the measurement rate, effect on the ethernet bandwidth (see section 7.2.3).		

10.5.40.2 Setup ROI Offset in X

Command	SetROI1OffsetX=x\r		
Parameter	Values of x: MLSL: 0...1279 MLWL: 0...2047	**Default:**	0
Description	MLWL: in steps of 1 MLSL: in steps of 32 Defines the offset of the ROI in X-direction in relation to the first line.		

10.5.40.3 Setup Subsampling in X

Command	SetROI1StepX=x\r		
Parameter	Values of x: 0: disabled 1: MLSL subsampling enabled, MLWL only step 1 2 ... x: only steps	**Default:**	0
Description	**MLSL:** If amount of pixel in the CMOS line (width X) set to half then the range of X looks like full. The measuring rate can be increased by double. **MLWL:** Decreases only the amount of data, has no effect to the measuring rate.		

10.5.40.4 Setup ROI Height in Z

Command	SetROI1HeightZ=x\r		
Parameter	Values of x: MLSL: 32...1024 MLWL: 32...2048	**Default:**	MLSL: 1024 MLWL: 2048
Description	Amount of camera lines to readout has an effect on the Ethernet bandwidth and the measurement rate.		

10.5.40.5 Setup ROI Offset in Z

Command	SetROI1OffsetZ=x\r		
Parameter	Values of x: MLSL: 0...1023 MLWL: 0...2047	**Default:**	0
Description	Defines the offset of the ROI in Z-direction in relation to the first line.		

10.5.40.6 Setup Subsampling in Z

Command	SetROI1StepZ=x\r		
Parameter	Values of x: 0: disabled 1: enabled	Default:	0
Description	If activated, every second line of the camera chip is read out so that the measuring rate can be doubled. This can worsen the resolution.		

10.5.40.7 Setting ROI in X/Z

Command	SetROI1_mm=x1,z1,x2,z2\r
Parameter	Values of x: x1: X coordinate of left top corner of the ROI in mm z1: Z coordinate of left top corner of the ROI in mm x2: X coordinate of right bottom corner of the ROI in mm z2: Z coordinate of right bottom corner of the ROI in mm
Description	Defines a region of interest (ROI) in mm. Because the measuring field has a trapezoidal shape but the shape of the ROI is rectangular please compare the sketch how the resulting ROI is defined (see Fig. 84).

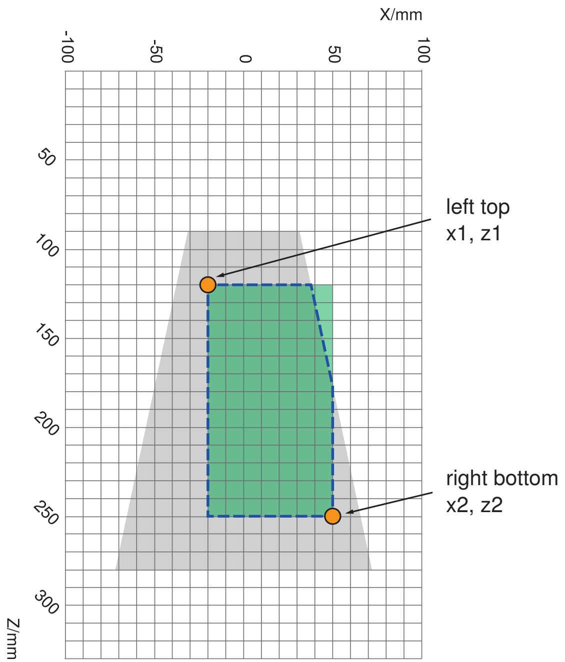


Fig. 84: Measuring field and ROI (X, Z)

Grey area: Measuring field
Green area: ROI (X, Z)
Area within dashed line: Resulting ROI

10.5.40.8 Setting ROI in Z in mm

Command	SetROI1Z_mm=z1,z2\r
Parameter	Values of x: z1: Start of the ROI in Z in mm z2: End of the ROI in Z in mm
Description	Defines a region of interest (ROI) in mm. Please compare the sketch how the resulting ROI is defined (see Fig. 85).

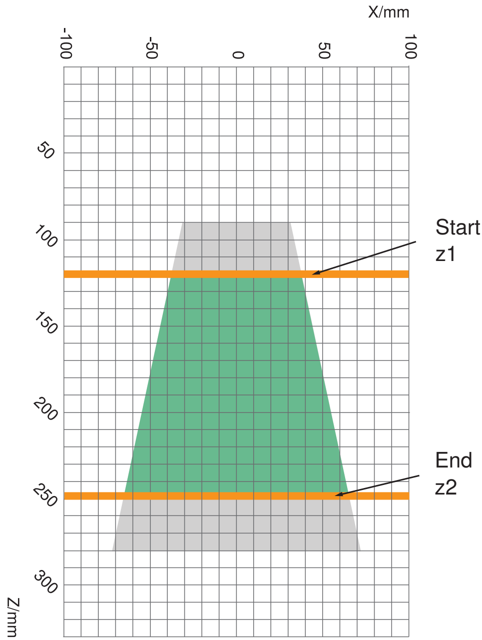


Fig. 85: Measuring field and ROI (Z)

Grey area: Measuring field
Green area: ROI (Z)

10.5.41 E/A Functions

10.5.41.1 Setup E/A Functions

The profile sensor offers 4 separate E/A functions. The following commands relate to these E/A functions and can be used for all E/As. The encoder HTL functions are only available for E/A 1 and E/A 2. The following explanation uses the syntax to set up the E/A 1. For addressing E/A 2 to E/A 4 use the same syntax:

Beispiel:

SetEA1Function=1
SetEA2Function=2
SetEA3FunctionLaserOff=0

Command	SetEA1Function=x\r		
Parameter	Values of x: 1: sync_in 2: sync_out 3: input 4: output 5: encoder_ab	**Default:**	5
Description	Encoder_A/B (E/A 1+ E/A 2): Input function for connecting an HTL (5 to 24 V, A/B channel) rotary encoder. This function must be set for E/A 1 and E/A 2 at the same time. This function is only available for E/A 1 and E/A 2. If the encoder function is enabled on E/A 1/2, then the encoder value in the GetXZI function will be provided from this encoder! If no E/A 1/2 encoder function is selected, then the encoder value in the GetXZI function will be provided from TTL-RS422.		

10.5.41.2 Laser Off

Command	SetEA1FunctionLaserOff=x\r		
Parameter	Values of x: 0: disabled 1: enabled	**Default:**	0
Description	E/A high state: laser is off E/A low state: laser is on The E/A should be set to input (see section 10.5.41.1) for this function to work.		

10.5.41.3 Profile Enable

Command	SetEA1FunctionProfileEnable=x\r		
Parameter	Values of x: 0: disabled 1: enabled	**Default:**	0
Description	E/A high state: profiles will be send to the host The E/A should be set to input (see section 10.5.41.1) for this function to work.		

10.5.41.4 Reset Counter

Command	SetEA1FunctionResetCounter=x\r
Parameter	Values of x: 0: disabled 1: enabled
Description	Enables the E/A pin to reset one or more counters in the profile sensor (see example SetEA1ResetCounterEncoderHTL oder SetEA1ResetCounterBaseTime). The E/A should be set to input (see section 10.5.41.1) for a working function.

10.5.41.5 Reset Counter Repeat

Command	SetEA1ResetCounterRepeat=x\r
Parameter	Values of x: 0: disabled 1: once 2: always
Description	If the function is disabled, the E/A will not reset any counter. If it is “1” the E/A will reset the counter only once when the E/A is active. If you need to reset the counter again, the command should be sent again to the profile sensor. “2” means that the reset counter E/A will always reset the counter each time the E/A is active.

10.5.41.6 Signal Edge for Counter Reset

Command	SetEA1ResetCounterSignaledge=x\r
Parameter	Values of x: 0: rising and falling edge 1: rising edge 2: falling edge
Description	Defines the edge of the signal to reset the counter. The E/A should be defined as an input, reset counter and reset counter repeat should be active (see section 10.5.41.1, 10.5.41.4 and 10.5.41.5).

10.5.41.7 Reset Base Time Counter

Command	SetEA1ResetCounterBaseTimeCounter=x\r
Parameter	Values of x: 0: disabled 1: enabled
Description	Enables the E/A to reset the basetime counter in the sensor. The E/A should be defined as an input, reset counter and reset counter repeat should be active (see section 10.5.41.1, 10.5.41.4 and 10.5.41.5).

10.5.41.8 Reset Picture Counter

Command	SetEA1ResetCounterPictureCounter=x\r
Parameter	Values of x: 0: disabled 1: enabled
Description	Enables the E/A to reset the picture counter in the sensor. The E/A should be defined as an input, reset counter and reset counter repeat should be active (see section 10.5.41.1, 10.5.41.4 and 10.5.41.5).

10.5.41.9 Reset Encoder HTL

Command	SetEA1ResetCounterEncoderHTL=x\r
Parameter	Values of x: 0: disabled 1: enabled
Description	Enables the E/A to reset the HTL encoder counter in the profile sensor. The E/A should be defined as an input, reset counter and reset counter repeat should be active (see section 10.5.41.1, 10.5.40.4 and 10.5.41.5).

10.5.41.10 Reset Encoder TTL

Command	SetEA1ResetCounterEncoderTTLRS422=x\r
Parameter	Values of x: 0: disabled 1: enabled
Description	Enables the E/A to reset the TTL encoder counter in the profile sensor. The E/A should be defined as an input, reset counter and reset counter repeat should be active (see section 10.5.41.1, 10.5.41.4 and 10.5.40.5).

Example 1:
Setting E/A 3 to reset HTL encoder and TTL encoder each time it receives a high signal:

SetEA3Function=3\r
SetEA3FunctionResetCounter=1\r
SetEA3ResetCounterRepeat=2\r
SetEA3ResetCounterSignaledge=2\r
SetEA3ResetCounterEncoderHTL=1\r
SetEA3ResetCounterEncoderTTLRS422=1\r

10.5.41.11 Setup E/A 1 Input Function

Command	SetEA1InputFunction=x\r		
Parameter	Values of x: 0: Ub inactive 1: Ub active	**Default:**	1
Description	The input signal can be inverted as a function.		

10.5.41.12 Setup Input Load

Command	SetEA1InputLoad=x\r		
Parameter	Values of x: 0: input load disabled 0 mA 1: input load enabled 2 mA	**Default:**	0
Description	Enable/disable the extra load on the E/A input to get 0 level defined (Helpful for some PLC hardware).		

10.5.41.13 Setup Output

Command	SetEA1Output=x\r		
Parameter	Values of x: 1: Push-Pull 2: PNP 3: NPN	**Default:**	1
Description	Determines the output mode for the E/A (Push-Pull, PNP or NPN).		

10.5.41.14 Setup Output Function

Command	SetEA1OutputFunction=x\r		
Parameter	Values of x: 0: NO 1: NC	**Default:**	0
Description	0: NO (normally open) 1: NC (normally close)		

10.5.41.15 Activate Input Counter

Command	SetEA1FunctionInputCounter=x\r		
Parameter	Values of x: 0: disable 1: enable	**Default:**	0
Description	Enables/disables the user counter function at the E/A. Use the ASCII command GetEAFunctionInputCounter to read the counter value, see section 10.7.		

NOTE!
The E/A should be set to input (SyncIn or UserInput) for the counter to work.

10.5.42 Setup User Data

Command	SetStatisticDataUserData=x\r		
Parameter	Values of x: 0...65535		
Description	This command helps the user to synchronize the communication between the host and the profile sensor. The command writes a user defined data into internal register (2 bytes) in the profile sensor. The user can read back the value using the command GetStatisticDataUserData in Function EhternetScanner_ReadData in scan mode, see section 10.4.12.		

NOTE!
Available in firmware version 1.2.0 or higher and DLL version 1.10.0 or higher.

10.6 Adjustments of the Library

10.6.1 Setup the FiFo Size

Command	SetLibraryScannerFiFoSize=x\r		
Parameter	Values of x: 4198400 ... 4294967295 (in bytes)	**Default:**	41984000
Description	This command is used to setup the shared library internal FiFo size in bytes. Call this command before calling the function EthernetScanner_Connect. Example: EthernetScanner_WriteData(0,"SetLibraryScannerFiFoSize=4198400", strlen("SetLibraryScannerFiFoSize=4198400"));		

NOTE!
This command is implemented in the DLL internally and not supported by the FW in the profile sensor.
The command must not be sent with another command via EthernetScanner_WriteData.

10.6.2 Setup the FiFo Mode

Command	SetLibraryScannerFiFoMode=x\r		
Parameter	Values of x: 0, 1	**Default:**	1
Description	x=0 deactivates the internal DLL FiFo buffer and the function EthernetScanner_GetXZIExtended delivers in this mode the latest available received profile (ignoring all other older profiles). x=1 activates the internal DLL FiFo buffer functionality.		

NOTE!
This command is implemented in the DLL internally and not supported by the FW in the profile sensor.

10.6.3 Automatic Connection Establishment between DLL and Sensor

Command	SetAutoConnectMode=x\r		
Parameter	Values of x: 0, 1	**Default:**	1
Description	x=0: Automatic connection is switched off x=1: Automatic connection is switched on. When switched on, the DLL tries to reestablish an interrupted connection between the DLL and the sensor. While the connection is interrupted, the status parameter in EthernetScanner_GetConnect-Status provides the function ETHERNETSCANNER_TCPSCANNERDISCONNECT (0).		

10.7 Read Properties of weCat3D Profile Sensor

The following table shows the current ASCII commands that can be used to read the properties of the profile sensor using the function EthernetScanner_ReadData.
The table shows also the availability of each command for each read mode.
See the demo project in the SDK for a code example.

ASCII Command	XML mode	Scan mode	Remarks
GetPictureCounter	o	x	
GetTimestamp (former: GetSystem-Time)	o	x	in μs (GetSystemTime is deprecat-ed.It is recommended to use from now on the new command GetTime-stamp).
GetStatisticDataUserData	o	x	
GetOrderNumber	x	o	
GetProductVersion	x	o	
GetProducder	x	o	
GetFirmwareVersion	x	o	
GetSerialNumber	x	o	
GetMAC	x	o	
GetWorkingRangeZStart	x	o	
GetWorkingRangeZEnd	x	o	
GetFieldWidthXStart	x	o	
GetFieldWidthXEnd	x	o	
GetPixelXMax	x	o	
GetPixelZMax	x	o	
GetOnOffCounter	x	o	
GetOnTimeCounter	x	o	
GetLinInfo	x	o	if the sensor is calibrated
GetUserString	x	o	
GetHeartBeat	x	o	
GetSocketConnectionTimeout	x	o	

ASCII Command	XML mode	Scan mode	Remarks
GetIOState	x	x	bit0: E/A 1 bit1: E/A 2 bit2: E/A 3 bit3: E/A 4
GetEncoderHTL	x	x	Value is output as unsigned int and must be converted to the signed int data type by the customer.
GetEncoderTTL	x	x	Value is output as unsigned int and must be converted to the signed int data type by the customer.
GetEncoder	o	x	Returns the value of the active encoder. The active encoder could be the HTL or the TTL encoder depending on the E/A setup
GetTemperature	x	x	Value is output as unsigned int and must be converted to the signed int data type by the customer.
GetScannerState	x	x	bit0: Ready (0=NOK; 1=OK) bit1: ExposureTime (0=NOK; 1=OK) bit2: LaserONTime (0=NOK; 1=OK) bit3: Not in use bit4: Not in use bit5: Measurement rate too high (0=NOK; 1=too high) bit6: Not in use bit7: Not in use
GetSignalEnable	x	x	The number of data per profile, see function SetSignalEnable
GetSignalContentZ	x	x	
GetSignalContentStrength	x	x	
GetSignalContentWidth	x	x	
GetSignalContentReserved	x	x	
GetSignalWidthMin	x	x	
GetSignalWidthMax	x	x	
GetSignalStrengthMin	x	x	
GetSignalSelection	x	x	
GetAcquisitionLineTime	x	x	
GetCameraRunning	x	x	
GetTriggerSource	x	x	
GetTriggerAmountProfilesY	x	x	
GetAmountProfilesY	x	x	
GetTriggerEncoderStep	x	x	
GetTriggerDelay	x	x	

ASCII Command	XML mode	Scan mode	Remarks
GetExposureTime	x	x	In Scan mode the current exposure time is shown also if auto exposure time is activated.
GetLaserActive	x	x	
GetROI1WidthX	x	x	
GetROI1OffsetX	x	x	
GetROI1StepX	x	x	
GetROI1HeightZ	x	x	
GetROI1OffsetZ	x	x	
GetROI1_mm	x	o	
GetROI1Z_mm	x	o	
GetSyncOut	x	x	
GetRangeImageNrProfiles	x	x	
GetRangeImageXScale	x	x	
GetRangeImageXOffset	x	x	
GetRangeImageZScale	x	x	
GetRangeImageZOffset	x	x	
GetOperatingMode	x	o	
GetSyncOutDelay	x	x	
GetEncoderTriggerFunction	x	x	
GetEncoderCountDirection	x	x	
GetEA1Function	x	x	
GetEA1FunctionLaserOff	x	x	
GetEA1FunctionProfileEnable	x	x	
GetEA1FunctionResetCounter	x	x	
GetEA1InputFunction	x	x	
GetEA1InputLoad	x	x	
GetEA1Output	x	x	
GetEA1OutputFunction	x	x	
GetEA1ResetCounterRepeat	x	x	
GetEA1ResetCounterSignaledge	x	x	
GetEA1ResetCounterBaseTimeCounter	x	x	
GetEA1ResetCounterPictureCounter	x	x	
GetEA1ResetCounterEncoderHTL	x	x	
GetEA1ResetCounterEncoderTTLRS422	x	x	
GetEA2Function	x	x	
GetEA2FunctionLaserOff	x	x	
GetEA2FunctionProfileEnable	x	x	
GetEA2FunctionResetCounter	x	x	
GetEA2InputFunction	x	x	

ASCII Command	XML mode	Scan mode	Remarks
GetEA2InputLoad	x	x	
GetEA2Output	x	x	
GetEA2OutputFunction	x	x	
GetEA2ResetCounterRepeat	x	x	
GetEA2ResetCounterSignaledge	x	x	
GetEA2ResetCounterBaseTimeCoun-ter	x	x	
GetEA2ResetCounterPictureCounter	x	x	
GetEA2ResetCounterEncoderHTL	x	x	
GetEA2ResetCounterEnco-derTTLRS422	x	x	
GetEA3Function	x	x	
GetEA3FunctionLaserOff	x	x	
GetEA3FunctionProfileEnable	x	x	
GetEA3FunctionResetCounter	x	x	
GetEA3InputFunction	x	x	
GetEA3InputLoad	x	x	
GetEA3Output	x	x	
GetEA3OutputFunction	x	x	
GetEA3ResetCounterRepeat	x	x	
GetEA3ResetCounterSignaledge	x	x	
GetEA3ResetCounterBaseTimeCoun-ter	x	x	
GetEA3ResetCounterPictureCounter	x	x	
GetEA3ResetCounterEncoderHTL	x	x	
GetEA3ResetCounterEnco-derTTLRS422	x	x	
GetEA4Function	x	x	
GetEA4FunctionLaserOff	x	x	
GetEA4FunctionProfileEnable	x	x	
GetEA4FunctionResetCounter	x	x	
GetEA4InputFunction	x	x	
GetEA4InputLoad	x	x	
GetEA4Output	x	x	
GetEA4OutputFunction	x	x	
GetEA4ResetCounterRepeat	x	x	
GetEA4ResetCounterSignaledge	x	x	
GetEA4ResetCounterBaseTimeCoun-ter	x	x	
GetEA4ResetCounterPictureCounter	x	x	
GetEA4ResetCounterEncoderHTL	x	x	
GetEA4ResetCounterEnco-derTTLRS422	x	x	

ASCII Command	XML mode	Scan mode	Remarks
GetEAFunctionInputCounter	o	x	
GetSettings=0	x	x	Returms the saved settings of the profile sensor in default as xml structure
GetSettings=1	x	x	Returns the saved settings of the profile sensor in set1 as xml structure
GetSettings=2	x	x	Returns the saved settings of the profile sensor in set2 as xml structure
GetSettings=3	x	x	Returns the current settings of the profile sensor as xml structure
GetCheckLinearizationMode	x	x	Returns "1" if the sensor supports profile linearization internally, "0" if the sensor does not. See SetLin-earizationMode command for more details.

(x) available; (o) not available

10.8 Data Structure

10.8.1 General

The profile information queried by the GetXZIExtended function are displayed separately as buffer for each value (X,Z,I). If the measured object is located outside the measuring range, the measured value is set to 0.

10.8.1.1 Buffer Struktur (one selected signal)

In case of just one selected signal (signal selection) the buffer structure appears in this order:

Buffer	X	Buffer	Z	Buffer	I	Buffer	Peakbreite	
0	double	0	double	0	int	0	int	1st point
1	double	1	double	1	int	1	int	2nd point
2	double	2	double	2	int	2	int	3rd point
... *		...		...		...		

*to...1280 MLSL / to...2048 MLWL

10.8.1.2 Buffer Structure (two selected signals)

If the signal selection is set up to get signal 1 and signal 2, the buffer contains the data in the following, different order:

Buffer	X	Buffer	Z	Buffer	I	Buffer	Peakwidth	
0	double	0	double	0	int	0	int	1st point 1st signal
1	double	1	double	1	int	1	int	1st point 2nd signal
2	double	2	double	2	int	2	int	2nd point 1st signal
3	double	3	double	3	int	3	int	2nd point 2nd signal
... *		...		...		...		

* to ...2560 MLSL / to...4096 MLWL

10.9 Obsolete Functions

10.9.1 Get General Sensor Information

Befehl	int EthernetScanner_GetInfo(void *pEthernetScanner, char *chInfo, int iBuffer, char *ch-Mode)
Parameter 1	void* : a handle to the profile sensor returned by the function “EthernetScanner_Connect”
Parameter 2	char*: a pointer to a raw buffer (Typ “char”), where the profile sensor information will be written.
Parameter 3	int: the length of the raw buffer. The programmer should make sure that the length of the raw buffer is larger than the length of the returned sensor information. You can use the header definition ETHERNETSCANNER_GETINFOSIZEMAX provided in "EthernetScan-nerSDKDefine.h" to define the length of the raw buffer in parameter 2.
Parameter 4	char* : Defines the mode of the function. There are two different modes supported by the function: “text” und “XML” (siehe Beschreibung unten).
Response	ETHERNETSCANNER_INVALIDHANDLE (-1000) if the sensor handle (parameter 1) is NULL or invalid. In text mode: If the size of the raw buffer (parameter 2) is smaller than the size of the data to be written, the function returns ETHERNETSCANNER_GETINFOSMALLERBUFFER (-2). In a suc-cessful operation the function returns the length of the data written into the raw buffer. In XML mode: If the size of the raw buffer (parameter 2) is smaller than the size of the data to be written, the function returns ETHERNET_GETINFOSMALLBUFFER (-2). In a successful operation the function returns the length of the data written into the raw buffer. If the function fails to call the XML data from the profile sensor, it returns ETHERNETSCANNER_GETINFOIN-VALIDXML (-4).
Description	In text mode: Returns basic information about the profile sensor as a text such as sensor name, working ranges, MAC, etc (see appendix 1 for an example). In XML mode: Returns a full description of the profile sensor in XML format. The XML contains general information about the profile sensor, the current values of all features as well as all ASCII commands supported by the profile sensor in the firmware (see appendix 2 for an exam-ple).

10.9.2 GetInfo (XML mode)

The following XML data desription shows a part of the data returned by the function EthernetScanner_GetInfo (through parameter 2) in the XML mode:


```
<?xml version="1.0" encoding="UTF-8"?>
<device>
        <general>
          <ordernumber>MLWL221</ordernumber>

          <productversion>1.40</productversion>
          <producer>wenglor sensoric GmbH</producer>
          <description>2D-/3D-profle sensors</description>
          <hardwareversion>
                    <general>1.4.0</general>
          </hardwareversion>
.
.
.
                              <encoder_ttl_rs422>
                                        <current>0</current>
                                        <default>0</default>
                                        <command>SetEA4ResetCounterEncoderTTLRS422</command>
                                                  <parameter>0</parameter>
                                                  <parameter>1</parameter>
                                                  <help>"0: disabled 1: enabled"</help>
                                        </encoder_ttl_rs422>
                                        <help>"dependency ea functionresetcounter XML-section"</help>
                              </resetcounter>
                    </ea4>
          </usrio>
        </settings>
</device>
```


10.9.3 GetInfo (Text mode)

The following data desription shows an example of the data returned by the function EthernetScanner_GetInfo (through parameter 2) in the text mode:


```
[general]
sn=6
z_start=65.000
z_range=60.000
x_range_at_start=40.000
x_range_at_end=58.000
widthX=1280
widthZ=1024
```

11. TCP/IP Socket Interface

11.1 Introduction

The weCat3D sensor has a TCP/IP socket interface which needs only a working TCP/IP socket communication. Over the TCP/IP socket interface the commands can be transmitted in ASCII format. The data packet is in a binary format. The TCP/IP socket interface is available in FW 1.2.0 or higher.

11.2 Setup the TCP/IP Socket Communication

To establish a TCP/IP socket communication please follow the steps below:

1. Open a client TCP/IP socket communication to the sensor via port 32001
2. Initialize the TCP/IP socket interface of the sensor by sending following commands (\r = carriage return)
   a. SetAcquisitionStop\r
   b. Wait until all data is read out
   c. SetInitializeAcquisition\r
   d. SetLinearizationMode=1\r
   e. SetAcquisitionStart\r
   f. After approx. 0.5 s, sensor information and profile data are transferred via the TCP/IP socket.

End of transmission:

a. SetAcquisitionStop\r
b. Wait until all data is read out

11.3 Data Format Definition

11.3.1 Basic Data Formats

Type	Name	Size in bytes
Unsigned int	Unsigned integer	4
Unsigned short	Unsigned integer	2
Unsigned char	Unsigned integer	1
Signed char	Signed integer	1
Float	Floating point number	4
Void	Void data type	not defined
Unsigned int[n]	Array unsigned integer of length n	4*n
Unsigned short[n]	Array unsigned integer of length n	2*n
Unsigned char[n]	Array unsigned integer of length n	1*n
Float[n]	Array floating point number of length n	4*n

11.3.2 Complex Data Formats

<table>
<tr><th>Type</th><th>Name</th><th>Content</th><th>Description</th><th>Type</th><th>Size in bytes</th></tr>
<tr><td rowspan="3">ROIXDetail</td><td rowspan="3">Complex data type ROIX definition</td><td>Start</td><td>Start of ROI in X in pixel</td><td>unsigned short</td><td rowspan="3">6</td></tr>
<tr><td>Length</td><td>Length of ROI in X in pixel</td><td>unsigned short</td></tr>
<tr><td>Sub-samp-ling</td><td>Subsampling in X:
MLSL/MLWL:
0: no subsampling
1: MLSL subsampling (every second column is read out)</td><td>unsigned short</td></tr>
<tr><td>ROIXDetail[n]</td><td>Array ROIXDetail of length n</td><td></td><td></td><td></td><td>6*n</td></tr>
<tr><td rowspan="3">ROIZDetail</td><td rowspan="3">Complex data type ROIZ definition</td><td>Start</td><td>Start of ROI in Z in pixel</td><td>unsigned short</td><td rowspan="3">6</td></tr>
<tr><td>Length</td><td>Length of ROI in Z in pixel</td><td>unsigned short</td></tr>
<tr><td>Sub-samp-ling</td><td>Subsampling in Z
MLWL/MLSL:
0 = no subsampling
MLWL/MLSL:
1 = subsampling
(every second line is read out)</td><td>unsigned short</td></tr>
<tr><td>ROIZDetail[n]</td><td>Array ROIZDetail of length n</td><td></td><td></td><td></td><td>6*n</td></tr>
</table>

11.4 General Structure

Each data packet (Container) starts with the Container-Tag and ends with a CRC-Tag (checksum). In the container other tags containing sensor information and measurement data.

Byte Offset	Tag	Name	Size in bytes	Type	Min. occurance	Max. occurance
0	0x021A01FF	Container ID	4	unsigned int	1	1
Variable	0x021A0101	General ID	4	unsigned int	0	1
Variable	0x021A0102	Statistic ID	4	unsigned int	0	1
Variable	0x021A0103	Description ID (xml)	4	unsigned int	0	1
Variable	0x021A0201	ID-ROI-X	4	unsigned int	0	1
Variable	0x021A0202	ID-ROI-Z	4	unsigned int	0	1
Variable	0x021A0301	ID-RegisterCameraMLSL	4	unsigned int	0	1
Variable	0x021A0302	ID-RegisterCameraMLWL	4	unsigned int	0	1
Variable	0x021A0401	ID-RegisterFPGAMLSL	4	unsigned int	0	1
Variable	0x021A0402	ID-RegisterFPGAMLWL	4	unsigned int	0	1
Variable	0x021A0601	ID-Scan	4	unsigned int	0	1
Variable	0x021A0602	ID-ScanLinear	4	unsigned int	0	1
Variable	0x00000001	SubID-ScanLinearHeader	4	unsigned int	0	1
Variable	0x00000002	SubID-ScanLinearData	4	unsigned int	0	1
Variable	0x021A0801	ID-ScaleParam	4	unsigned int	0	1
Container-ID-Size-12	0x021AFFFF	CRC-ID	4	unsigned int	1	1

11.5 Structure of a Tag

Every tag starts with the tag ID and the total size of the tag in bytes.

Element	Description	Size in bytes	Type
Tag ID	Unique ID of the tag	4	unsigned int
Tag Size	Size of the tag in bytes	4	unsigned int
Tag Content	Content of the tag, depending on type	Tag size - 8 bytes	depending on tag

11.6 Description of Tag

The byte offset is always related to the beginning of the tag. All examples are in little endian formatted.

NOTE!
The container size can vary depending on the firmware version

11.6.1 Container Tag

The container tag contains the root of the data structure.

Byte Offset	Tag Data	Description	Size in bytes	Type
0	Container-ID	0x021A01FF A complete data package is included in the container	4	unsigned int
4	Container-ID-Size	Total size of the tag in bytes	4	unsigned int

11.6.2 General Tag

The general tag contains informaton like encoder values.

Byte Offset	Tag Data	Description	Size in bytes	Type
0	General-ID	Content: 0x021A0101	4	unsigned int
4	Size	Total size of the tag in bytes	4	unsigned int
8	PicCnt	Picture counter (always +1)	2	unsigned short
10	BaseTimeCnt	Internal FPGA counter in μs	4	unsigned int
14	Encoder HTL	Current HTL encoder value	4	unsigned int
18	SavedEncoderHTL	Stored HTL encoder value using reset encoder command	4	unsigned int
22	Encoder RS422	Current RS422 encoder value	4	unsigned int
26	SavedEncoderRS422	Stored RS422 encoder value using reset encoder command	4	unsigned int
30	E/A1 + E/A2	Current state of digital E/A1 and E/A2 Bit0: Input load E/A1 (0 = off) Bit1: Status E/A1 Bit2: Reserved Bit3: Reserved Bit4: Input load E/A2 (0 = off) Bit5: Status E/A2 Bit6: reserved Bit7: reserved	1	unsigned char
31	E/A3 + E/A4	Current state of digital E/A3 and E/A4 Bit0: Input load E/A3 (0 = off) Bit1: Status E/A3 Bit2: Reserved Bit3: Reserved Bit4: Input load E/A4 (0 = off) Bit5: Status E/A4 Bit6: reserved Bit7: reserved	1	unsigned char

32	Status Register	Bit0: Ready OK Bit1: Reserved Bit2: Reserved Bit3: Line numbers OK Bit4: Reserved Bit5: Overtrigger bit, triggering too fast Bit6: Reserved	2	unsigned short
34	Differential Inputs (Encoder422)	Signal TTL encoder inputs Bit0: ChA, Bit1: ChB, Bit2: ChC	1	unsigned char
35	Intensity-Peak1	Mean intensity of current profile, first peak	2	unsigned short
37	Intensity-Peak2	Mean intensity of current profile, second peak	2	unsigned short
39	ValidPoints-Peak1	Number of valid points in current profile, first peak	2	unsigned short
41	ValidPoints-Peak2	Number of valid points in current profile, second peak	2	unsigned short
43	Counter from input signal	Current counter of a user defined I/O (must be activated). Use SetEA1InputFunctionCounter... SetEA4InputFunctionCounter	4	unsigned int
47	CurrentExpTime	Current exposure time in μs	3	unsigned char[3]
50	OPT3013	Bit0: Reserved Bit1: Blinking mode Bit2: Measurement mode Bit3: Profile enable status Bit4: Dynamic trigger status Bit5: Profile points detection status Bit6: Red laser status Bit7: Blinking mode profiles sending status	1	unsigned char
51	Reserved		1	unsigned char

11.6.3 Statistic Tag

The statistic tag contains sensor information like temperature

Byte Offset	Tag Data	Description	Size in bytes	Type
0	Statistic-ID	0x021A0102	4	unsigned int
4	Statistic-Data-Size	Total size of the tag in bytes	4	unsigned int
8	Voltage1	Input voltage in Volt/100	2	unsigned short
10	Reserved	Reserved	2	unsigned short
12	CPU-FiFo	FiFo status CPU in bytes	4	unsigned int
16	FPGA-FiFo	FiFo status FPGA in bytes	4	unsigned int
20	Reserved	Reserved	6	void

26	OnOffCounter-CPU	Counter switching on sensor	2	unsigned short
28	OnTimeCounter-CPU	Operation timer in 1/4 seconds	4	unsigned int
32	Temperature-CPU	Temperature in grad Celsius of CPU	1	signed char
33	Reserved	Reserved	2	void
35	Temperature-Laser	Temperature in grad Celsius of laser	1	signed char
36	LaserPower	PWM-Signal, only for MLSL2x7x and MLWLx7x	2	unsigned short
38	Mac address	Mac address	6	unsigned char[6]
44	Frequency: camera	In Hz	2	unsigned short
46	Bandwith: Ethernet	In *10 kBytes	2	unsigned short
48	Reserved	Reserved	5	void
53	User-Data	Can be set by command SetStatisticDataUserData=xxx	2	unsigned char[2]
55	Reserved	Reserved	1	void
56	Reserved	Reserved	4	unsigned char[4]

11.6.4 Description Tag

The description tag contains the XML description of the sensor settings.

Byte Offset	Tag Data	Description	Size in bytes	Type
0	Description-ID	0x021A0103	4	unsigned int
4	Description-Size	Total size of the tag in bytes	4	unsigned int
8	Description Data (xml)	Sensor data in XML format	Variable	unsigned char [size-8]

11.6.5 ROI-X Tag

The ROI-X Tag contains information about the ROI settings in X.

Byte Offset	Tag Data	Description	Size in bytes	Type
0	ROI-X ID	0x021A0201	4	unsigned int
4	ROI-X Size	Total size of the tag in bytes	4	unsigned int
8	X-Number	Number n of ROI in X	2	unsigned short
10	ROI-X Details	Definition of n ROI in X	6*n	ROIXDetail[n]
10+6*n	Reserved	Reserved	Size-(10+6*n)	unsigned short

11.6.6 ROI-Z Tag

The ROI-X Tag contains information about the ROI settings in Z.

Byte Offset	Tag Data	Description	Size in bytes	Type
0	ROI-Z ID	0x021A0202	4	unsigned int
4	ROI-Z Size	Total size of the tag in bytes	4	unsigned int
8	Z-Number	Number n of ROI in Z	2	unsigned short
10	ROI-Z Details	Definition of n ROI in Z	6*n	ROIZDetail[n]
10+6*n	Reserved	Reserved	Size-(10+6*n)	unsigned short

11.6.7 RegisterCameraMLSL

(only for MLSL sensors)

Byte Offset	Tag Data	Description	Size in bytes	Type
0	RegisterCameraMLSL	0x021A0301	4	unsigned int
4	Size	Total size of the tag in bytes	4	unsigned int
8	Reserved	Reserved	Size-8	unsigned char[Size-8]

11.6.8 RegisterCameraMLWL

(only for MLWL sensor)

Byte Offset	Tag Data	Description	Size in bytes	Type
0	RegisterCameraMLWL	0x021A0302	4	unsigned int
4	Size	Total size of the tag in bytes	4	unsigned int
8	Reserved	Reserved	Size-8	unsigned char[Size-8]

11.6.9 Register FPGAMLSL

(only for MLSL sensors)

Byte Offset	Tag Data	Description	Size in bytes	Type
0	RegisterFPGAMLSL	0x021A0401	4	unsigned int
4	Size	Total size of the tag in bytes	4	unsigned int
8	Reserved	Reserved	Size-8	unsigned char[Size-8]

NOTE!
Size may be changed in the case of firmware updates.

11.6.10 Register FPGAMLWL

(only for MLWL sensors)

Byte Offset	Tag Data	Description	Size in bytes	Type
0	RegisterFPGAMLWL	0x021A0402	4	unsigned int
4	Size	Total size of the tag in bytes	4	unsigned int
8	Reserved	Reserved	Size-8	unsigned char[Size-8]

NOTE!
Size may be changed in the case of firmware updates.

11.6.11 Linearization Table

The linearization table contains information used by the DLL which is provided by the SDK. The content is not documented.

Byte Offset	Tag Data	Description	Size in bytes	Type
0	Linearization tag	0x1907	2	unsigned short
4	Size	Total size of the tag in bytes	4	unsigned int
8	Data	Not documented	Size-10	void
Size-4	CRC	Checksum over all data without last 4 bytes	4	unsigned int

11.6.12 ScanNonLinear

Reserved in case that the data are not processed inside of the sensor.

Byte Offset	Tag Data	Description	Size in bytes	Type
0	Scan	0x021A0601	4	unsigned int
4	Size	Total size of the tag in bytes	4	unsigned int
8	Reserved	Reserved	Size-8	void

11.6.13 ScanLinear

Contains the data and information of the measured profile.

Byte Offset	Tag	Description	Size in bytes	Type	Min. occurance	Max. occurance
0	0x021A0602	ScanDataLinear	4	unsigned int	1	1
4	0x00000001	SubID-ScanDataLinearHeader	4	unsigned int	1	1
8	0x00000002	SubID-ScanDataLinearData	4	unsigned int	1	1

Byte Offset	Tag Data	Description	Size in bytes	Type
0	ScanLinear	0x021A0602	4	unsigned int
4	Size	Total size of the tag in bytes	4	unsigned int

11.6.14 SubID-ScanLinearHeader

Contains information how the data are formatted

Byte Offset	Tag Data	Description	Size in bytes	Type
0	SubID-ScanDataLinearHeader	0x00000001	4	unsigned int
4	ScanDataLinearHeader-Size	Total size of the tag in bytes	4	unsigned int
ScanDataLinearHeaderData				
8	NumberOfPoints	MLSL:1280 MLWL: 2048	4	unsigned int
12	NumberOfPeaks	1 or 2	1	unsigned char
13	NumberOfElementsPerPoint	Max 4	1	unsigned char
14	HDR: 0 = ExpTime1 1 = ExpTime2		1	unsigned char
15	Reserved	Reserved	5	unsigned char[5]
Element 1 of 4				
20	ID-Name[0]: 0 = Dummy 1 = X 2 = Z 3 = Y 4 = I 5 = Peak width (PW)	2 = Z	1	unsigned char
21	Type: 0 = unsigned int 1 = float	0	1	unsigned char
22	Size in bits	16	1	unsigned char
23	Reserved	Reserved	1	unsigned char
Element 2 of 4				
24	ID-Name[0]: 0 = Dummy 1 = X 2 = Z 3 = Y 4 = I (Bit7-0: Int-Bit 10-2) 5 = Peak width (PW)	4 = I	1	unsigned char
25	Type: 0 = unsigned int	0	1	unsigned char
26	Size in bits	10	1	unsigned char
27	Reserved		1	unsigned char

Byte Offset	Tag Data	Description	Size in bytes	Type
Element 3 of 4				
28	ID-Name[0]: 0 = Dummy 1 = X 2 = Z 3 = Y 4 = I 5 = Peak width (PW) 5 = I-Low + PW(Bit7-6: Int-Low-Bit1-0, Bit5-0: PW-Bit-5..0)	5 = PW	1	unsigned char
29	Type: 0 = insigned int	0	1	unsigned char
30	Size in bits	6	1	unsigned char
31	Reserved		1	unsigned char
Element 4 of 4				
32	ID-Name[0]: 0 = Dummy 1 = X 2 = Z 3 = Y 4 = I 5 = Peak width (PW) 5 = I-height (Int-height-Bit7-2)	1 = X	1	unsigned char
33	Type: 0 = unsigned int	0	1	unsigned char
34	Size in bits	16	1	unsigned char
35	Reserved		1	unsigned char
36	Reserved	Reserved	4	unsigned char[4]

11.6.15 SubID-ScanLinearData

Contains the data.

Byte Offset	Tag Data	Description	Size in bytes	Type
0	SubID-ScanDataLinearData	0x00000002	4	unsigned int
4	ScanDataLinearData-Size	Total size of the tag in bytes	4	unsigned int
8	Z, I, PW, X	Structure as defined in SubID-ScanLinearHeader section 11.6.14	6*1280 for MLSL 6*2048 for MLWL	unsigned short[1280][3] for MLSL unsigned short[2048][3] for MLWL

Example how the data are typical formatted for one data point:

Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Value X [16 bit]		Intensity [10 bit] + Peak Width [6 bit]		Value Z [16 bit]	
Bit 15...Bit 0		Bit 15...Bit 6	Bit 5...Bit 0	Bit 15...Bit 0	

Definition of structure as new type (code snippets):

```
typedef struct _LINDATA_Z_I_PW_X
        {
                unsigned short usZ;
                unsigned short usPWIntensity;
                unsigned short usX;
        } structLinData_Z_I_PW_X;
```

Loop going through array and data and read the individual data rows (for more details s. example in SDK):

```
for (int i = 0; i < numPoints; i++)
        {
                for (iCurrentPeak = 0;
                        iCurrentPeak < MLContainer->ScanDataLin.ScanDataLinGeneral.
ucNumberOfPeaks &
                        iCurrentPeak < ETHERNETSCANNER_PEAKSPERCMOSSCANLINEMAX;
                        iCurrentPeak++)
                {
                        int iLinDataCnt = i * (iCurrentPeak + 1);

                        //if (bActiveZ)
                        {
                                m_doEthernetScannerBufferZ[iZeilenCnt] = _structLin-
Data_Z_I_PW_X[iLinDataCnt].usZ;
                                m_doEthernetScannerBufferZ[iZeilenCnt] *= MLContain-
er->ScanDataLinScaleParams.flZScale;
                                m_doEthernetScannerBufferZ[iZeilenCnt] += MLContain-
er->ScanDataLinScaleParams.flZOffset;
                        }
                        //if (bActiveIntensity)
                        {
                                m_iEthernetScannerBufferI[iZeilenCnt] = ((_struct-
LinData_Z_I_PW_X[iLinDataCnt].usPWIntensity >> 6) & 0x3FF);
                        }
                        //if (bActivePeakWidth)
                        {
                                m_iEthernetScannerBufferPeakWidth[iZeilenCnt] = (_
structLinData_Z_I_PW_X[iLinDataCnt].usPWIntensity) & 0x3F;
                        }
                        //if (bActiveX)
                        {
                                m_doEthernetScannerBufferX[iZeilenCnt] = _structLin-
Data_Z_I_PW_X[iLinDataCnt].usX;
```

```
                              m_doEthernetScannerBufferX[iZeilenCnt] *= MLContain-
er->ScanDataLinScaleParams.flXScale;
                              m_doEthernetScannerBufferX[iZeilenCnt] += MLContain-
er->ScanDataLinScaleParams.flXOffset;
                       }
               }
               iZeilenCnt++;
       }
```

11.6.16 ScaleParam

Contains the information how the dat must be scaled to convert it into mm dimension.

Byte Offset	Tag Data	Description	Size in bytes	Type
0	ScaleParam	0x021A0801	4	unsigned int
4	Size	Total size of the tag in bytes	4	unsigned int
8	X-Scale	Scaling factor X in mm	4	float
12	X-Offset	Offset X in mm	4	float
16	Z-Scale	Scaling factor Z in mm	4	float
20	Z-Offset	Offset Z in mm	4	float

X value [mm] = X-Scale*integer value x + X-Offset
Z value [mm] = Z-Scale*integer value z + Z-Offset

11.6.17 CRC

Tag for Checksum

Byte Offset	Tag Data	Description	Size in bytes	Type
0	CRC	0x021AFFFF	4	unsigned int
4	Size	Total size of the tag in bytes	4	unsigned int
8	Dummy data	Total size container must be modulo 64 bytes		unsigned int[Description-ID-Size-12]
Size-4	CRC-Sum	Check sum container without 4 last bytes (32 bit CRC Polynominal 0x04C11DB7)	4	unsigned int

11.7 Typical Data Sets

After connection to the sensor following data are typically transmitted by default.

1. The so called linearization table which is not used by the user and can be ignored.
2. The XML description of the sensor settings. The description is in plain terms formatted as XML.
3. The measurement data after the setup is done, see section 10.3.

11.7.1 Overview Typical Data Stream MLSL

Tag	**Tag ID**
After open socket communication	
Linearization table	
Linearization table	0x1907
Description sensor settings in XML	
Container	0x021a01ff
Description	0x021a0103
CRC	0x021AFFFF
After command SetAcquisitionStart Measurement data	
Container	0x021A01FF
ROI-X	0x021A0201
ROI-Z	0x021A0202
General	0x021A0101
Statistic	0x021A0102
SCaleParam	0x021A0801
ScanLinear	0x021A0602
RegisterFPGAMLSL	0x021A0401
RegisterCameraMLSL	0x021A0301
CRC	0x021AFFFF
Each new measurements generate a new container	

11.7.2 Overview Typical Data Stream MLWL

Tag	**Tag ID**
After open socket communication	
Linearization table	
Linearization table	0x1907
Description sensor settings in XML	
Container	0x021a01ff
Description	0x021a0103
CRC	0x021AFFFF
After command SetAcquisitionStart Measurement data	
Container-ID	0x021A01FF
General-ID	0x021A0101
Statistic-ID	0x021A0102
ID-RegisterCameraMLWL	0x021A0302
ID-RegisterFPGAMLWL	0x021A0402
ID-ROI-X	0x021A0201
ID-ROI-Z	0x021A0202
ScaleParam	0x021A0801
ScanLinear	0x021A0602
CRC	0x021AFFFF
Each new measurements generate a new container	

11.7.3 Example First Data After Connection

Linearization table:

Tag	Tag size in bytes	Offset in bytes	0	1	2	3	4	5	6	7	Tag element	Content
0x1907	182880	0	07	19	60	ca	02	00	4c	49	Linearization table	
		snipped data	22	7d	7d	00	91	8f	38	57	Size	182880
											Data	not documented
											CRC	1463324561

XML description:

Tag	Tag size in bytes	Offset in bytes	0	1	2	3	4	5	6	7	Tag element	Content
0x021a01ff	41388	0	ff	01	1a	02	ac	a1	00	00	Container	
											Size	41388
0x021a0103	41368	8	03	01	1a	02	98	a1	00	00	XML description	
			3c	3f	78	6d	6c	20	76	65	Size	41368
			72	73	69	6f	6e	3d	22	31	Content	<?xml ver-sion="1.0" encoding="UTF-8" ?> <device> <g ... ice>
		snipped XML data	2e	30	22	20	65	6e	63	6f		
			69	63	65	3e	0d	0a	00	00		
0x021AFFFF	12	41376	ff	ff	1a	02	0c	00	00	00	CRC	
			06	3f	d6	ff					Size	12
											Dummy data	Dummy data to increase total container byte size to a value which is modulo 64 bytes (9280 bytes modulo 64 bytes =0).
											CRC-Sum (32 bit CRC Polynom 0x04C11DB7)	4292230918

11.7.4 Example MLSL Container

Tag	Tag size in bytes	Offset in bytes	0	1	2	3	4	5	6	7	Tag element	Content
0x021a01ff	9280		ff	01	1a	02	40	24	00	00	Container	
											Size	9280
0x021a0201	16	8	01	02	1a	02	10	00	00	00	ROI-X ID	
			01	00	00	00	00	05	00	00	Size	16
											X-Number	1
											ROI-X Details	0;1024;0
0x021a0202	16	24	02	02	1a	02	10	00	00	00	ROI-Z ID	
			01	00	00	00	00	04	00	00	Size	16
											X-Number	1
											ROI-X Details	0;1024;0

Tag	Tag size in bytes	Offset in bytes	0	1	2	3	4	5	6	7	Tag element	Content
0x021a0101	52	40	01	01	1a	02	34	00	00	00	General	
			06	38	6b	4d	22	e0	01	00	Size	52
			00	00	00	00	00	00	01	00	PicCnt	14342
			00	00	00	00	00	00	88	84	BaseTimeCnt	3760344427
			5f	00	07	c4	03	00	00	00	EncoderHTL	1
			05	00	00	00	00	00	00	96	SavedEncoderHTL	
			00	00	00	00					EncoderRS422	1
											SavedEncoderRS422	
											Current state of digital E/A1 and E/A2: Bit0: Input load E/A1 (0 = off) Bit1: Status E/A1 Bit2: Reserved Bit3: Reserved Bit4: Input load E/A2 (0 = off) Bit5: Status E/A2 Bit6: Reserved Bit7: Reserved	
											Current state of digital E/A3 and E/A4 Bit0: Input load E/A3 (0 = off) Bit1: Status E/A3 Bit2: Reserved Bit3: Reserved Bit4: Input load E/A4 (0 = off) Bit5: Status E/A4 Bit6: Reserved Bit7: Reserved	
											M2GL-Status: Register_128	Bit 7 =1
											Differential Inputs(Encoder422)	7
											Bit0: ChA, Bit1: ChB, Bit2: ChC	
											Intensity-Peak1	964
											Intensity-Peak2	
											ValidPoints-Peak1	1280
											ValidPoints-Peak2	
											Counter from Input Signal	
											CurrentExpTime	150
											Reserved	
0x021a0102	60	92	02	01	1a	02	3c	00	00	00	Statistic	
			47	05	0e	00	02	00	00	00	Statistic-Data-Size	60
			18	21	00	00	5f	00	84	ad	Voltage1	1351
			49	00	5f	00	84	ad	49	00	Reserved	
			38	38	38	38	ff	00	54	4a	CPU-FiFo	2
			05	0a	08	04	c7	00	01	07	FPGA-FiFo	8472
			01	07	00	00	00	00	00	00	Reserved	
			00	00	00	00					OnOffCounter-CPU	95
											OnTimeCounter-CPU	4828548*1/4 [s]=1207137 s
											Temperatur-CPU	56
											Reserved	
											Temperatur-Laser	56
											LaserPower	255
											mac address	04:08:0a:05:4a:54
											Frequency: Camera	199
											Bandwidth: Eth	1793*10 kB=17930 kB
											Reserved	
											User-Data	0x0000
											Reserved	
											Reserved	

Tag	Tag size in bytes	Offset in bytes	0	1	2	3	4	5	6	7	Tag element	Content
0x021a0801	24	152	01	08	1a	02	18	00	00	00	ScaleParam	
			7c	85	79	3a	6e	56	ef	c1	Description-ID-Size	24
			2b	ed	85	3a	b5	ff	79	42	X-Scale	0,00095185
											X-Offset	
											Z-Scale	0,00102178
											Z-Offset	62,4997139
0x021a0602	7736	176	02	06	1a	02	38	1e	00	00	ScanLinear	
			01	00	00	00	20	00	00	00	ScanLinear-ID -Size	7736
			00	05	00	00	01	04	00	00	SubID-ScanDataLinearHeader	0x00000001
			00	00	00	00	02	00	10	00	ScanDataLinearHeader-Size	32
			04	00	0a	00	05	00	6	00	ScanDataLinearHeaderData:	
			01	00	10	00	00	00	00	00	NumberOfPoints	1280
			02	00	00	00	08	1e	00	00	NumberOfPeaks	1
			cc	59	08	ce	87	19	d0	59	NumberOfElementsPerPoint	4
			c8	d2	ae	19	fa	59	c8	d5	HDR: 0=ExpTime1, 1=ExpTime2	
			d0	19	fc	59	c8	d6	f8	19	Reserved	5 bytes
			fb	59	c8	d9	21	1a	e7	59	Element: 1 from 4	
			c9	d1	4c	1a	d2	59	c8	da	ID-Name[0]: 0=Dummy, 1=X, 2=Z, 3=Y, 4=I, 5=Peak width (PW)	2=Z
			78	1a	d0	59	08	d7	a1	1a	Type: 0=unsigned int, 1=float	0=unsigned int
			d1	59	88	d5	c9	1a	d1	59	Size in bits	16
			08	d8	f1	1a	d3	59	c8	db	Reserved	
			19	1b	d6	59	c8	dc	41	1b	Element: 2 from 4	
			ec	59	c9	d0	66	1b	ee	59	ID-Name[0]: 0=Dummy, 1=X, 2=Z, 3=Y, 4=I, 5=Peak width (PW)	4=I
			c9	cf	8e	1b	ee	59	89	d3	4=I (Bit7-0: Int-Bit10-2)	
		Snipped	b6	1b	ed	59	c9	d6	df	1b	Type: 0=unsigned int	0=unsigned int
		data	eb	59	49	d7	08	1c	e9	59	Size in bits	10
			09	d5	30	1c	da	59	c8	da	Reserved	
			5b	1c	ee	59	89	cf	80	1c	Element: 3 from 4	
			f2	59	49	d3	a8	1c	f1	59	ID-Name[0]: 0=Dummy, 1=X, 2=Z, 3=Y, 4=I, 5=Peak width (PW)	5
			49	d0	d1	1c	db	59	48	d2	5=I-Low + PW(Bit7-6: Int-Low-Bit1-0, Bit5-0: PW-Bit-5..0)	
			fc	1c	dc	59	08	d0	24	1d	Type: 0=unsigned int	0=unsigned int
			05	5a	48	d5	47	1d	0a	5a	Size in bits	6
			48	d8	6e	1d	0e	5a	c8	d9	Reserved	
			96	1d	14	5a	08	de	bd	1d	Element: 4 from 4	
			17	5a	88	e0	e5	1d	2b	5a	ID-Name[0]: 0=Dummy, 1=X, 2=Z, 3=Y, 4=I, 5=Peak width (PW)	1=X
			09	d5	0b	1e	2d	5a	c9	d2	Type: 0=unsigned int	0=unsigned int
			33	1e	2b	5a	89	d0	5b	1e	Size in bits	16
			10	5a	08	d6	88	1e	05	5a	Reserved	
			c8	ce	b2	1e	e1	59	08	cd	Reserved	
			e0	1e	e0	59	48	ce	08	1f	SubID-ScanDataLinearData	0x0000002
		snipped data	e0	59	88	ce	31	1f	f2	59	ScanDataLinearData-Size	7688
			47	dd	56	1f	f6	59	87	dd		
			7e	1f	09	5a	08	d1	a4	1f	Z,I,PW,X	1280 data values (size 7680 bytes)
			0b	5a	08	d5	cc	1f	0b	5a		
			8a	e5	7b	61	09	dd	b4	e5		
0x021a0401	300	7912	01	04	1a	02	2c	01	00	00	RegisterFPGAMLSL	
			5f	80	88	13	00	00	00	00	Description-ID-Size	300
			00	00	00	96	00	00	00	00	Reserved	292 bytes reserved
			00	00	00	00						
0x021a0301	1032	8212	01	03	1a	02	08	04	00	00	RegisterCameraMLSL	
			d0	50	01	00	00	00	00	00	Description-ID-Size	1032

Tag	Tag size in bytes	Offset in bytes	0	1	2	3	4	5	6	7	Tag element	Content
			00	00	00	00	00	00	00	00	Reserved	1024 bytes reserved
			01	cd	01	c9	00	c1	20	00		
0x021affff	36	9244	ff	ff	1a	02	24	00	00	00	CRC-Tag-ID	
			00	00	00	00	00	00	00	00	CRC-Tag-ID-size	36
			00	00	00	00	00	00	00	00	Dummy data	Dummy data to increase total container byte size to a value which is modulo 64 bytes (9280 bytes modulo 64 bytes =0).
			00	00	00	00	00	00	00	00	CRC-Sum (32 bit CRC Polynom 0x04C11DB7)	3655526239
			5f	e7	e2	d9						
Total size		9280										

11.7.5 Example MLWL Container

Tag	Tag size in bytes	Offset in bytes	0	1	2	3	4	5	6	7	Tag element	Content
0x021a01ff	12992		ff	01	1a	02	c0	32	00	00	Container	
											Size	12992
0x021a0101	52	8	01	01	1a	02	34	00	00	00	General	
			b7	b8	21	7e	ca	3b	00	01	Size	52
			00	00	00	00	00	00	01	00	PicCnt	8632
			00	00	00	00	00	00	00	20	BaseTimeCnt	1271561761
			5f	00	02	36	02	00	00	00	EncoderHTL	256
			08	00	00	00	00	00	00	95	SavedEncoderHTL	0
			00	00	7c	00					EncoderRS422	1
											SavedEncoderRS422	0
											USRIO1+USRI-O2(Bit3:in,Bit2:oe,Bit1:inn,Bit0:sk)	
											USRIO3+USRI-O4(Bit3:in,Bit2:oe,Bit1:inn,Bit0:sk)	
											M2GL-Status: Register_128	
											Differential Inputs(Encoder422)	2
											Bit0: ChA, Bit1: ChB, Bit2: ChC	
											Intensity-Peak1	0
											Intensity-Peak2	8
											ValidPoints-Peak1	0
											ValidPoints-Peak2	0
											Counter from Input Signal	0
											CurrentExpTime	149
											OPT3013	
											Reserved	
0x021a0102	60	60	02	01	1a	02	3c	00	00	00	Statistic	
			d6	08	0c	00	3c	04	00	00	Statistic-Data-Size	60
			00	00	00	00	10	00	a7	49	Voltage1	2262
			25	01	10	00	d7	09	49	00	Reserved	
			2b	2e	00	2b	ff	00	54	4a	CPU-FiFo	1084
			05	0a	06	8c	64	00	00	00	FPGA-FiFo	

Tag	Tag size in bytes	Offset in bytes	0	1	2	3	4	5	6	7	Tag element	Content
			01	02	00	00	00	00	00	01	Reserved	
			00	00	00	00					OnOffCounter-CPU	16
											OnTimeCounter-CPU	4786647*1/4 [s]=1207137 s
											Temperatur-CPU	43
											Reserved	
											Temperatur-Laser	43
											LaserPower	255
											mac address	84:74:5:10:6:140
											Frequency: Camera	100
											Bandwidth: Eth	0*10 kB=0 kB
											Reserved	
											User-Data	0x0000
											Reserved	
											Reserved	
0x021a0302	136	120	02	03	1a	02	88	00	00	00	RegisterCameraMLWL	
		Snipped data	00	00	08	00	00	00	00	00	Size	136
			00	00	00	00	00	00	00	00	Reserved	
			00	00	00	62	00	43	53	05		
0x021a0402	320	256	02	04	1a	02	40	01	00	00	RegisterFPGAMLWL	
		Snipped data	5f	80	10	27	00	00	00	00	Size	320
			00	00	00	96	00	00	00	00	Reserved	
			00	00	00	00	00	00	00	00		
0x021a0201	16	576	01	02	1a	02	10	00	00	00	ROI-X ID	
			01	00	00	00	00	08	00	00	Size	16
											X-Number	1
											ROI-X Details	0;8;0
											Reserved	
0x021a0202	16	592	02	02	1a	02	10	00	00	00	ROI-Z-ID	
			01	00	00	00	00	08	00	00	Size	16
											Z-Number	1
											ROI-Z Details	0;8;0
											Reserved	
0x021a0801	24	608	01	08	1a	02	18	00	00	00	ScaleParam	
			c8	cc	e6	3a	c6	16	66	c2	Size	24
			c1	cd	07	3b	6d	c2	a3	42	X-Scale	0.00176086
											X-Offset	-57.5222
											Z-Scale	0.0020722
											Z-Offset	81.8797
0x021a0602	12344	632	02	06	1a	02	38	30	00	00	ScanLinear	
			01	00	00	00	28	00	00	00	ScanLinear-ID -Size	12344
			00	08	00	00	01	04	00	00	SubID-ScanDataLinearHeader	0x00000001
			00	00	00	00	02	00	10	00	ScanDataLinearHeader-Size	40
			04	00	0a	00	05	00	06	00	ScanDataLinearHeaderData:	
			01	00	10	00	00	00	00	00	NumberOfPoints	2048
			02	00	00	00	08	30	00	00	NumberOfPeaks	1

Tag	Tag size in bytes	Offset in bytes	0	1	2	3	4	5	6	7	Tag element	Content
			4e	77	c8	ae	84	db	5b	77	NumberOfElementsPerPoint	4
			c8	ae	72	db	62	77	c8	b0	HDR: 0=ExpTime1, 1=ExpTime2	
			5c	db	6c	77	c8	b2	48	db	Reserved	5 bytes
			77	77	c8	b1	34	db	7e	77	Element: 1 from 4	
			89	a7	1f	db	84	77	08	af	ID-Name[0]: 0=Dummy, 1=X, 2=Z, 3=Y, 4=I, 5=Peak width (PW)	2=Z
			0a	db	84	77	88	af	f3	da	Type: 0=unsigned int, 1=float	0=unsigned int
			8c	77	c8	b0	de	da	8f	77	Size in bits	16
			88	b2	c8	da	95	77	08	b3	Reserved	
			b2	da	9d	77	88	af	9e	da	Element: 2 from 4	
			aa	77	88	ae	8a	da	ac	77	ID-Name[0]: 0=Dummy, 1=X, 2=Z, 3=Y, 4=I, 5=Peak width (PW)	4=I
			c8	b1	74	da	b6	77	48	b5	4=I (Bit7-0: Int-Bit10-2)	
			60	da	b6	77	c8	b3	49	da	Type: 0=unsigned int	0=unsigned int
			bc	77	c8	af	34	da	ca	77	Size in bits	10
			07	b7	20	da	ca	77	47	b6	Reserved	
			0a	da	cd	77	07	b6	f4	d9	Element: 3 from 4	
		Snipped data	ce	77	c7	b6	dd	d9	d3	77	ID-Name[0]: 0=Dummy, 1=X, 2=Z, 3=Y, 4=I, 5=Peak width (PW)	5=PW
			88	ae	c8	d9	da	77	48	ad	5=I-Low + PW(Bit7-6: Int-Low-Bit1-0, Bit5-0: PW-Bit-5..0)	
			b2	d9	e1	77	08	ac	9e	d9	Type: 0=unsigned int	0=unsigned int
			ef	77	47	b5	8a	d9	f0	77	Size in bits	6
			07	b7	74	d9	fd	77	88	b0	Reserved	
			60	d9	fd	77	88	af	4a	d9	Element: 4 from 4	
			02	78	48	ad	34	d9	0e	78	ID-Name[0]: 0=Dummy, 1=X, 2=Z, 3=Y, 4=I, 5=Peak width (PW)	1=X
			c8	ad	20	d9	18	78	87	b8	Type: 0=unsigned int	0=unsigned int
			0c	d9	20	78	c8	ae	f6	d8	Size in bits	16
			26	78	08	ae	e2	d8	29	78	Reserved	
			c8	af	cc	d8	2c	78	88	ae	Reserved	
			b6	d8	30	78	c8	ac	9f	d8	SubID-ScanDataLinearData	0x0000002
			3c	78	88	ad	8c	d8	45	78	ScanDataLinearData-Size	12296
			88	ae	77	d8	45	78	c8	ae		
			60	d8	4e	78	48	ad	4c	d8	Z,I,PW,X	2048 data values (size 12288 bytes)
			4d	78	08	ad	34	d8	4f	78		
			b7	14	1f	b2	c8	89	97	14		
0x021affff	16	12976	ff	ff	1a	02	10	00	00	00	CRC-Tag-ID	
			00	00	00	00	c5	d0	65	9b	CRC-Tag-ID-Size	16
											Dummy data	Dummy data to increase total container byte size to a value which is modulo 64 bytes (9280 bytes modulo 64 bytes =0).
											CRC-Sum (32 bit CRC Polynom 0x04C11DB7)	2607141061
Total size		12992										

11.8 Implementation Recommendation

For easy implementation it is recommended to define a complex data type in the structure of the container. The bits of the container are copied to the complex data type. See the provided SDK example for details.

11.9 CRC Checksum Calculation

The CRC checksum can be calculated using following algorithm provided in code snippets.

Definitions in header file:

```
#define CRCPOLYNOMIAL 0x04C11DB7L

/*!
 * Function to calculate the CRC checksum of the container tag.
 * \param[in] crc_accum start value of CRC calculation
 * \param[in] *data_blk_ptr pointer to the data in the container tag
 * \param[in] data_blk_size size of the data set equals to container size - 4
 * \return value of the calculated checksum
 */
unsigned int CalculateCRC(unsigned int crc_accum, unsigned char *data_blk_ptr,
unsigned int data_blk_size);
```

Implementation of function:

```
unsigned int CalculateCRC(unsigned int crc_accum, unsigned char *data_blk_ptr,
unsigned int data_blk_size)
{
   register unsigned int i, j;
   unsigned int uiCRCTable[256];
      boolean bCRCTableInitialize = false;;

   if (data_blk_size > 10000000)
   {
       return 0;
   }

   if (bCRCTableInitialize == false)
   {
       bCRCTableInitialize = true;
       register unsigned short int i, j;
       register unsigned int crc_accum;

       for (i = 0; i<256; i++)
       {
              crc_accum = ((unsigned int)i << 24);
              for (j = 0; j < 8; j++)
              {
                     if (crc_accum & 0x80000000L)
```

```
                              crc_accum = (crc_accum << 1) ^ CRCPOLYNOMIAL;
                       else
                              crc_accum = (crc_accum << 1);
                }
                uiCRCTable[i] = crc_accum;
         }
    }

    for (j = 0; j<data_blk_size; j++)
    {
         i = ((int)(crc_accum >> 24) ^ *data_blk_ptr++) & 0xFF;
         crc_accum = (crc_accum << 8) ^ uiCRCTable[i];
    }
    return crc_accum;
}
```

Example usage:

```
/*!
 * ucBuffer is a pointer to the data of the container tag
 * uiBuffer is the size of the container tag
 */
unsigned int uiCalculatedCRC = CalculateCRC(-1, ucBuffer, uiBuffer - 4);
```

12. GigE Vision Interface

GigE Vision is an industrial standard which allows the integration of cameras and also 2D-/3D-sensors. The sensor act as GigE Vision server and a GigE Vision compatible software act as client.
To connect a weCat3D sensor using GigE Vision two ways are possible.

1. Embedded GigE Vision: Activating the embedded GigE Vision Mode, which is integrated in the sensor (see section 12.1)
2. External GigE Vision: Using a Windows based GigE Vision interface running on the IPC.

12.1 Embedded GigE Vision

To activate the embedded GigE Vision mode either the web interface or the OLED display can be used (see sections 7 and 8).

12.2 External GigE Vision

12.2.1 Introduction

This section describes how to use the external GigE Vision interface for weCat3D sensors.

The weCat3D GigE Vision interface is available for download on http://www.wenglor.com.
It is located on the product page of the 2D/3D Profile Sensors under the tab Download.

Content:

- weCat3D GigE Vision interface (64bit)
- Halcon Demo program

12.2.2 System Requirements

The weCat3D GigE Vision interface is developed as an external application that can run on any computer with Windows 7 or 10, x64, Linux Ubuntu 16.04, 18.08 or OpenSuse,version 42.
The minimum requirements to run one application are i3 Intel processor or any processor that supports SSE2 instructions set (please refer to your CPU datasheet). 4 GByte RAM or higher and a Gigabit Ethernet adapter. These requirements are only valid for running one weCat GigE Vision interface per computer. If several weCat GigE Vision interfaces should run on one computer (connection to several profile sensors) the requirements shall increase.

12.2.3 Network and Computer Setup

In order to guarantee a smooth experience of the weCat3D GigE Vision interface the computer as well as the network adapter should be set up accordingly:

12.2.3.1 Deactivation of the GigE Vision Filters (Drivers)

Normally when a GigE Vision client (Halcon, Matrix Vision, Ebus player, Eyevision, etc.) is installed the client also installs a GigE Vision filter/driver.These filters/drivers are used to improve the communication between the client and the GigE Vision device by filtering the UDP packets and reduce the CPU load of the computer.

Since the weCat3D profile sensors use an external application to make the profile sensor compatible with GigE Vision standards and since the interface could run on the same computer as the client (localhost) it is important to deactivate all GigE Vision filters/drivers. The GigE Vision filters/drivers block all large UDP packets which are transmitted between the GigE Vision interface and the client on localhost. If the GigE Vision filter is active, the client could fail to receive any images from the weCat3D GigE Vision interface.

NOTE!
For deactivating the GigE Vision filters/drivers make a right click on the network connection you are using. Select “Properties” in the context menu and the window with the network properties will be displayed (see section 8.5.3).

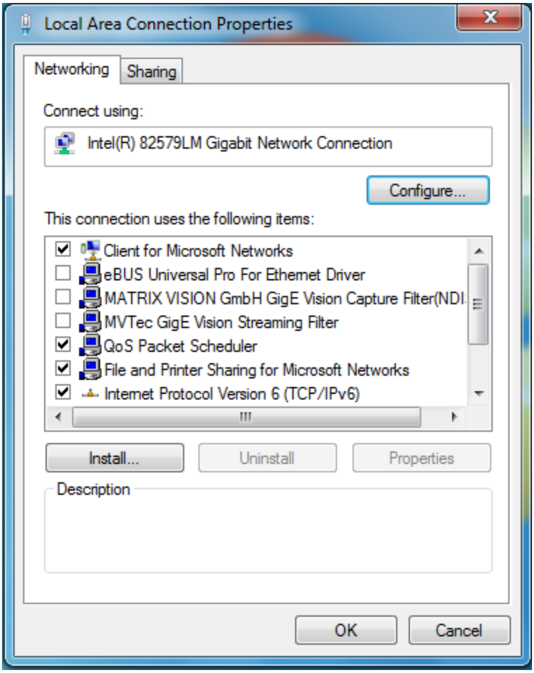


Fig. 86: Local Area Connection Properties

NOTE!
Make sure, that all installed GigE Vision filters/drivers are deactivated.

12.2.3.2 Set up the Network Adapter Features

Make a right click on the network connections you are using to set up the network adapter features. Select “Properties” in the context menu. This will display the network properties (see Fig. 86). Click on “Configure” and then on the tab “Advanced” (see Fig. 87).

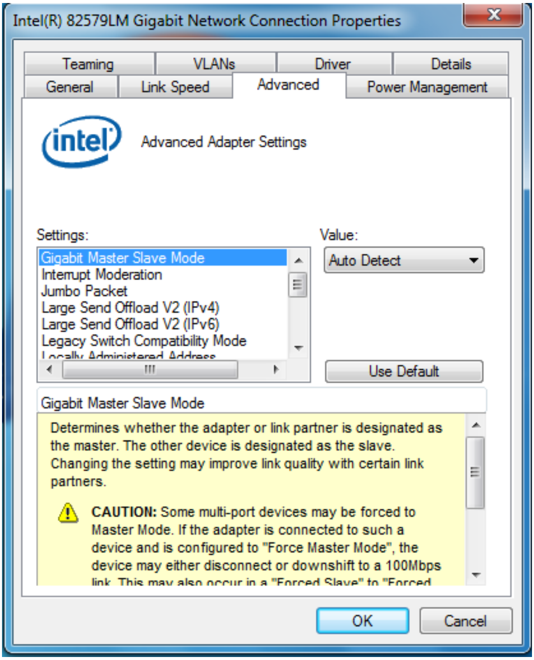


Fig. 87: Gigabit Network Connection Properties

The following features have to be set up:

- Jumbo frames: change it to the highest possible value
- Transmit descriptor (or transmit buffer): change to highest possible value
- Max. IRQ per second: 1000
- Interrupt moderation: On
- Interrupt moderation rate: Extreme

Depending on the network card/network driver it may be possible that not all features mentioned above are available. Please change all those which exist.

12.2.3.3 Deactivating Firewall and Antivirus

In some cases the Windows firewall and antivirus programs tend to block some of the UDP packets sent between the weCat3D GigE Vision interface and the GigE Vision client. It is highly recommended to switch off these programs.

12.2.4 Starting the weCat3D GigE Vision Interface as a Service

Go to the directory in the console window of your operating system where the weCat3D GigE Vision interface is saved and run the interface with the following command:
Syntax: weCat3DGigEInterface.exe -i SCANNER_IP -s SERVER_IP
SCANNER_IP is the IP of the weCat3D profile sensor you want to connect to.
SERVER_IP is the IP for the connection between the interface and the network.

The interface connects to the sensor through the given weCat3D Sensor IP (SCANNER_IP). Once the interface connects to the specified sensor, it reads the sensor's registers and updates the corresponding GigE Vision features. Finally, the interface connects to the network through the given server IP (SERVER_IP). The server IP should match the main IP of the Network Interface Card (NIC).
To get a list of all available IPs from the computer, just start the weCat3D GigE Vision interface without any input arguments or with the input argument (–h).
If the SERVER_IP is not available in the system, the GigE Vision interface will add this IP temporarily to the chosen NIC (see option -n, Fig. 88). The IP will be removed after a system restart. The GigE Vision interface should run with administration rights enabling it to add the IP into the system.

The interface also supports multiple options:

```
C:\weCat3DGigEInterface>weCat3DGigEInterface.exe
Version: 2.0.0

The weCat3DGigeInterface is an application that makes the weCat3D scanners compatible with any GigE compatible client. T
he application is console based. It is Windows and Linux compatible.

Available options:

 -i [SCANNER_IP] the IP address of the weCat3D scanner.

 -s [SERVER_IP] the IP address of the server, to which the application connects.
(Windows) The user defines the interface the server should be connected to (see -n).(Linux) unlike windows, the user in
linux can use any IP to connect the server. The user, however, should add this IP to the system manually

 -n [X] (only windows) the interface index, thorugh which the application connects.
If the server IP given by the option -s is not registered in the system, the server will add this IP to the system
(requires to run the application as system adminstrator).

 -r enable auto connect to the scanner. If this mode is enabled and a disconnected state is detected, the application wi
ll try to reconnect to the scanner for unlimited time. Otherwise the application will end itself

 -d print out in the console debug messages. The debug messages are the comands sent and received from the client.

 -f [FILENAME] print the debug messages into an external file. The debug messages are the commands sent and received fro
m the client.

 -u [USERNAME] set the user defined name in the scanner (Only in FW 1.1.x and higher).

 -p [X] save X scans  into a PCL compatible file format. The point cloud is saved after receiving the StartAcquisition c
ommand from the client.
The new point cloud will overwrite the old one.

 -t [TIMEOUT] set the profile receive timeout. If the weCat3DGigEInterface did not receive a profile from the scanner wi
thin timeout; the application sends the GigE image to the network without waitingthe height of the image to complete. De
fault value for timeout is 1000 [ms]. Set timeout to 0 to disable timeout.

 -w [FILENAME] set the name of the PCL compatible file, if not given; a file with default name (ScanData.txt) is used.

 -h print out the help text.

For more Info, please refer to the weCat3DGigeInterface user manual
```

Fig. 88: weCat3DGigE interface

- -i [SCANNER_IP]: Describes the IP address of the weCat3D Profile Sensor.
- -s [SERVER_IP]: Gives the server's IP address to which the application is connected. Windows users have to define the interface to which the server should be connected (see also option -n). Linux users can use any IP address to connect to the server. This IP must be added to the system manually.
- -n [X]: Defines the NIC's interface where the interface adds the IP, if it is not available in the system. Start the interface with option -h to see the index of your NIC.

NOTE!
This option requires to run the application with administration rights.

- -r: Enables auto reconnect to the weCat3D Profile Sensor in case of a lost connection.
- -d: Prints debug messages in the console. The debug messages are the commands sent and received from the GigE Vision client.
- -f [FILENAME]: Saves the debug messages in an external file.

ATTENTION!
"-d" and "-f" options decrease the performance and the communication time of the weCat3DGigEInterface with the GigE client. It should be used only for debugging purpose. In a normall runnig mode, "-d" and "-f" options should not be used.

- -u [USERNAME]: A specific user defined name. Some GigE Vision client applications (like Halcon) require this name as an extra paramter when they connect to the GigE Vision device.
- -p [X]: Saves X scans into a Point Cloud Library (PCL) compatible file format. The application will start to save the point cloud after receiving the StartAcquisition command from the client. If a new StartAcquisition command is received, the new point cloud will overwrite the old one.
- -t [TIMEOUT]: Sets the profile receive timeout. If the weCat3DGigEInterface did not receive a profile from the sensor within timeout the weCat3DGigEInterface sends the GigE image to the network without waiting for the height of the image to complete. The weCat3DGigEInterface fills the missing scans and chunk data with zeros. Default value for timeout is 1000 ms. Set timeout to 0 to disable timeout.
- -w [FILENAME]: Sets the name of the PCL compatible file. If not given, a file with default name (ScanData.txt) is used.
- -h: Displays a help text on how to use the interface in the console.

12.2.5 Start Multi Instance of the weCat3DGigEInterface

Below is an example of connecting multiple weCat3D profile sensors to multiple instance of weCat3DGigEInterface on the same host (see Fig. 89).
Suppose that the we have two weCat3D sensors, the first has the IP address 192.168.100.1 and the second has the 192.168.100.2
The user can connect both sensors to two different weCat3DGigEInterfaces by starting the weCat3DGigEInterface with option -n (refer to option -n in section 12.2.4):

weCat3DGigEInterface.exe -s 192.168.100.101 -i 192.168.100.1 -n 2 and

weCat3DGigEInterface.exe -s 192.168.100.102 -i 192.168.100.2 -n 2

192.168.100.101 and 192.168.100.102 are temporary SERVER_IPs added by the weCat3DGigEInterface in the operating system, -n 2 is the index of the network interface where the weCat3DGigEInterface adds the new SERVER_IPs. You can detect the index of the network interface you want to add the IPs on it by running the weCat3DGigEInterface without any input arguments (or with -h Option).

```
Below is a list of inteface indeces and their main IPs in this system:
Interface 0  IP: 192.168.56.1
Interface 1  IP: 172.20.112.132
Interface 2  IP: 192.168.100.197
Interface 3  IP: 127.0.0.1
press ENTER to exit.
```

Fig. 89: Indices of available network interfaces

NOTE!
A static IP address should be assigned to the selected network interface.

NOTE!
For this example to work, the user must start the weCat3DGigEInterface in administration mode.

In this case the client recognizes two GigE devices having the IPs 192.168.100.101 and 192.168.100.102.

The temporary SERVER_IPs are deleted after either:

1. Ending the weCat3DGigEInterface by hitting the return button on the terminal where the weCat3DGigEInterface runs or

2. PC restart. Closing the terminal window does not delete the temporary SERVER_IPs and prevent the weCat3DGigEInterface from starting again with the same SERVER_IP.

12.3 GigE Vision Features

The weCat3D GigE Vision interface provides several features organized into different categories:

- Image Format Control
- Acquisition Control
- Digital I/O Control
- Counter And Timer Control
- Encoder Control
- Profile Control
- Device Control
- Scan3dControl
- ChunkDataControl

NOTE!
Some features and categories are only visible in Expert or Guru mode.

12.3.1 Image Format Control

Feature	ComponentSelector
Access Mode	Read/write
Parameter	Intensity/Range
Description	Selects the component to be transmitted in the output frame. • Intensity: The interface sends the intensity values of the scanned profile from the weCat3D profile sensor in Mono10/Mono10Packed pixel format. • Range: The interface sends the 3D points of the computed profile from the weCat3D profile sensor in Mono16 (only supported by external GigE Vision interface) or Coord3D_ABC32f pixel format. **NOTE!** It is recommended to use Coord3D_ABC32f pixel format. Using Mono16 only the full image format should be used and no ROI should be defined (compare command RegionSelector). If Mono16 in PixelFormat is selected, then weCat3DGigEInterface sends a rectified 2.5D image which is suitable for different image processing algorithms. The following equations are needed in order to compute the X/Z coordinates in the coordinate system of the sensor from the rectified image: *Distance Z(i) [mm] = (PixelValue(i) × Scan3dCoordinateScale[CoordinateC] + Scan3dCoordinateOffset[CoordinateC]* *Distance X(i) [mm] = i × Scan3dCoordinateScale[CoordinateA] + Scan3dCoordinateOffset[CoordinateA]* "i" is the position of the pixel (column coordinate in image space) in each row, where each row represents an unique profile. Scan3dCoordinateScale[CoordinateA]: Scale factor of the X-axis Scan3dCoordinateScale[CoordinateC]: Scale factor of the Z-axis Scan3dCoordinateOffset[CoordinateA]: Offset factor of the X-axis Scan3dCoordinateOffset[CoordinateC]: Offset factor of the Z-axis Please refer to the Scan3dControl category for more details. If the value of a pixel at position (i) is zero, then it is invalid. Please note that profile sensors do not provide Y-coordinates. Thus an encoder value could be used to distribute the profiles along the Y-direction. Please refer to the feature ExtraData or to the feature ChunkEncoderValue in ChunkDataControl category. If Coord3D_ABD32f in PixelFormat is selected, then the weCat3DGigEInterface sends the scan data of the profile sensor as a native point cloud format according to the new GigE Vision standard 2.0. The Y-coordinate in this pixel format is computed from the encoder value or from the timestamp value, see Scan3dCoordinateSource feature. Use the features Scan3dCoordinateScale[CoordinateB] and Scan3dCoordinateOffset[CoordinateB] to setup the scale and the offset used to convert the encoder value or timestamp value into mm. Please refer to Scan3dControl category for more details. The advantage of this pixel format is that the GigE Vision client should be able to decode the received data natively into point cloud format without any extra effort from the user side.

Feature	ComponentEnable
Access Mode	Read/write
Parameter	0/1
Description	Enables (1)/disables (0) sending the selected component in the output frame. In embedded GigE Vision interface the "Range" and "Intensity" components are always enabled and can not disabled.

NOTE!
The GigE Vision client software should support the GigE Vision standard 2.1 (Multipart data type) to be able to decode the multi component frame correctly.

Feature	RegionSelector
Access Mode	Read/write
Parameter	Scan3DExtraction0/Region0
Description	If Region0 is selected, the features Width, Height, OffsetX und OffsetY define the size of the ROI on the 2D camera chip in pixels. If Scan3DExtraction0 is selected, the features Width, Height, OffsetX und OffsetY control the size of the output image transfered to the client. RegionSelector implementation is compatible with GeniCam standard naming convention (version 2.4).

Feature	Width		
Access Mode	Read/write*		
Parameter	MLSL: 32...1280 pixel MLWL: 32...2048 pixel	**Default**	MLSL: 1280 MLWL: 2048
Description	If RegionSelector = Region0, it defines the width of the selected ROI in pixel. If RegionSelector = Scan3DExtraction0, it defines the number of points per scan. Per default Width[Scan3DExtraction0] = Width[Region0], unless the feature SignalEnable = First_and_Second where Width[Scan3DExtraction0] = 2 × Width[Region0].		

* Width[Scan3DExtraction0] is read only, Width[Region0] is read/wrote

Feature	Height
Access Mode	Read/write (PC version) Read only (embedded version)
Parameter	The values depend on the Region selected in RegionSelector
Description	If RegionSelector = Region0, it defines the height of the selected ROI in pixel. The size of the ROI affects the laser sensor's scan rate. Value: MLSL: 32...1024 pixel (default = 1024) MLWL: 32...2048 pixel (default = 2048) If RegionSelector = Scan3DExtraction0, it defines the number of profiles to include in each image. Each row of the image represents an unique profile (acquired from the profile sensor). Value: External GigE Vision interface: MLSL / MLWL: 1...10000 (default = 1) Embedded GigE Vision interface: MLSL: 1 (read only) **NOTE!** The interface sends an image only when the number of the received profiles reaches the value of Height[Scan3DExtraction0], or timeout is triggered. For continuous profile monitoring task like a tracking guide system, the number of profiles should be set to a smaller value (1) to enable a continuous image (profile) transfer.

Feature	OffsetX		
Access Mode	Read/write		
Parameter	MLSL: 0...1279 pixel MLWL: 0...2047 pixel	**Default**	0
Description	If RegionSelector = Region0, it defines the Offset in X of ROI in pixels. If RegionSelector = Scan3DExtraction0: Read only (not used).		

Feature	OffsetY		
Access Mode	Read/write		
Parameter	MLSL: 0...1023 pixel MLWL: 0...2047 pixel	**Default**	0
Description	If RegionSelector = Region0, it defines the Offset in Y of ROI in pixels. If RegionSelector = Scan3DExtraction0: Read only (not used).		

The features Height[Region0], Width[Region0], OffsetX[Region0] and OffsetY[Region0] define the size of the ROI in pixels for the internal camera. The min. and max. values for those features depend on hardware. Adjusting the ROI has an influence on the max. measurement rate.
The feature Width[Region0] sets the number of points in X direction. Reducing the value of Width[Region0] decreases the capture area in X direction and hence, reduces the amount of read out pixels from the sensor. This allows to increase the measuring rate and reduce the network load. The feature value "Width" will be updated automatically to match the value of the feature Width[Region0] (to keep the width of the image equal to the number of output points per profile of the sensor).

The Height[Region0] feature sets the capture area for the internal camera of the sensor in Z (Y in camera coordinate system) direction. Reducing the Height[Region0] feature decreases the working range of the sensor, but increases the measurement rate.
The features OffsetX[Region0] and OffsetY[Region0] define the start position of the capture area for the sensor's camera in X and Z (in camera coordinate system) direction respectively.

Feature	PixelFormat
Access Mode	Read/write
Parameter	Mono10/Mono10Packed/Mono16/Coord3D_ABC32f
Description	This command defines the type of image sent to the client. • Mono10/Mono10Packed: Only available if Intensity component is selected. • Mono16: Only available if Range component is selected. Only available in the external GigE Vision interface. • Coord3D_ABC32f: Only available if Range component is selected.

Feature	PayloadSize
Access Mode	Read only
Description	Returns the size of the expected image payload in bytes. Depends on the features Width, Height und PixelFormat.

Feature	SensorWidth
Access Mode	Read only
Description	Delivers the effective width of the integrated camera in pixel (MLSL: 1024 pixels; MLWL: 2048 pixels).

Feature	SensorHeight
Access Mode	Read only
Description	Delivers the effective height of the integrated camera in pixel (MLSL: 1280 pixels; MLWL: 2048 pixels).

Feature	Scan3DSortX		
Access Mode	Read/write		
Parameter	On Off	**Default**	Off
Description	Switch the sorting function of the point cloud on/off with respect to X values (ascending). Normally, output data from the sensor are already sorted with respect to X. In some extreme cases it could happen that output data are not sorted due to the calibration process and the extreme orientation of the surface in front of the sensor **NOTE!** Switching the feature to "ON" could reduce the performance of the weCat3D GigE Vision interface. Only available in the external GigE Vision interface..		

12.3.2 Acquisition Control

Feature	AcquisitionMode		
Access Mode	Read/write		
Parameter	Continous/SingleFrame		
Description	Defines the acquisition mode of the interface: in continuous mode, once the interface receives the command StartAcquisition from the client, it keeps sending images until the client sends a StopAcquisition command. In SingleFrame mode, the interface sends only one image per StartAcquisition command and the client does not need to send a StopAcquisition command.		

Feature	AcquisitionLineRate		
Access Mode	Read/write		
Parameter	MLSL: 10...4000 Hz (ab Firmware Version 1.0.10: 1...4000 Hz) MLWL: 10...6000 Hz (ab Firmware Version 1.0.10: 1...6000 Hz)	**Default**	MLSL: 200 MLWL: 175
Description	Defines the measurement rate of the sensor in Hz (i.e. number of measured profiles per second). The maximum value for the feature is hardware dependent, and it depends on the value of the feature ExposureTime as well as the size of the active ROI. Please refer to section 7 and the features to set up the ROI in this documentation. The AcquisitionLineRate is only considered if the LineStart trigger mode is Off (the sensor is in Intern trigger mode). The measurement rate of the sensor in other trigger modes depends on input signals and value of TriggerDivider. **NOTE!** If the AcuisitionLineRate is set to a large value without taking into consideration the size of the ROI, the sensor will fail to send reliable profiles and the bit5 in ChunkScannerState is set to 1.		

Feature	ResultingAcquisitionLineRate		
Access Mode	Read only		
Description	Shows the actual measurement rate of the sensor, updates every 1 second.		

Feature	ExposureTime		
Access Mode	Read/write		
Parameter	0...100 000	**Default**	150
Description	Sets the exposure time of the integrated camera in μs. For further information see section 10.5.2.		

The allowed values for AcquisitionLineRate and ExposureTime depend on each other. The allowed value for AcquisitionLineRate and ExposureTime should hold for the following equation:

$$1000000 \times (1 / AcquisitionLineRate) \geq ExposureTime + 40\ (\mu s)$$

Feature	TriggerSelector
Access Mode	Read/write
Parameter	LineStart/FrameStart/AcquisitionActive
Description	Selects the trigger function to configure. • LineStart: Defines the trigger settings of the sensor for generating a profile. • FrameStart: Defines the sensor‘s trigger settings in the fixed mode. • AcquisitionActive: Acts as a global trigger enable/disable.

A frame is defined as an image where each row of the image represents a scanned profile. The trigger source for generating a profile is defined in TriggerSource (after selecting “LineStart” in the TriggerSelector feature).
If the “FrameStart” trigger function is off, the weCat3D GigE interface sends continuous frames, taking into consideration the trigger source in “LineStart” function.
If the “FrameStart” trigger function is on, then the weCat3D GigE interface will send a frame only after receiving a new trigger signal as defined in TriggerSource (after selecting “FrameStart” in TriggerSelector).
For further information refer to “fixed mode” in section 7.2.4.

ATTENTION!
It is not possible to select the same trigger source in “LineStart” and “FrameStart”.

The “AcquisitionActive” trigger function is used as a global trigger enable function. If “AcquisitionActive” mode is on, the weCat3D sensor will generate profiles only when the selected line in TriggerSource (after selecting the “AcqusitionActive” trigger function in TriggerSelector) is active.
For further information refer to “Profile enable” pin function in chapter “I/O Settings” in section 7.2.4.

NOTE!
It is recommended to use the “AcquisitionActive” trigger function only when Height[Scan3DExtraction0] is set to 1, since it is difficult to synchronize the duration of the “AcquistionActive” signal with the end of the frame.

ATTENTION!
It is not possible to activate both “FrameStart” and “AcquisitionActive” since both modes share the same sensor resources.

Following some examples with timechart to illustrate the relationship between “LineStart”, “FrameStart” and “AcquisitionActive”.

Example 1 (see Fig. 90):
Height[Scan3DExtraction0] = 5
TriggerSelector = LineStart
TriggerSource = Off, Line, Encoder or Software
TriggerSelector = FrameStart
TriggerMode = Off
TriggerSelector = AcquisitionActive
TriggerMode = Off

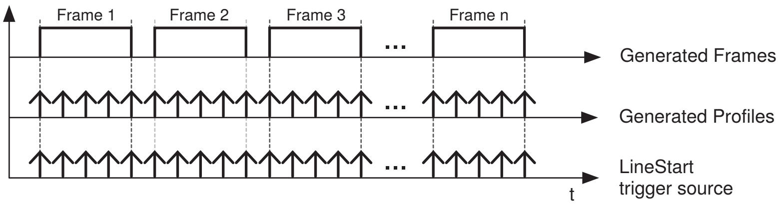


Fig. 90: Example 1

Example 2 (see Fig. 91):
Height[Scan3DExtraction0] = 5
TriggerSelector = LineStart
TriggerSource = Off, Line, Encoder oderSoftware
TriggerSelector = FrameStart
TriggerMode = On
TriggerSource = Line1
TriggerSelector = AcquisitionActive
TriggerMode = Off

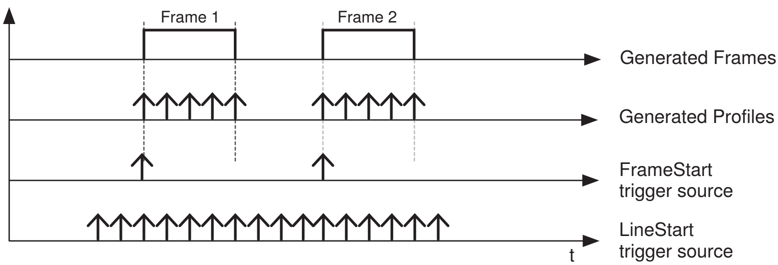


Fig. 91: Example 2

Example 3 (see Fig. 92):
Height[Scan3DExtraction0] = 1
TriggerSelector = LineStart
TriggerSource = Off, Line, Encoder or Software
TriggerSelector = FrameStart
TriggerMode = Off
TriggerSelector = AcquisitionActive
TriggerMode = On
TriggerSource = Line3

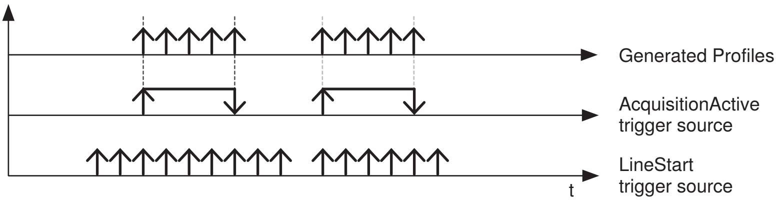


Fig. 92: Example 3

Feature	TriggerMode
Access Mode	Read/write
Parameter	On/Off
Description	TriggerMode switches the trigger function selected in the feature TriggerSelector on or off. If LineStart is selected: • **Off:** The sensor is in Intern trigger mode. The sensor generates profiles according to the value set in feature AcquisitionLineRate. • **On:** The sensor generates profiles from the source defined in TriggerSource. If FrameStart is selected: • **Off:** The sensor is in dynamic mode, the frames are sent without wating for frame trigger signal. • **On:** The sensor is in FixedFrame mode. The sensor sends a frame only when it receives a trigger signal definded in TriggerSource. If AcquisitionActive is selected: • **On:** The global enable is activated. • Off: Global disable is deactivated. **NOTE!** It is not possible to switch on “FrameStart” and “AcquisitionActive” at the same time, because both trigger functions share some of the sensor‘s resources.

Feature	TriggerSource
Access Mode	Read/write
Parameter	Line1/Line2/Line3/Line4/Encoder1/Encoder2/Software
Description	Line1...Line4: Depending on the trigger function selected in TriggerSelector feature: With “LineStart” function the sensor will generate a profile only when the selected line is activated (SyncIn mode). With “FrameStart” function the sensor will send one frame each time the selected line is activated (fixed mode). With “AcquisitionActive” function the sensor will generate profiles as long as the selected line is active (ProfileEnable mode). Encoder1: Only available in “LineStart” trigger function. The sensor will generate a profile with each HTL encoder step. The HTL encoder should be connected to the profile sensor and activated. Please find more information how to connect an encoder in section 7.2. Encoder2: Only available in “LineStart” trigger function. The sensor will generate a profile with each TTL encoder step. The TTL encoder should be connected to the sensor and activated. Please find more information how to connect an encoder in section 7.2. Software: Available in “LineStart” and “FrameStart” trigger functions. With “LineStart” function the sensor scans one profile each time a trigger software command is received. With “FrameStart” function the sensor sends one frame each time a trigger software command is received. **NOTE!** It is not possible to select “Software” as a trigger source for both “LineStart” and “FrameStart” at the same time.

Feature	TriggerActivation
Access Mode	Read/write
Parameter	RisingEdge/FallingEdge/LevelHigh/LevelLow
Description	Only available if the selected TriggerSource is Line1...Line4. It defines the activation signal. “RisingEdge” and “FallingEdge” are available for the trigger functions “LineStart” and “FrameStart”. “LevelHigh” and “LevelLow” are available only for “AcquisitionActive” trigger function.

Feature	TriggerDelay
Access Mode	Read/write
Parameter	0...1 000 000
Description	Sets a constant delay for profile acquisition in μs (only available at intern trigger source). The allowed value depends on the values of AcquisitionLineRate and ExposureTime.

Feature	TriggerDivider
Access Mode	Read/write
Parameter	0...65535
Description	Sets the division factor for the external trigger source. This feature is available in “Encoder” and “Line1...Line4” trigger source. For example, if the TriggerDivider is set to 150, the sensor triggers on the encoder values 150, 300, 450 and so on.

Feature	TriggerSoftware
Access Mode	Command button
Description	Sends a software trigger command to weCat3D Profile Sensor.

Feature	AcquisitionStart
Access Mode	Command button
Description	The command starts the acquisition of profiles (image) from the weCat3D Profile Sensor.

Feature	AcquisitionStop
Access Mode	Command button
Description	The command stops the acquisition of profiles (image) from the weCat3D Profile Sensor.

12.3.3 Digital I/O Control (E/A)

Feature	SyncOut (deprecated)		
Access Mode	Read/write		
Parameter	0...100 000	Default	0
Description	This command sets the signal‘s width on “SyncOut” pin in μs (see section 7.2.4).		

Feature	SyncOutDelay (deprecated)		
Access Mode	Read/write		
Parameter	0...100 000	Default	0
Description	This command sets a constant delay between the sensor‘s trigger and the “SyncOut” signal in μs (see section 7.2.3).		

NOTE!
The features SyncOut and SyncOutDelay have been deprecated since they are not defined in the GenICam Standard Features Naming Conversions (SFNC). The features are set to invisible. Old programs can still use these two features. Both have been replaced by other features (see section 12.3.4). It is highly recommended to use the new defined features which apply to the GenICam SFNC.

Feature	LineSelector
Access Mode	Read/write
Parameter	Line1/Line2/Line3/Line4
Description	Selects the user I/O to be configured.

Feature	LineMode
Access Mode	Read/write
Parameter	Input/Output
Description	Defines the selected line as input or output.

Feature	LineInverter
Access Mode	Read/write
Parameter	True/False
Description	Controls the signal's inversion of the selected line. Only available, if the selected line is "Input".

Feature	LineStatus
Access Mode	Read only
Parameter	True/False
Description	Displays the current status of the selected line.

Feature	LineStatusAll
Access Mode	Read only
Parameter	Integer
Description	Returns the current status of all available I/O signals at time of polling in a single bitfield. Values: bit 0: I/O1 bit 1: I/O2 bit 2: I/O3 bit 3: I/O4

Feature	LineSource
Access Mode	Read/write
Parameter	UserOutput/Timer1Active
Description	Defines the control of ouput signal for the selected line. Only available, if the selected line is "Output". • "UserOutput": allows the user to activate the output signal manually (see the feature UserOutputValue). • "Time1Active": The output signal is activated by Timer1, see CounterAndTimerControl.

Feature	UserOutputValue
Access Mode	Read/write
Parameter	True/False
Description	Activates the signal of the selected line. The selected line should be “Output” and the LineSource feature value should be “UserOutput”.

Feature	UserOutputValueAll
Access Mode	Read/write
Parameter	Integer
Description	Sets the value of all bits of the user output register.

Feature	UserOutputValueAllMask
Access Mode	Read/write
Parameter	Integer
Description	Sets the write mask to apply it to the value specified by UserOutputValueAll before writing it in the user output register.

Feature	OutputFunction
Access Mode	Read/write
Parameter	Push_Pull/PNP/NPN
Description	Controls the current electrical format of the selected line. Only available, if the selected line is “Output”.

Feature	InputLoad
Access Mode	Read/write
Parameter	True/False
Description	Controls the current electrical format of the selected user line. Only available, if the selected line is “Input”.

12.3.4 Counter und Timer Control

Feature	TimerSelector
Access Mode	Read/write
Parameter	Timer1
Description	Selects the timer to be configured (corresponds to SyncOut, please see example below).

Feature	TimerTriggerSource
Access Mode	Read/write
Parameter	LineTrigger
Description	Selects the trigger source for starting the timer.

Feature	TimeDuration
Access Mode	Read/write
Parameter	0...100 000
Description	Defines the active signal's duration (in μs) of the timer.

Feature	TimerDelay
Access Mode	Read/write
Parameter	0...100 000
Description	Defines the delay (in μs) between triggering and activating the timer.

The following example illustrates how to set up E/A4 to SyncOut function
(sensor settings: Signal Width = 1000 μs and SyncOutDelay = 5000 μs):

in DigitalIOControl:
LineSelector = Line4
LineMode = Output
LineSource = Timer1Active

in CounterAndTimerControl:
TimerSelector = Timer1
TimerTriggerSource = LineTrigger
TimerDuration = 1000 µs
TimerDelay = 5000 µs

12.3.5 Encoder Control

It is possible to connect two types of encoders to the profile sensor: HTL encoder through E/A1 and E/A2 and TTL encoder through the specific input pins. Only one encoder can trigger the sensor in “LineStart” function. Please refer to section 7.2 on how to connect and activate the encoders.

Feature	EncoderSelector
Access Mode	Read/write
Parameter	Encoder1/Encoder2
Description	Selects the encoder to be configured. Encoder 1 refers to HTL encoder and encoder 2 refers to TTL encoder.

Feature	EncoderSourceA
Access Mode	Read/write
Parameter	Line1/Off
Description	Selects the input line for encoder A signal. Only available for Encoder1 (HTL).

Feature	EncoderSourceB
Access Mode	Read/write
Parameter	Line2/Off
Description	Selects the input line for encoder B signal. Only available for Encoder1 (HTL).

NOTE!
If the features EncoderSourceA or EncoderSourceB are switched off, the HTL encoder is deactivated and the sensor can be triggered through TTL encoder.

Feature	EncoderOutputMode
Access Mode	Read/write
Parameter	PositionUp/PositionDown/DirectionUp/DirectionDown/Motion
Description	PositionUp: The encoder triggers the sensor only in one direction (counting up) and only if the new encoder value is higher than the highest last value (see section 7.2.4). Same as “PositionUp”, but in opposite direction. PositionDown: The encoder triggers the sensor only in one direction without considering DirectionUp: the last position. Same as „DirectionUp“, but in opposite direction. DirectionDown: The encoder triggers the sensor in each direction (counting up or down). Motion:

Feature	EncoderResetSource
Access Mode	Read/write
Parameter	Line1/Line2/Line3/Line4/Off
Description	Selects the signals to reset both encoders.

Feature	EncoderResetActivation
Access Mode	Read/write
Parameter	AnyEdge/RisingEdge/FailingEdge
Description	Selects the activation mode of the EncoderResetSource signal.

Feature	EncoderReset
Access Mode	Command button
Parameter	Encoder1/Encoder2
Description	Software command to reset both encoders.

Feature	EncoderValue
Access Mode	Read only
Parameter	Encoder specific
Description	Displays the encoder value of the selected encoder.

12.3.6 Profile Control

Feature	SignalEnable		
Access Mode	Read/write		
Parameter	First/Second/First_and_Second	**Default**	First
Description	The command sets the number of profiles given out per position. For further information see section 7.2.3.		

Feature	SignalSelection		
Access Mode	Read/write		
Parameter	Top/Strength/Signal Width/Bottom	**Default**	Strength
Description	The command sorts the peaks that are received by the internal camera according to one of the listed criteria. There are four criteria for peak sorting in SignalSelection: peak 1, intensity, width and peak 2. For example: if the sorting criteria in SignalSelection is set to peak 1, then the sensor will sort the peaks based on their position on the camera chip. For further information see section 7.2.3.		

Feature	SignalWidthMin		
Access Mode	Read/write		
Parameter	0...63 pixels	**Default**	0
Description	This is a filter to define the minimum peak width for the evaluation in pixels.		

Feature	SignalWidthMax		
Access Mode	Read/write		
Parameter	0...63 pixels	**Default**	63
Description	This is a filter to define the maximum peak width for the evaluation in pixels.		

Feature	SignalStrengthMin		
Access Mode	Read/write		
Parameter	0...1023	Default	0
Description	Defines the minimum signal strength to evaluate the signal.		

The features SignalWidthMax and SignalWidthMin define the peak maximum and minimum width in pixels for evaluation, while SignalStrengthMin defines the signal's minimum strength.

12.3.7 Device Control

Most of the features in the Device Control category are for displaying information about the connected sensor, thus they are hardware dependent.

Feature	DeviceType
Access Mode	Read only
Response	Transmitter
Description	General info about the device

Feature	DeviceModelName
Access Mode	Read only
Response	Sensor specific
Description	General info about the device

Feature	DeviceVendorName
Access Mode	Read only
Response	wenglor sensoric GmbH
Description	Name of the device vendor

Feature	DeviceVersion
Access Mode	Read only
Response	Sensor specific
Description	General Info about the device

Feature	DeviceFirmwareVersion
Access Mode	Read only
Response	Sensor specific
Description	General info about the device

Feature	DeviceSerialNumber
Access Mode	Read only
Response	Sensor specific
Description	General Info about the device

Feature	DeviceTLType
Access Mode	Read only
Response	GigEVision
Description	General Info about the transport layer type of the device.

Feature	DeviceTemperatureSelector
Access Mode	Read/write
Parameter	CPU
Description	Selects the location of the device where the temperature is measured.

Feature	DeviceTemperature
Access Mode	Read only
Response	Location specific
Description	Device temperature at the selected location in degrees Celsius (°C).

Feature	DeviceReset
Access Mode	Command button
Description	Reset of the device to its power up state. **NOTE!** For updating all feature values after a reset command, the GigE Vision client should be disconnected from the profile sensor and reconnected again.

Feature	AsciiCommand
Access Mode	Write only (string)
Description	Sends an ASCII command to the weCat3D profile sensor. List of supported ASCII commands are summarized below. A detailed description is given in section 10.7. Sending an ASCII command is recommended only if the feature is not directly implemented in the GigE Vision feature tree.

NOTE!
Sending ASCII commands could result in a defined behaviour. ASCII commands should be sent only in StopAcquisition mode. Further information to ASCII commands see section 10.7).

ASCII Commands:

SetExposureTime=x	SetSyncOut=x
SetAutoExposureMode=x	SetSyncOutDelay=x
SetAutoExposureTimeMin=x	SetSignalEnable=x
SetAutoExposureTimeMax=x	SetSignalWidthMin=x
SetAutoExposureIntensityRangeMin=x	SetSignalWidthMax=x
SetAutoExposureIntensityRangeMax=x	SetSignalSelection=x
SetAutoExposureRangeXMin=x	SetLinearizationMode=x
SetAutoExposureRangeXMax=x	SetEncoderCountDirection=x
SetAcquisitionLineTime=x	SetROI1WidthX=x
SetHDR=x	SetROI1OffsetX=x
SetExposureTime2=x	SetROI1StepX=x
SetLaserDeactivated=x	SetROI1HeightZ=x
SetUserLED=x	SetROI1OffsetZ=x
SetSignalContentZ=x	SetROI1StepZ=x
SetSignalContentStrength=x	SetEA1Function=x
SetSignalContentWidth=x	SetEA1FunctionLaserOff=x
SetSignalContentReserved=x	SetEA1FunctionProfileEnable=x
SetSocketConnectionTimeout=x	SetEA1FunctionResetCounter=x
SetHeartBeat=x	SetEA1ResetCounterRepeat=x
SetResetEncoder\r	SetEA1ResetCounterSignaledge=x
SetResetPictureCounter	SetEA1ResetCounterBaseTimeCounter=x
SetSettingsSave=x	SetEA1ResetCounterPictureCounter=x
SetResetBaseTime	SetEA1ResetCounterEncoderHTL=x
SetSettingsLoad=x	SetEA1ResetCounterEncoderTTLRS422=x
SetTriggerSource=x	SetEA1InputFunction=x
SetTriggerEncoderStep=x	SetEA1InputLoad=x
SetTriggerDelay=x	SetEA1Output=x
SetEncoderTriggerFunction=x	SetEA1OutputFunction=x
SetTriggerAmountProfilesY=x	SetEA1FunctionInputCounter=x
SetAmountProfilesY=x	

12.3.8 Scan3dControl

Feature	Scan3dCoordinateSelector
Access Mode	Read/write
Response	CoordinateA/CoordinateB/CoordinateC
Description	Selects the idividual axis for 3D information/transformation. CoordinateA is for X axis CoordinateB is for encoder (Y) axis CoordinateC is for Z axis

Feature	Scan3dCoordinateScale
Access Mode	Read/write (see below)
Description	Scale factor used to transfrom a pixel value (in Mono16 images) or Encoder/Timestamp value (if CoordinateB is selected) into mm coordinates. The access mode and the value of the feature is updated according to the selected values in PixelFormat and Scan3dCoordinateSelector as follows: Coord3D_ABC32f: Scan3dCoordinateSelector = CoordinateA Scan3dCoordinateScale = 1 (read only) Scan3dCoordinateSelector = CoordinateB Scan3dCoordinateScale = 1 (read/write) Scan3dCoordinateSelector = CoordinateC Scan3dCoordinateScale = 1 (read only) Mono16: Scan3dCoordinateSelector = CoordinateA Scan3dCoordinateScale = Device specific (read/write) Scan3dCoordinateSelector = CoordinateB Scan3dCoordinateScale = 1 (read/write) Scan3dCoordinateSelector = CoordinateC Scan3dCoordinateScale = Device specific (read only)

There are two ways to compute the Y coordinates (Coordinate B) in the weCat3DGigeInterface: Using the Encoder or using the Timestamp, see feature Scan3dCoordinateSource.

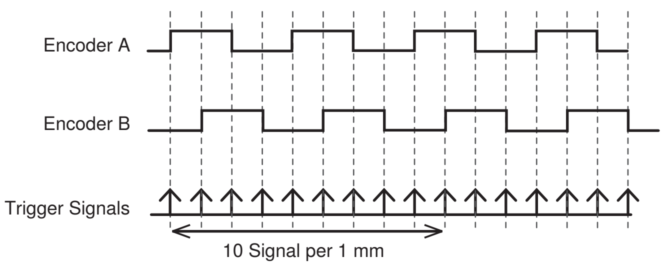


Fig. 93: Encoder trigger signals

Example:

If Scan3dCoordinateSource = Encoder and encoder trigger is 10 signals per 1 mm, then
Scan3dCoordinateScale = 1/10 = 0.1

If Scan3dCoordinateSource = Timestamp and the linear speed of the conveyor belt is 10 [mm/s] then
Scan3dCoordinateScale = 10 (speed of conveyor belt) * 10^-6 (convert from μs to s) = 0.00001 [mm/μs]

Feature	Scan3dCoordinateOffset
Access Mode	Read/write (see below)
Description	Offset factor, used to transform a pixel value (in Mono16/RGB16/RGB16Planar images) or encoder value (if CoordinateB is selected) into mm coordinates. The access mode and the value of the feature is updated according to the selected values in PixelFormat and Scan3dCoordinateSelector as follows: Coord3D_ABC32f: Scan3dCoordinateSelector = CoordinateA Scan3dCoordinateOffset = 0 (read only) Scan3dCoordinateSelector = CoordinateB Scan3dCoordinateOffset = 0 (read/write) Scan3dCoordinateSelector = CoordinateC Scan3dCoordinateOffset = 0 (read only) Mono16: Scan3dCoordinateSelector = CoordinateA Scan3dCoordinateOffset = Device specific (read/write) Scan3dCoordinateSelector = CoordinateB Scan3dCoordinateOffset = 0 (read/write) Scan3dCoordinateSelector = CoordinateC Scan3dCoordinateOffset = Device specific (read only)

Feature	Scan3dCoordinateSource
Access Mode	Read/write
Response	Encoder/Timestamp
Description	This feature is only available when CoordinateB in Scan3dCoordinateSelector is selected. This feature defines the source to compute the Y coordinate for the Range component of the image in Coord3D_ABD32f pixel format.

Feature	Scan3dInvalidDataFlag
Access Mode	Read only
Response	True/false
Description	Enables the definition of a non-valid point flag value in the received data. The flag is enabled only in CoordinateC (Z axis). The feature is not available if CoordinateB is selected.

Feature	Scan3dInvalidDataValue

Access Mode	Read only
Description	The value which indentifies a non-valid pixel/point if Scan3dInvalidDataFlag is enabled. The flag is only enabled in CoordinateC (Z axis) and the value is 0. The feature is not available if CoordinateB is selected.

Feature	Scan3dAxisMin
Access Mode	Read only
Description	The minimum valid transmitted coordinate value of the selected axis. The feature is not available if CoordinateB is selected.

Feature	Scan3dAxisMax
Access Mode	Read only
Description	The maximum valid transmitted coordinate value of the selected axis. The feature is not available if CoordinateB is selected.

12.3.9 ChunkDataControl

Use ChunkData instead of ExtraData to get data (like encoder value or state of the E/A's) related to each scan line (row) in the received image.
ChunkData is implemented according to the latest GigE Vision standard introduced in version 2.0.
Use the feature ChunkScanLineSelector to read the chunk value from a specific scan line in the received image.

Feature	ChunkSelector
Access Mode	Read/write
Response	ChunkPictureCounter / ChunkTimestamp / ChunkLineStatusAll / ChunkEncoderValue / ChunkScannerStatus
Description	Selects the chunk feature to enable or control.

Feature	ChunkEnable
Access Mode	Read/write
Response	True/false
Description	Enables/disables the selected chunk feature to be transmitted with data stream.

Feature	ChunkScanLineSelector
Access Mode	Read/write
Description	Index for vector representation of one chunk value per line in the received data.

Feature	ChunkPictureCounter
Access Mode	Read only
Description	Returns the value of the picture counter of the selected line (row) in the received data Chun kPictureCounter[ChunkScanLineSelector].

Feature	ChunkTimestamp
Access Mode	Read only
Description	Returns the value of the timestamp of the selected line (row) in the received data ChunkTimestamp[ChunkScanLineSelector].

Feature	ChunkDeviceTemperature
Access Mode	Read only
Description	Returns the value of the temperature of the selected line (row) in the received data ChunkTemperature[ChunkScanLineSelector].

Feature	ChunkLineStatusAll
Access Mode	Read only
Description	Returns the value of the LineStatusAll of the selected line (row) in the received data ChunkLineStatusAll[ChunkScanLineSelector]. The ChunkLineStatusAll encodes the state of all lines E/A1...E/A4 at the time of generating the profile as follows: bit0: E/A1 Status bit1: E/A2 Status bit2: E/A3 Status bit3: E/A4 Status

Feature	ChunkEncoderValue
Access Mode	Read only
Description	Returns the value of the activated encoder value of the selected line (row) in the received data ChunkEncoderValue[ChunkScanLineSelector].

Feature	ChunkScannerStatus
Access Mode	Read only
Description	Returns the value of the sensor state of the selected line (row) in the received data ChunkScannerState[ChunkScanLineSelector].The ChunkScannerState encodes the state of the profile sensor at the time of generating the profile as follows: bit0: Ready (0=NOK; 1=OK) bit1: ExposureTime (0=NOK; 1=OK) bit2: LaserONTime (0=NOK; 1=OK) bit3: Not in use bit4: Not in use bit5: Measurement rate too high (0=NOK; 1=too high) bit6: Not in use bit7: Not in use

12.4 Troubleshooting

12.4.1 Connection Broken

In the case that the connection between the weCat3D GigE Vision interface (service) and the sensor is broken, the interface sends an Event message (Event No.: 10) with the error message: connection to the sensor is lost. After that, the interface service will close itself automatically.

12.4.2 No Connection to the Sensor

In the case that the weCat3D GigE Vision interface fails to connect to the sensor, the interface will not start and the GigE Vision server will not be available online! This case happens when the sensor is already connected to other clients or when the interface failed to receive data from the sensor.
It is possible to check the connection status of the profile sensor by the web interface, please see section 7.

12.4.3 The weCat3D GigE Interface Is Not Available Online

If the given server IP is not available in the system, the interface will fail to go online. As a result, the client can not detect the interface (to see a list of available IPs just start the weCat3D GigE interface without any input parameters or with the input argument “-h”).
Possible solutions are either to add the IP address manually into the operating system or to start the application with the option “-n” (see section 12.2.4).

12.4.4 The Sensor Triggers Too Fast

The bit5 in sensor status is set to 1 (see the ExtraData feature) and the feature AcquisitionStatus will be “TooFast”.

12.4.5 The sensor Sends Profiles Faster than the Network Can Handle

This happens when the network card is not compatible with GigE Vision or the network rate is 100 Mbit instead of 1 Gbit. This results in loosing some of the profiles (see section 12.2.3 on how to set up the network adapter).

12.4.6 The client does not receive images

If the weCat3D GigE interface is located and running on the same machine where the GigE client software is running, then in some cases it is necessary to deactivate the additional GigE Vision driver of the client software directly inside the connection settings of the active Local Area Connection.

13. Maintenance Instructions

NOTE!

- This sensor does not require any cyclical recalibration.
- Cleaning of both lens covers at regular intervals is recommended in order to assure uniform good quality of the measured values. A commercially available cloth for cleaning eyeglasses can be used for this purpose.
- Do not clean the sensor with solvents or cleansers which could damage the product.

14. Proper Disposal

wenglor sensoric GmbH does not accept the return of unusable or irreparable products. Respectively valid national waste disposal regulations apply to product disposal.

15. EU Declaration of Conformity

The EU declaration of conformity can be found on our website at www.wenglor.com in the product's separate download area.

wenglor sensoric GmbH
wenglor Straße 3
88069 Tettnang
GERMANY

Phone: +49 7542 5399-0
info@wenglor.com

For further wenglor contacts please see www.wenglor.com