# SmartRunner 3-D Stereo

# 3-D Stereo Sensor

Manual

Your automation, our passion.

PEPPERL+FUCHS

With regard to the supply of products, the current issue of the following document is applicable: The General Terms of Delivery for Products and Services of the Electrical Industry, published by the Central Association of the Electrical Industry (Zentralverband Elektrotechnik und Elektroindustrie (ZVEI) e.V.) in its most recent version as well as the supplementary clause: "Expanded reservation of proprietorship"

**Worldwide**

Pepperl+Fuchs Group
Lilienthalstr. 200
68307 Mannheim
Germany
Phone: +49 621 776 - 0
E-mail: info@de.pepperl-fuchs.com

**North American Headquarters**

Pepperl+Fuchs Inc.
1600 Enterprise Parkway
Twinsburg, Ohio 44087
USA
Phone: +1 330 425-3555
E-mail: sales@us.pepperl-fuchs.com

**Asia Headquarters**

Pepperl+Fuchs Pte. Ltd.
P+F Building
18 Ayer Rajah Crescent
Singapore 139942
Phone: +65 6779-9091
E-mail: sales@sg.pepperl-fuchs.com
https://www.pepperl-fuchs.com

# 1 Introduction

## 1.1 Content of this Document

This document contains information required to use the product in the relevant phases of the product life cycle. This may include information on the following:

- Product identification
- Delivery, transport, and storage
- Mounting and installation
- Commissioning and operation
- Maintenance and repair
- Troubleshooting
- Dismounting
- Disposal

**Note**

For full information on the product, refer to the further documentation on the Internet at www.pepperl-fuchs.com.

**Note**

For specific device information such as the year of construction, scan the QR code on the device. As an alternative, enter the serial number in the serial number search at www.pepperl-fuchs.com.

The documentation comprises the following parts:

- This document
- Datasheet

In addition, the documentation may comprise the following parts, if applicable:

- EU-type examination certificate
- EU declaration of conformity
- Attestation of conformity
- Certificates
- Control drawings
- Instruction manual
- Functional safety manual
- Other documents

## 1.2 Target Group, Personnel

Responsibility for planning, assembly, commissioning, operation, maintenance, and dismounting lies with the plant operator.

Only appropriately trained and qualified personnel may carry out mounting, installation, commissioning, operation, maintenance, and dismounting of the product. The personnel must have read and understood the instruction manual and the further documentation.

Prior to using the product make yourself familiar with it. Read the document carefully.

## 1.3 Symbols Used

This document contains symbols for the identification of warning messages and of informative messages.

### Warning Messages

You will find warning messages, whenever dangers may arise from your actions. It is mandatory that you observe these warning messages for your personal safety and in order to avoid property damage.

Depending on the risk level, the warning messages are displayed in descending order as follows:

**Danger!**

This symbol indicates an imminent danger.

Non-observance will result in personal injury or death.

**Warning!**

This symbol indicates a possible fault or danger.

Non-observance may cause personal injury or serious property damage.

**Caution!**

This symbol indicates a possible fault.

Non-observance could interrupt the device and any connected systems and plants, or result in their complete failure.

### Informative Symbols

**Note**

This symbol brings important information to your attention.

### Action

1. This symbol indicates a paragraph with instructions. You are prompted to perform an action or a sequence of actions.

# 2 Product Specifications

## 2.1 Use and Application

This manual describes the "SmartRunner Explorer 3-D" 3-D stereo sensor (referred to as the sensor below). The sensor combines two 2-D images taken from different positions and finds matches between the acquired images to create a depth image.

A bright projector, which sits between the two cameras, projects a structured light onto the object to be imaged. The auxiliary structures of the structured light enable a higher number of pixels and their changes in position to be detected on the surface of the object so that complete, homogeneous depth information of the scene can be generated from it.

### How It Works

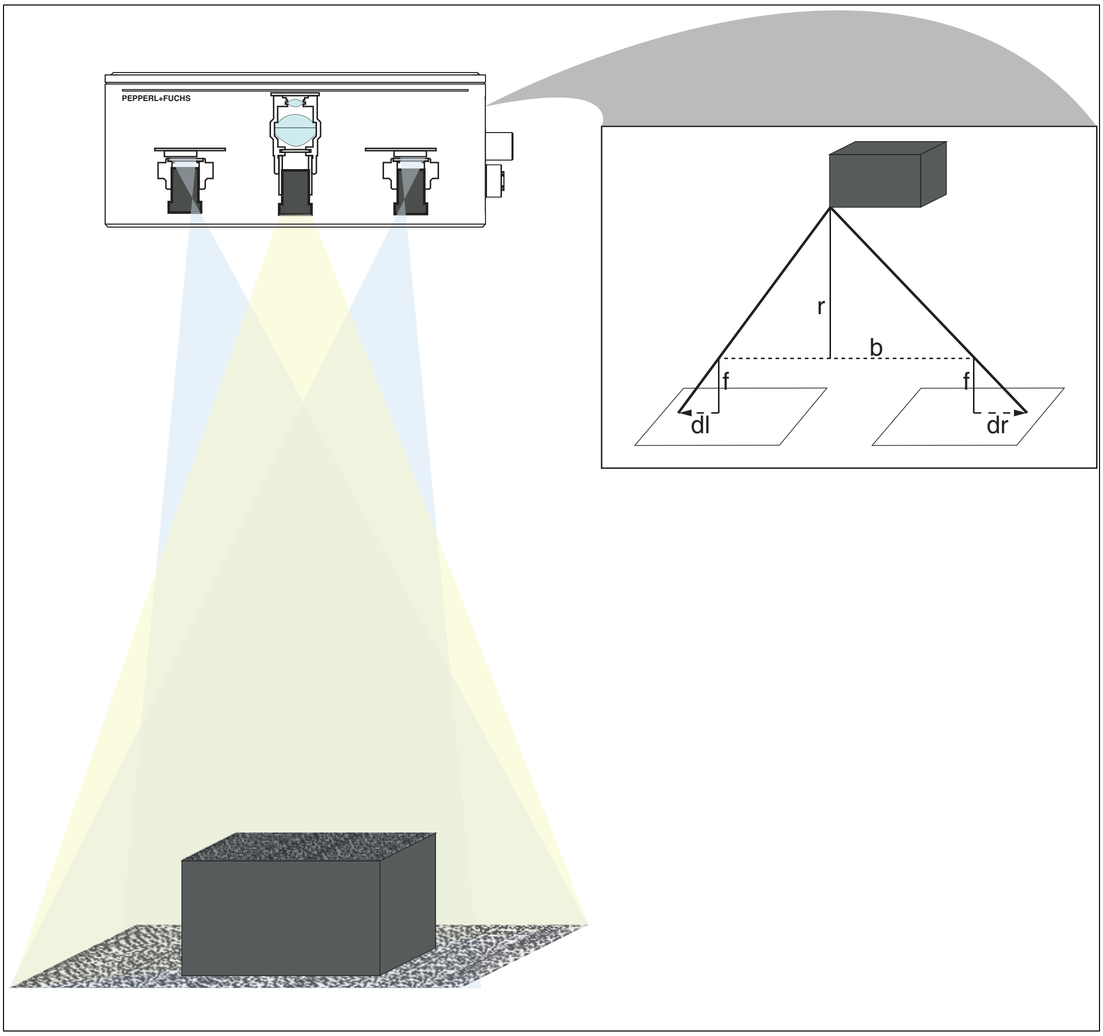


Figure 2.1 A sketch of how it works

The two single cameras, which are offset, take pictures of the same situation at the same time. In both images, the corresponding distance to a point is measured. The disparity, or offset, between the two images is calculated. The disparity (d), the focal length (f), and the base length (b) are used to create a depth in the image based on the geometric principle of triangulation. These measurements are transferred as raw data to a PC, which produces the depth image of the captured scene.

**1st step**: Two images taken from different positions.

**2nd step**: Matches found between the two images.

**3rd step**: Depth image created.

The sensor is connected to a PC via a Gigabit Ethernet network. The sensor sends the measurement results to the computer, which is running a custom application that retrieves the measurement results and processes the data contained in them.

## 2.2 GenICam

The SmartRunner Explorer 3-D sensors are GenICam-compatible devices.

GenICam (generic interface for cameras) is a common programming interface for different types of cameras.

GenICam simplifies configuration when connecting proprietary cameras. In industrial vision, when connecting a camera, the respective camera type and its parameters must be configured. GenICam makes this configuration easier. The camera communicates its configuration data to the computer. During operation, the computer asks the camera whether a particular function is supported.

GenICam provides the user with a graphical user interface that displays the features of different digital cameras in the same way, regardless of the manufacturer and camera type. This applies both to the designation, which is standardized by the SFNC (Standard Feature Naming Convention), and to certain camera-specific functions.

**Note**

Further details can be found in the current SFNC, which can be obtained via the website of the EMVA (European Machine Vision Association):

https://www.emva.org/standards-technology/genicam/genicam-downloads/

## 2.3 Symbols on the Sensor

### Overview of Labeling

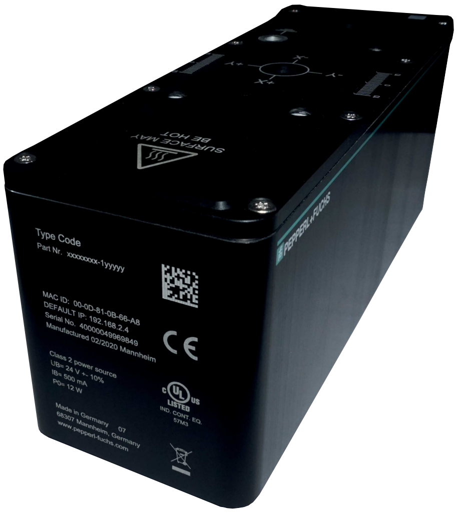
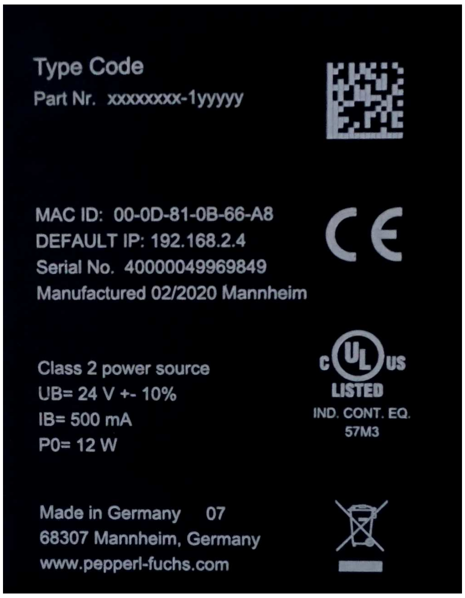


Figure 2.2 Overview of labeling

### Nameplate



Type Code
Part Nr. xxxxxxxx-1yyyyy

MAC ID: 00-0D-81-0B-66-A8
DEFAULT IP: 192.168.2.4
Serial No. 40000049969849
Manufactured 02/2020 Mannheim

Class 2 power source
UB= 24 V +- 10%
IB= 500 mA
P0= 12 W

c UL us
LISTED
IND. CONT. EQ.
57M3

Made in Germany 07
68307 Mannheim, Germany
www.pepperl-fuchs.com

Figure 2.3 Nameplate

The nameplate provides the following information:

- Part number
- CE marking
- UKCA marking
- MAC ID: 00-0D-81-0B-66-A8
- Standard IP: 192.168.2.4
- Serial number: 40000049969849
- Manufactured 02/2020 Mannheim
- UL marking
- Class 2 power source
- UB = 24 V ±10 %
- IB = 500 mA
- P0 = 12 W
- Made in Germany 07

- 68307 Mannheim, Germany
- www.pepperl-fuchs.com
- Crossed-out garbage can symbol

### Disposal

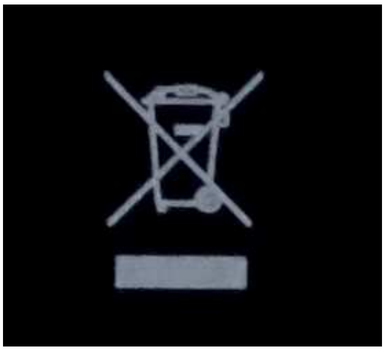


Figure 2.4 Disposal

Devices marked with this symbol must not be disposed of with household waste. These devices must only be disposed of by specialist companies approved at the operating location.

### Hot Surface

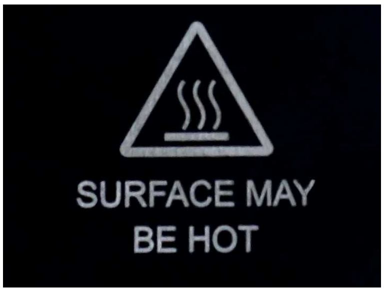


Figure 2.5 Hot surface

This label indicates a possible risk of burns from hot surfaces. Exercise the necessary caution when touching the sensor in this area.

## 2.4 Dimension Drawing

The housing of the SmartRunner 3-D Stereo has the following dimensions.

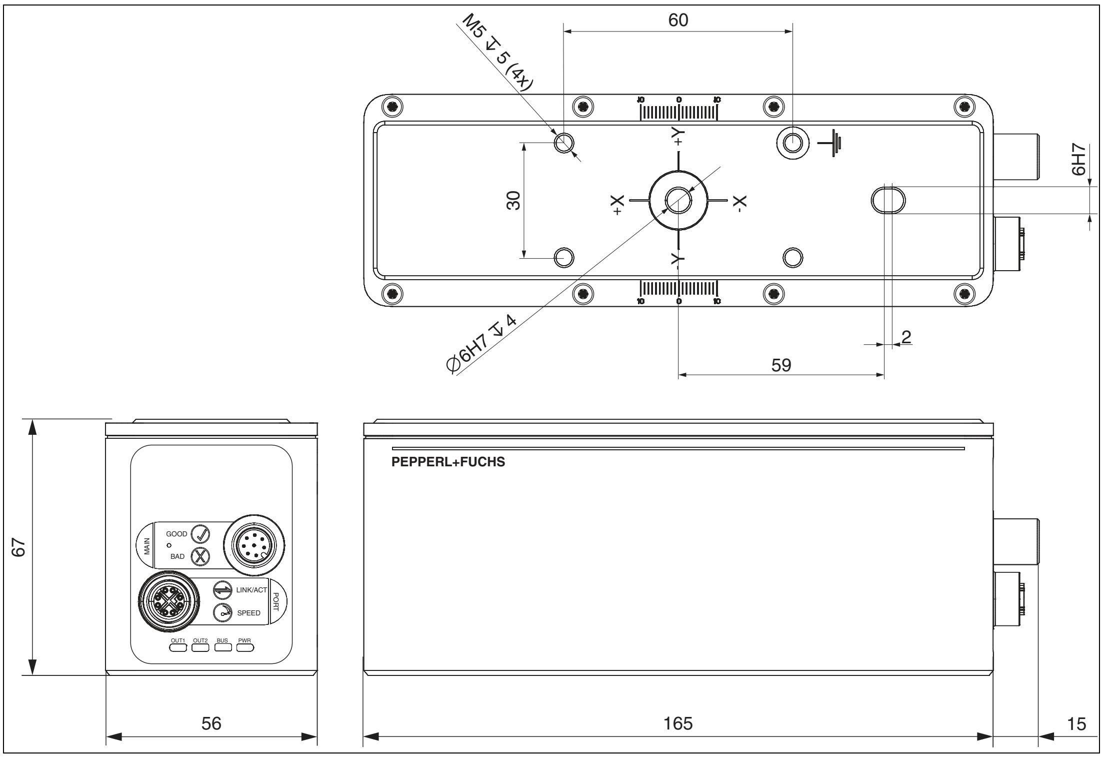


Figure 2.6 Aligning the sensor

All dimensions in mm.

**Note**

Further technical data can be found on the datasheet for the sensor. You will find the datasheets on our website.

## 2.5 Display Elements

The sensor has eight LEDs that signal various operating states or events.

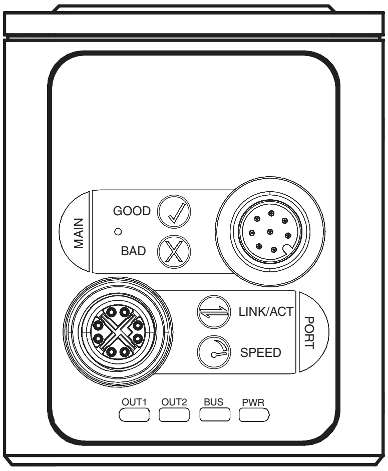


Figure 2.7 Display elements

| Designation | | LED color | Description |
|---|---|---|---|
| MAIN | GOOD ✓ | Green | Reserved |
| | BAD ✗ | Red | Reserved |
| PORT | ⇄ LINK/ACT | Yellow | The LED flashes at short intervals when bus data is being exchanged. |
| | SPEED | Green/ yellow | • No connection or 10 MBit connection—LED off<br>• 100 MBit connection—LED lights up yellow<br>• 1 Gbit connection—LED lights up green |
| Operating mode | OUT1 | Green/ yellow | Reserved |
| | OUT2 | Yellow | The OUT2 LED flashes once briefly when a new evaluation starts. |
| | BUS | - | No function |
| | PWR | Red | • The LED lights up red when the sensor restarts or is in update mode.<br>• The LED lights up red in the event of a system-related malfunction. |

## 2.6 Accessories

| Model number | Description |
|---|---|
| LTGLÜT 24V 2xM12 Xm 8pol ROB | M12 plug, 8-pin, straight, to M12 socket, 8-pin, straight, A-coded, 360° shielding, self-locking screw connection, PUR cable, torsion-proof, shielded, halogen-free |
| V19SX-G-GN5M-PUR-ABG-V45-G | Bus cable, Ethernet, M12 plug, X-coded, 8-pin to RJ45, Ethernet-coded, 8-pin, PUR cable, green, Cat6A, shielded |
| V19-G-BK1.5M-PUR-U/ABG | Single-ended female cordset, M12, straight, A-coded, 8-pin, PUR cable, black, shielded, UL-approved |

Additional accessories can be found online at www.pepperl-fuchs.com.

# 3 Installation

## 3.1 Storage and Disposal

Keep the original packaging. Always store and transport the device in the original packaging.

Store the device in a clean and dry environment. The permitted ambient conditions must be considered, see datasheet.

The device, built-in components, packaging, and any batteries contained within must be disposed in compliance with the applicable laws and guidelines of the respective country.

## 3.2 Preparation

### Unpacking the Device

1. Check the packaging and contents for damage.

   ↳ In the event of damage, inform the shipping company and notify the supplier.

2. Check the package contents against your order and the shipping documents to ensure that all items are present and correct.

   ↳ Should you have any questions, direct them to Pepperl+Fuchs.

3. Retain the original packaging in case the device is to be stored or shipped again at a later date.

## 3.3 Sensing Range

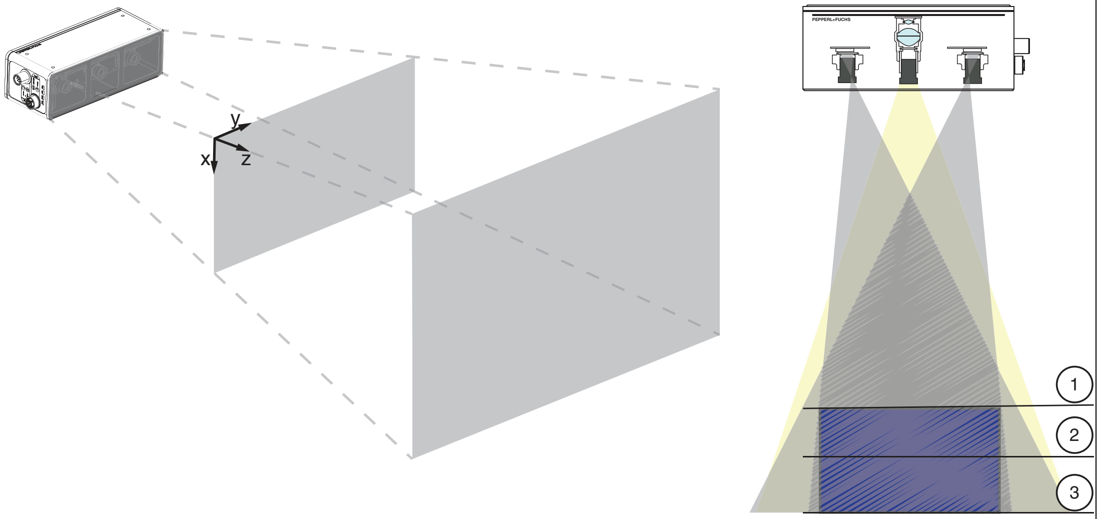


Figure 3.1 Sensing range

| Position | Description | Distance z [mm] | Measuring range x [mm] | Measuring range y [mm] | Resolution x [mm] | Resolution y [mm] | Resolution z [mm] |
|---|---|---|---|---|---|---|---|
| 1 | Minimum operating distance | 600 | 350 | 400 | 0.35 | 0.35 | 0.6 |
| 2 | Operating distance | 900 | 530 | 560 | 0.5 | 0.5 | 1.2 |
| 3 | Maximum operating distance | 1000 | 590 | 610 | 0.6 | 0.6 | 1.4 |

### Resolution

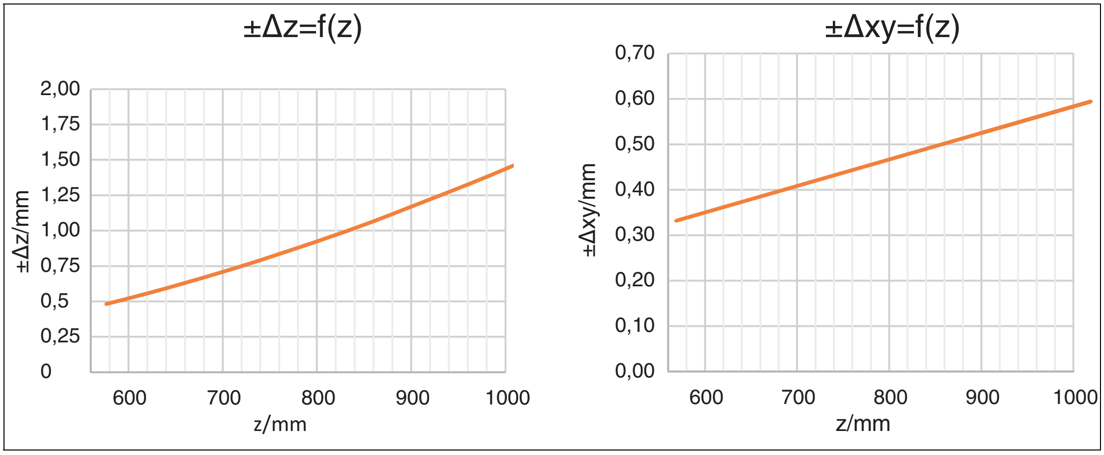


Figure 3.2 Resolution Δz and Δxy

### Measuring range

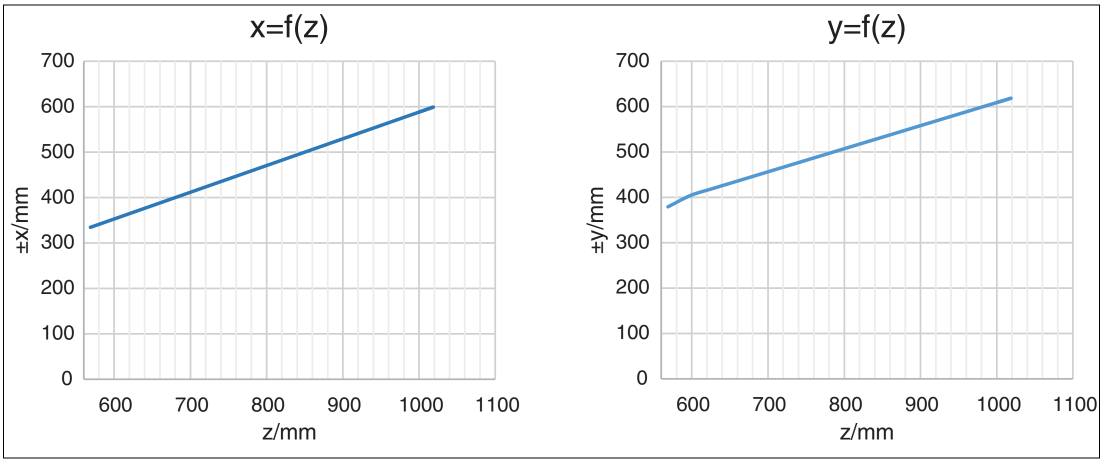


Figure 3.3 Measuring range x and y

## 3.4 Mounting

**Note**

**Mounting the Sensor**

To obtain reliable measurements, follow these instructions when mounting:

- Protect the sensor against direct and prolonged sunlight.
- Do not point the sensor into the sun.
- Prevent condensation from forming by ensuring that the sensor is not subjected to any major temperature fluctuations.
- Avoid placing objects in the scene that are not part of the target shot, especially mirrors or other shiny surfaces/objects.
- Avoid placing the sensor flat in the middle of a surface.
- Keep the glass on the device clean. Use soft cloths and standard glass cleaner to clean the device if necessary.
- Maintain a stable housing temperature during operation.
- Secure the sensor.

We recommend that the optical surfaces are cleaned, and that screws and electrical connections are checked at regular intervals.

The surface must be level to prevent the housing from becoming misaligned when the fittings are tightened. Ensure that there is still sufficient space to connect the connection cable to the sensor once the sensor is mounted.

**Caution!**

**Damage to the equipment caused by improper mounting!**

Device components can be damaged if the permissible screw-in depth and the maximum permissible tightening torque is exceeded.

Note that the threads on the bottom of the housing are not thru–holes.

Observe the maximum permissible screw-in depth to avoid damaging the device or mounting it incorrectly.

Never exceed the maximum permissible tightening speed for the fixing screws. The maximum tightening torque of the mounting screws must not exceed 2 Nm.

**Note**

**Positioning the Sensor**

The sensor has two reference holes on the back of the housing with a diameter of 6 mm and a depth of 3.5 mm. One of these holes is located exactly in the center of the sensor. The other hole is designed as a slot to prevent over-specification.

### Mounting the sensor

The sensor includes 4 M5 threads on the back of the housing to ensure easy and stable mounting in your installation.

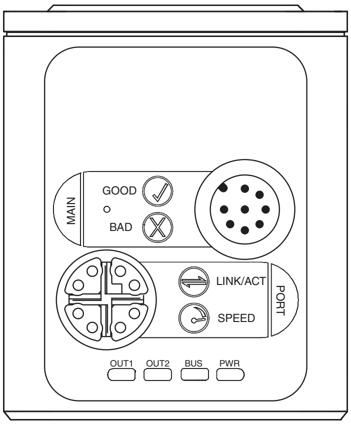
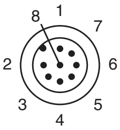
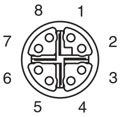
1. Place the sensor carefully in the required position in your installation. Ensure that the M5 threaded bushes on the sensor are correctly aligned with the corresponding holes or threaded holes on the installation.
2. Screw in the M5 screws clockwise by hand to ensure that they engage with the threads.

> **Note**
>
> One of the four fixing screws is not anodised, so it can be used to earth the device.

3. Use the screwdriver to tighten the M5 screws. Make sure not to overtighten the screws to avoid damaging the thread or the sensor. The maximum torque is 2 Nm.
4. Finally, check that the sensor is securely mounted to your installation. Make sure there is no loose connection or play.

## 3.5 Electrical Connection

### Connection Assignment

<table>
<tr><td rowspan="6"> </td><td colspan="2"><b>MAIN</b></td></tr>
<tr><td colspan="2">M12 plug, A-coded, 8-pin</td></tr>
<tr><td> </td><td>1. IN trigger<br>2. + UB<br>3. nc<br>4. nc<br>5. nc<br>6. nc<br>7. GND<br>8. nc</td></tr>
<tr><td colspan="2"><b>PORT</b></td></tr>
<tr><td colspan="2">M12 socket, X-coded, 8-pin</td></tr>
<tr><td> </td><td>1. MDX0+<br>2. MDX0-<br>3. MDX1+<br>4. MDX1-<br>5. MDX3+<br>6. MDX3-<br>7. MDX2+<br>8. MDX2-</td></tr>
</table>

### Connecting the Supply Voltage

The sensor is connected electrically via an 8-pin M12 **MAIN** connector plug on the front of the housing. The power is supplied via this connection. To connect the sensor, proceed as follows:

1. Plug the M12 8-pin socket into the plug on the front of the housing.
2. Screw the union nut onto the connector until it reaches the end stop. This ensures that the power cable cannot be pulled out inadvertently.

## 3.6 Setting up Windows Network Communication between the Sensor and a PC/Laptop

When the sensor is delivered, it has a fixed IP address: **192.168.2.4**. To enable communication within the network, the network settings of your PC/laptop must be synchronized with the sensor and may need to be adjusted. To do so, proceed as follows:

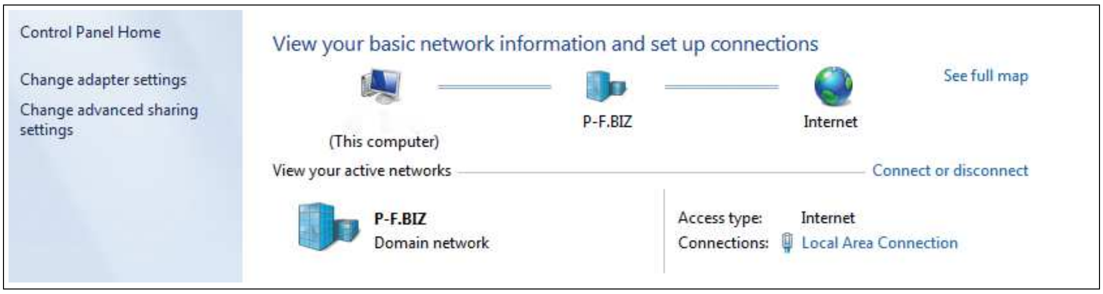
**Note**

Due to the large amount of data supplied by the sensor per second, the sensor(s) must be connected to the PC via a Gigabit Ethernet network without any other interfering data traffic.

### Setting the IP Address

The following section describes how to check the network connection settings of your Windows PC and adapt them accordingly. The illustrations in this description were created using Windows 7. The description below also applies to later versions of Windows.

1. Click the Windows **"Start"** button.
2. Select **"Control Panel > Network and Sharing Center."**
3. Click **"Change adapter settings."**



Control Panel Home

Change adapter settings

Change advanced sharing settings

View your basic network information and set up connections

See full map

(This computer)

P-F.BIZ

Internet

View your active networks

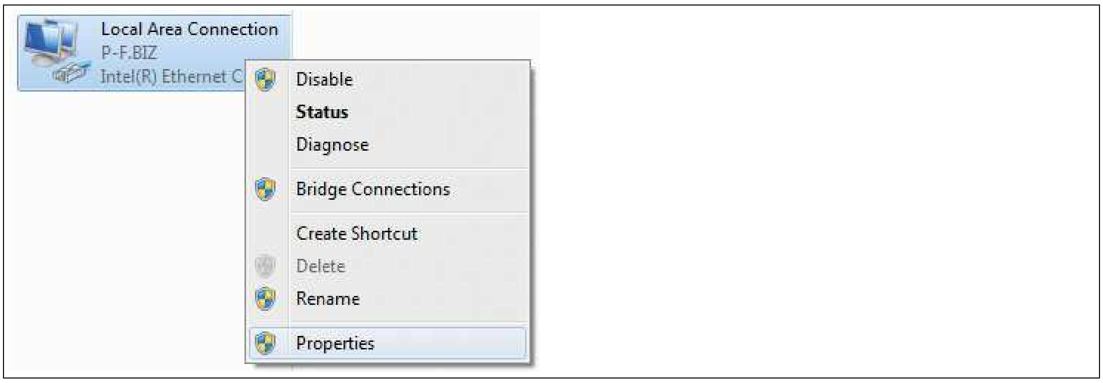
Connect or disconnect

P-F.BIZ
Domain network

Access type: Internet

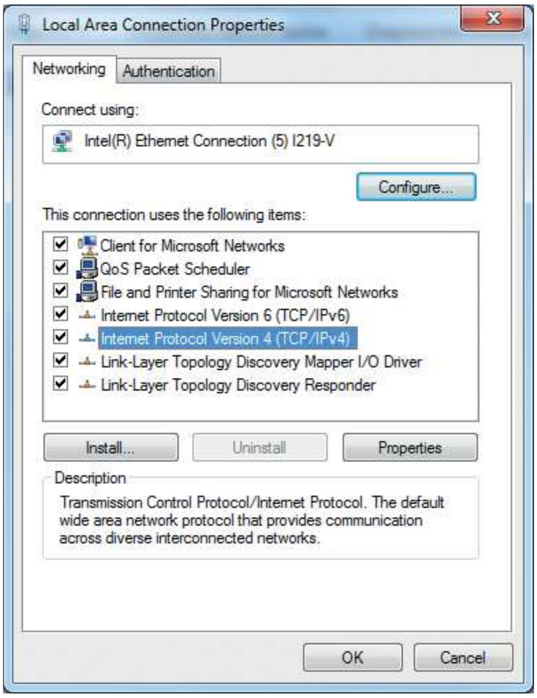
Connections: Local Area Connection

4. Select the required connection and right-click on your selection. In the selection window, select **Properties**.



Local Area Connection
P-F.BIZ
Intel(R) Ethernet C

5. Double-click **"Internet Protocol Version 4 (TCP/IPv4)."**



Local Area Connection Properties

Networking | Authentication

Connect using:

Intel(R) Ethemet Connection (5) I219-V

Configure...

This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ Internet Protocol Version 6 (TCP/IPv6)
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☑ Link-Layer Topology Discovery Mapper I/O Driver
- ☑ Link-Layer Topology Discovery Responder

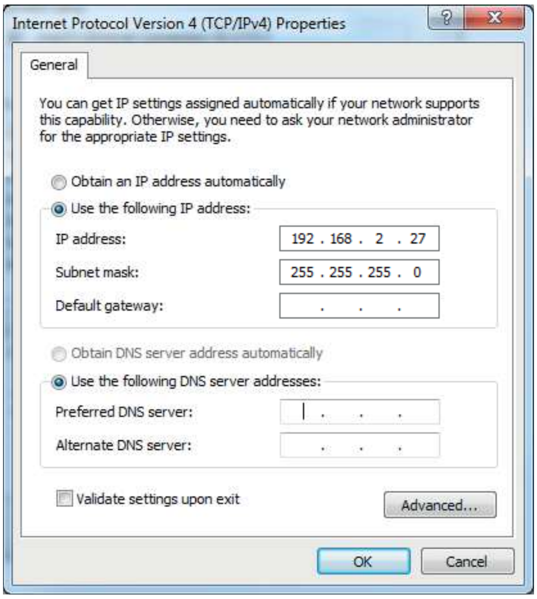
Install... | Uninstall | Properties

Description

Transmission Control Protocol/Intemet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

OK | Cancel

↳ The **Properties** window for the TCP/IP protocol opens.



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 2 . 27

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

Advanced...

OK | Cancel

6. Select the **"General"** tab.
7. Select the input function "**Use the following IP address.**"
8. Enter the IP address of the sensor, but only the first three segments of the IP address. The last segment must be different from the IP address of the sensor.
9. In this example, enter the following IP address and subnet mask:
   - **IP address: 192.168.2.27**
   - **Subnet mask: 255.255.255.0**
10. Click **OK** and click **Cancel** in the next dialog.

    ↳ This completes the network configuration so that the device can be used.

**Note**

Changes to the network settings of the PC/laptop require advanced user rights. If necessary, consult with your administrator.

# 4 Commissioning

Overview of the commissioning steps:

1. Setting the operating distance, see chapter 3.3
2. Mounting the sensor, see chapter 3.4
3. Making the electrical connection, see chapter 3.5
4. Setting up Windows network communication between the sensor and a PC/laptop, see chapter 3.6
5. Configuring the sensor using the ViSolution configuration software, see chapter 6

# 5 Configuration

## 5.1 Parameters (Features)

Before the sensor can be used in the system, it must be parameterized. The features (parameters) used for the sensor are listed below. The GenICam™ standard uses the term "feature" as the common term for parameters. Most of the features correspond to GenICam™ SFNC. "I40 parameters" are user-defined features.

The "read (r)/write (w)" column indicates whether the feature is readable ("r"), writable ("w"), or both ("r/w").

### AcquisitionControl

Provides features associated with image capture, trigger control, and exposure control. This includes:

| Feature name | Data type | read (r)/ write (w) | Value | | enum values / default string value | SFNC 2.6 | Description |
|---|---|---|---|---|---|---|---|
| | | | Min | Max | | | |
| TriggerEnabled | bool | r/w | - | - | - | - | - |
| AutoTriggerFrameRate | float | r/w | - | - | - | 5.5.6 | Controls the acquisition rate (in hertz) at which the images are acquired. Frame rate in Hz (AutoTrigger) |
| ExposureTime | float | r/w | 0.15 | 1 | - | 5.7.4 | Sets the exposure time (in ms) |
| TriggerSoftware | command | -/w | - | - | - | 5.6.3 | Generates an internal trigger. |
| TriggerSource | enum | r/w | - | - | Software, Line1, AutoTrigger | 5.6.4 | Specifies the internal signal or physical input line to be used as the trigger source. |
| TriggerActivation | enum | r/w | - | - | RisingEdge | 5.6.5 | Specifies the trigger activation mode. |
| OutputMode | enum | r/w | - | - | DisparityC, CalibratedABC_Grid, LeftRaw, RightRaw, LeftRectified, RightRectified, StereoRaw, StereoRectified, DisparityCAndRaw, DisparityCAndRectified | 21.4.7 | Specifies the content for the output data |

## Stereo Proprietary Parameters

Contains features related to control and to general information about the sensor. These are used to identify the sensor during the enumeration process and to obtain information about the sensor resolution. Additional information and controls that affect the general status of the sensor are also included in this category.

| Feature name | Data type | read (r)/ write (w) | Value | | enum values / default string value | SFNC 2.6 | Description |
|---|---|---|---|---|---|---|---|
| | | | Min | Max | | | |
| SGBMPenalty1 | int | r/w | 0 | 31 | - | - | Tuning parameter (effort to smooth a disparity jump of 1 in the output) |
| SGBMPenalty2 | int | r/w | 0 | 31 | - | - | Tuning parameter (effort to smooth a disparity jump of more than 1 in the output) |
| SGBMUniqueness | int | r/w | 0 | 100 | - | - | Tuning parameter (maximum relative effort for the best disparity compared to the second best disparity, in percent) |
| SGBMEnable-Median | bool | r/w | 0 | 1 | - | - | Allows media filtering of the output disparity card |
| Illumination | enum | r/w | - | - | Projector,<br>None,<br>Onboard | - | Light source to be used |
| Gain | int | r/w | 0 | 480 | - | - | Amplification of light sensitivity in 1/10 dB |

## 5.2 VsxProtocolDriver

### General

**VsxProtocolDriver** is the driver used to support the creation of a graphical user interface for sensors that use the VSX protocol. The driver connects to the sensor and handles the communication in accordance with the protocol. The user can access functions for setting parameters on the sensor, retrieving parameter values from the sensor, and saving and loading whole parameter sets both locally and on the sensor. The user can also receive sensor images. Each function also contains an error object from which information can be obtained in the event of an error in the function.

The driver is implemented in C# and requires .NET Standard 2.0 or higher.

The functions of the driver can be used **synchronously** or **asynchronously**. For this, the required instance must be created using the Init function.

**Asynchronous**

```
VsxProtocolDriver _asyncVsxDriver = VsxProtocolDriver.Init(string
serialPort, int baudrate, string sensorType, TheSensor.ConnectionType
connectionType, string pluginName = "");
```

**Synchronous**

```
VsxProtocolDriverSync _syncVsxDriver =
VsxProtocolDriverSync.Init(string serialPort, int baudrate, string
sensorType, TheSensor.ConnectionType connectionType, string
pluginName = "");
```

### Synchronous and Asynchronous Functions

The auxiliary classes used in the parameters are described below.

### Static Functions

#### UDP Broadcast

```
public static async Task<(bool Succ, List<Device> DeviceList, Error
ErrorDesc)> UdpDeviceList()
public static (bool Succ, List<Device> DeviceList, Error ErrorDesc)
UdpDeviceList()
```

The function returns a list of the Vsx devices that are found on the network via UDP broadcast.
Possible error IDs: VSX_DRIVER_CONNECTION_ERROR

#### New Driver Instance for TCP/IP

```
public static VsxProtocolDriver Init(string ipAddress, int port =
VSXPORT, string pluginName = "")
public static VsxProtocolDriverSync Init(string ipAddress, int port
=VSXPORT, string pluginName = "")
```

Initializes a new driver instance that can be used to communicate with the device via TCP/IP.
While the IP address must be specified, the default VSXPORT = 50005 can be used.
Possible error IDs: None

#### New Driver Instance, Serial

```
public static VsxProtocolDriver Init(string serialPort, int baudrate, string sensorType, TheSensor.ConnectionType connectionType,
string pluginName = "")
public static VsxProtocolDriverSync Init(string serialPort, int baudrate, string sensorType, TheSensor.ConnectionType connectionType,
string pluginName = "")
```

Initializes a new driver instance that allows serial communication with the device.
Possible error IDs: None

### IVsxMessage

```
public static (bool Succ, Error ErrorDesc) SaveData(string filename, IVsxMessage message)
public static (bool Succ, Error ErrorDesc) SaveData(string filename, IVsxMessage message)
```

Saves an IVsxMessage under the specified file name.
Possible error IDs: VSX_DRIVER_DATA_ERROR, VSX_DRIVER_INVALID_DATA_ERROR, VSX_DRIVER_SAVE_FILE_ERROR

### Point Cloud File

```
public static (bool Succ, Error ErrorDesc) Save3DPointCloudData(string filename, VsxImageData2Message x, VsxImageData2Message y, VsxImageData2Message z)
public static (bool Succ, Error ErrorDesc) Save3DPointCloudData(string filename, VsxImageData2Message x, VsxImageData2Message y, VsxImageData2Message z)
```

Saves a point cloud file consisting of levels x, y, and z under the specified file name.
Possible error IDs: VSX_DRIVER_DATA_ERROR, VSX_DRIVER_SAVE_FILE_ERROR

## Non-Static Functions

### Establishing a Connection

```
public async Task<(bool Succ, Error ErrorDesc)> Connect(int timeout = CONNECTION_TIMEOUT_MS)
public (bool Succ, Error ErrorDesc) Connect(int timeout = VsxProtocolDriver.CONNECTION_TIMEOUT_MS)
```

Establishes a connection to the device using the parameters set using Init. CONNECTION_TIMEOUT_MS = 1000 can be used as the timeout for opening the connection. A connection to the device must be established in order to use all non-static functions.
Possible error IDs: VSX_DRIVER_CONNECTION_ERROR

### Disconnecting

```
public async Task<(bool Succ, Error ErrorDesc)> Disconnect()
public (bool Succ, Error ErrorDesc) Disconnect()
```

Disconnects from the device.
Possible error IDs: None

### Device Information

```
public async Task<(bool Succ, Device CurrentDevice, Error ErrorDesc)> GetCurrentDeviceInformation()
public (bool Succ, Device CurrentDevice, Error ErrorDesc) GetCurrentDeviceInformation()
```

Returns information about the device.
Possible error IDs: VSX_DRIVER_CONNECTION_ERROR

### Device Features

```
public async Task<(bool Succ, float XmlVersion, Hashtable FeatureList, Error ErrorDesc)> GetFeatureList()
public (bool Succ, float XmlVersion, Hashtable FeatureList, Error ErrorDesc) GetFeatureList()
```

Returns the list of features available on the device.
Possible error IDs: VSX_DRIVER_CONNECTION_ERROR

### Device Parameters

```
public async Task<(bool Succ, List<Parameter> ParameterList, Error ErrorDesc)> GetParameterList()
public (bool Succ, List<Parameter> ParameterList, Error ErrorDesc) GetParameterList()
```

Returns a list of all parameters available on the device, including detailed information and their current values.
Possible error IDs: VSX_DRIVER_GENERAL_ERROR, VSX_DRIVER_DATA_ERROR, VSX_DRIVER_CONNECTION_ERROR

### Returning the Value of a Single Device Parameter

```
public async Task<(bool Succ, object parameterValue, Error ErrorDesc)> GetSingleParameterValue(Parameter parameter)
public async Task<(bool Succ, object parameterValue, Error ErrorDesc)> GetSingleParameterValue(string parameterId)
public async Task<(bool Succ, object parameterValue, Error ErrorDesc)> GetSingleParameterValue(ushort settingsVersion, ushort configVersion, string configId, string parameterId)
public (bool Succ, object parameterValue, Error ErrorDesc) GetSingleParameterValue(Parameter parameter)
public (bool Succ, object parameterValue, Error ErrorDesc) GetSingleParameterValue(string parameterId)
public (bool Succ, object parameterValue, Error ErrorDesc) GetSingleParameterValue(ushort settingsVersion, ushort configVersion, string configId, string parameterId)
```

Returns the value of an individual parameter of the device. This can be defined via a parameter object which can be obtained, for example, from the list for the "GetParameterList" function. Alternatively, it is uniquely defined by its version and IDs. A shortened version of this also only accepts the ParameterId. This version can be used if the ParameterId of the desired parameter is unique across the device.
Possible error IDs: VSX_DRIVER_CONNECTION_ERROR, VSX_DRIVER_DATA_ERROR

### Setting the Value of a Single Device Parameter

```
public async Task<(bool Succ, Error ErrorDesc)> SetSingleParameterValue(Parameter parameter, object value)
public async Task<(bool Succ, Error ErrorDesc)> SetSingleParameterValue(string parameterId, object value)
public async Task<(bool Succ, Error ErrorDesc)> SetSingleParameterValue(ushort settingsVersion, ushort configVersion, string configId, string parameterId, object value)
public (bool Succ, Error ErrorDesc) SetSingleParameterValue(Parameter parameter, object value)
public (bool Succ, Error ErrorDesc) SetSingleParameterValue(string parameterId, object value)
public (bool Succ, Error ErrorDesc) SetSingleParameterValue(ushort settingsVersion, ushort configVersion, string configId, string parameterId, object value)
```

Sets the value of an individual parameter on the device. The parameter is defined by transferring the function parameters in the same way as for the GetSingleParameterValue function (see description for this function). In addition, the desired value is transferred.
Possible error IDs: VSX_DRIVER_CONNECTION_ERROR, VSX_DRIVER_DATA_ERROR

### Changing Network Settings

```
public async Task<(bool Succ, Error ErrorDesc)> SetNetworkSettings(string ipAddress, string networkMask, string gateway)
public (bool Succ, Error ErrorDesc) SetNetworkSettings(string ipAddress, string networkMask, string gateway)
```

Changes the network settings on the device. The connection to the device is then disconnected and must be re-established using the "Connect" function.
Possible error IDs: VSX_DRIVER_CONNECTION_ERROR

### Sending the Firmware File

```
public async Task<(bool Succ, Error ErrorDesc)> SendFirmware(string fileName)
public (bool Succ, Error ErrorDesc) SendFirmware(string fileName)
```

Sends the firmware file to the device under the specified path and file name. The current status can be read out via "Firmware StateChannelReader" while the update is in progress.
Possible error IDs: VSX_DRIVER_CONNECTION_ERROR, VSX_DRIVER_DEVICE_ERROR

### Reading and Saving the Parameter Set

```
public async Task<(bool Succ, Error ErrorDesc)> DownloadParameterSet(string destinationFileName)
public (bool Succ, Error ErrorDesc) DownloadParameterSet(string destinationFileName)
```

Reads the current parameter set from the device and saves it under the specified path and file name.
Possible error IDs: VSX_DRIVER_SAVE_FILE_ERROR

```
public async Task<(bool Succ, Error ErrorDesc)> SaveParameterSetOnDevice()
public (bool Succ, Error ErrorDesc) SaveParameterSetOnDevice()
```

Saves the current parameter settings on the device. The set values are then set each time the device is started.
Possible error IDs: VSX_DRIVER_CONNECTION_ERROR

### Loading Parameter Settings

```
public async Task<(bool Succ, List<Parameter> ParameterList, Error ErrorDesc)> LoadParameterSetOnDevice()
public (bool Succ, List<Parameter> ParameterList, Error ErrorDesc) LoadParameterSetOnDevice()
```

Loads the parameter settings saved via "SaveParametersetOnDevice()" on the device. The parameters then have the previously saved values. A current parameter list is returned.
Possible error IDs: VSX_DRIVER_GENERAL_ERROR, VSX_DRIVER_DATA_ERROR, VSX_DRIVER_CONNECTION_ERROR

### Loading Factory Settings

```
public async Task<(bool Succ, List<Parameter> ParameterList, Error ErrorDesc)> LoadDefaultParameterSetFromDevice()
public (bool Succ, List<Parameter> ParameterList, Error ErrorDesc) LoadDefaultParameterSetFromDevice()
```

Loads the factory settings of all parameters on the device. A current parameter list is returned.
Possible error IDs: VSX_DRIVER_GENERAL_ERROR, VSX_DRIVER_DATA_ERROR, VSX_DRIVER_CONNECTION_ERROR

### Grabbing Sensor Data

```
public void ResetDynamicContainerGrabber(int bufferSize, int startCondition = -1, Strategy strategy = Strategy.DROP_OLDEST)
public void ResetDynamicContainerGrabber(int numberOfItems, int startCondition = -1, Strategy strategy = Strategy.DROP_OLDEST)
```

Restarts the grabbing of sensor data. The sensor data for each trigger is compressed in VsxDynamicContainer. The data can be called up with the GetDynamicContainer function. For more details, see the container description. The bufferSize specifies how many containers can be buffered by the driver, the startCondition from which containers should be buffered, and the strategy to be implemented when the buffer is full. DROP_OLDEST indicates that the oldest container currently stored is discarded, while DROP_WRITE means the latest container received is discarded.
Possible error IDs: None

### Output of the Oldest Buffered "DynamicContainer"

```
public async Task<(bool Succ, IVsxDynamicContainer Container, int NumberOfDiscardedItems, Error ErrorDesc)> GetDynamicContainer(int timeoutMs = Timeout.Infinite)
public (bool Succ, IVsxDynamicContainer Container, int NumberOfDiscardedItems, Error ErrorDesc) GetDynamicContainer(int timeoutMs = Timeout.Infinite)
```

Returns the oldest buffered "DynamicContainer" (see "ResetDynamicContainerGrabber"). "timeoutMs" specifies the time in ms for which an attempt is made to read a container from the buffer. If no container is present during this time, the function returns an error.
Possible error IDs: VSX_DRIVER_INIT_ERROR, VSX_DRIVER_TIMEOUT_ERROR

### Grabbing Log Data

```
public void ResetLogMessageGrabber(int bufferSize, int typeMask, Strategy strategy = Strategy.DROP_OLDEST)
public void ResetLogMessageGrabber(int bufferSize, int typeMask, Strategy strategy = Strategy.DROP_OLDEST)
```

Restarts the grabbing of log data. The log data can be called up with the "GetLogMessage" function. The "bufferSize" specifies how many log data messages can be buffered by the driver. The "typeMask" specifies which log data types are to be transferred from the device, and the "strategy" specifies what should happen when the buffer is full. "DROP_OLDEST" indicates that the oldest log message currently stored is discarded, while "DROP_WRITE" means the latest log message received is discarded.
Possible error IDs: None

### Output of the Oldest Buffered "LogMessage"

```
public async Task<(bool Succ, VsxLogMessage LogMessage, int NumberOfDiscardedItems, Error ErrorDesc)> GetLogMessage(int timeoutMs = Timeout.Infinite)
public (bool Succ, VsxLogMessage LogMessage, int NumberOfDiscardedItems, Error ErrorDesc) GetLogMessage(int timeoutMs = Timeout.Infinite)
```

Returns the oldest buffered "LogMessage" (see "ResetLogMessageGrabber"). "timeoutMs" specifies the time in ms for which an attempt is made to read a log message from the buffer. If no log message is present during this time, the function returns with an error.
Possible error IDs: VSX_DRIVER_INIT_ERROR, VSX_DRIVER_TIMEOUT_ERROR

## Special Asynchronous Functions

### Information about Firmware Updates

```
public ChannelReader<FirmwareState> FirmwareStateChannelReader
```

Can be read out asynchronously and contains information about the current status during a firmware update.

# Properties

## Connection Status

```
public bool Connected { get; }
```

Indicates the connection status.

## Timeout

```
public int WaitTimeout { get; set; }
```

Timeout in ms, indicating how long the device will wait for a response to its request. Depending on the connection type, the default value is "DEFAULT_ETHERNET_TIMEOUT_MS" or "DEFAULT_SERIAL_TIMEOUT_MS."

## Number of Discarded "DynamicContainer" or "LogMessages"

```
public int MissingContainerFramesCounter { get; }
public int MissingLogMessagesCounter { get; }
```

Specifies the number of "DynamicContainer" or "LogMessages" discarded since the last reset grabber, once there was no more space in the buffer.

## Number of Buffered "DynamicContainer" or "LogMessages"

```
public int DynamicContainerQueueSize { get; }
public int LogMessageQueueSize { get; }
```

Specifies how many dynamic containers or log messages can be buffered by the driver.

## Indication of Containers or Log Messages to Be Discarded

```
public Strategy DynamicContainerGrabberStrategy { get; }
public Strategy LogMessageGrabberStrategy { get; }
```

Specifies which containers or log messages are to be discarded if the buffer is full. "DROP_OLDEST" discards the oldest stored container or log message, while "DROP_WRITE" discards the latest one received.

# Events

## Loss of Connection

```
public event Action<string, string> OnDisconnect
```

Triggered as soon as the driver detects a loss of connection to the device. Parameters are the IP of the previously connected device and a message indicating why the connection was lost. This event is only triggered if a TCP/IP connection is used.

# Auxiliary Classes

## Device

```
public string PhysicalAddress;
public int PhysicalPort;
public string IpAddress;
public string NetworkMask;
public string Gateway;
public string MacAddress;
public string Identifier;
public string FirmwareVersion;
public string SensorType;
```

Contains information about the currently connected device.

### Error

```
public ErrorId Id;
public string Tag;
public string Message;
```

Contains information about an error that has occurred.

### Possible Error IDs

```
VSX_DRIVER_NO_ERROR = 0x0,
VSX_DRIVER_INIT_ERROR = -0x1,
VSX_DRIVER_TIMEOUT_ERROR = -0x2,
VSX_DRIVER_SAVE_FILE_ERROR = -0x3,
VSX_DRIVER_DATA_ERROR = -0x4,
VSX_DRIVER_CONNECTION_ERROR = -0x5,
VSX_DRIVER_INVALID_DATA_ERROR = -0x6,
VSX_DRIVER_DEVICE_ERROR = -0x7,
VSX_DRIVER_GENERAL_ERROR = -0x1000
```

### Parameter

```
ushort settingsVersion;
ushort configVersion;
string configId;
string parameterId;
string name;
Vsx.ParameterTypes type;
Vsx.ValueTypes valueType;
bool enable;
bool visible;
object min;
object max;
string userLevel;
object value;
object defaultValue;
string unit;
List<ItemTuple> items;
```

Contains information about a device parameter. Important properties here are the details of the version and IDs, which are required for setting a parameter. Another property with "Value" is the current value of the parameter. Not every property is used for every parameter.

### Firmware State

```
public int Id;
public string Tag;
public string Message;
```

Contains information about the current status of a firmware update that is running.

### VsxDynamicContainerMessage: IVsxMessage

```
public bool ContainsMessage(string tag)
public IVsxMessage GetMessage(string tag)
```

Contains a list of "IVsxMessages" which in turn contain data sent by the device. The contained messages are identified in the list by means of a string. The possible messages are device-specific.

### VsxImageData2Message: IVsxMessage

```
public ImageData2Format;
public int Width;
public int Height;
public int LinePitch;
public long FrameCounter;
public double CoordinateScale;
public double CoordinateOffset;
public double AxisMin;
public double AxisMax;
public double InvalidDataValue;
public byte[] ImageData;
public float[] ImageDataFloats;
```

Contains image data of a specific image. Depending on whether the individual image values are bytes or floats, they are stored in the respective "ImageData" or "ImageDataFloats" array.

### VsxDisparityDescriptorMessage: IVsxMessage

```
public double FocalLength;
public double PrincipalPointU;
public double PrincipalPointV;
public double Baseline;
```

### VsxTransformationMessage: IVsxMessage

```
public double TranslationTX;
public double TranslationTY;
public double TranslationTZ;
public double QuaternionQ0;
public double QuaternionQ1;
public double QuaternionQ2;
public double QuaternionQ3;
```

## SmartRunner 3-D

A "DynamicContainer" received by the SmartRunner 3-D can contain the following messages:

| TagName | Type | Status | Description |
|---|---|---|---|
| "LeftRaw" | ImageData2[Mono8] | Optional | Left raw image |
| "RightRaw" | ImageData2[Mono8] | Optional | Right raw image |
| "LeftRectified" | ImageData2[Mono8] | Optional | Left rectified image |
| "RightRectified" | ImageData2[Mono8] | Optional | Right rectified image |
| "DisparityC" | ImageData2[Coord3D_C16] | Optional | Disparity map (1) |
| "DisparityDesc" | DisparityDescriptor | Optional | Needed by Disparity map (1) |
| "CalibratedA"[1] | ImageData2[Coord3D_A16] | Optional | X layer of calibrated grid |
| "CalibratedB"[2] | ImageData2[Coord3D_B16] | Optional | Y layer of calibrated grid |
| "CalibratedC"[3] | ImageData2[Coord3D_C16] | Optional | Z layer of calibrated grid |
| "Transformation" | Transformation | Optional | Transformation of XYZ |
| "Confidence" | ImageData2[Confidence8] | Optional | Confidence map (2) |

[1] The three layers "CalibratedA," "CalibratedB," and "CalibratedC" are not sent on the device side. The "ProcessDeviceData" function (VsxDynamicContainerMessage container) generates these three messages and calculates the respective level of the "PointCloud."

[2] The three layers "CalibratedA," "CalibratedB," and "CalibratedC" are not sent on the device side. The "ProcessDeviceData" function (VsxDynamicContainerMessage container) generates these three messages and calculates the respective level of the "PointCloud."

[3] The three layers "CalibratedA," "CalibratedB," and "CalibratedC" are not sent on the device side. The "ProcessDeviceData" function (VsxDynamicContainerMessage container) generates these three messages and calculates the respective level of the "PointCloud."

# 6 Operation with ViSolution

The "ViSolution" software can be used to configure sensor parameters and evaluate measurement results. The software is also very helpful when mounting the sensor, for adjusting camera parameters such as the exposure time.

The user interface offers various options for displaying the measurement results. You can also save measurement results to a file for later use. You can change the settings for the sensor and immediately see how the changes affect the measurement result. The user interface is therefore useful for determining the best parameter settings for a particular application.

After the sensor has been set up to provide measurement data, the settings can be saved to a parameter file on the PC.

To do this, you must have the **ViSolution** configuration software installed on your PC/laptop. This configuration software for a PC/laptop can be downloaded free of charge from www.pepperl-fuchs.com.

## 6.1 Establishing a Connection to the Sensor

After starting the ViSolution software, the following "Device overview" window is displayed:

### Device Overview

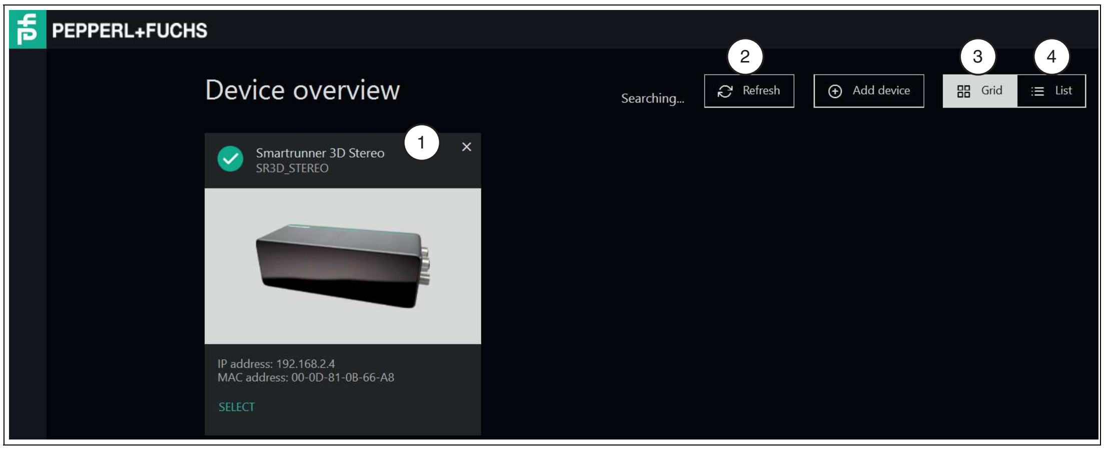


Figure 6.1 Device overview

| Position | Designation | Function |
|---|---|---|
| 1 | Device overview | All identified sensors are displayed here. You will also receive information about the sensor.<br>• Name of the sensor<br>• Sensor type<br>• IP address of the sensor<br>• MAC address of the sensor<br>SELECT is highlighted in green for sensors to which a connection can be established; if no connection can be established, SELECT is grayed out.<br>If the list contains multiple sensors, you can use the SELECT button to select the sensor that you wish to connect to. |
| 2 | Refresh | You can use the "Refresh" button to start a sensor search. |

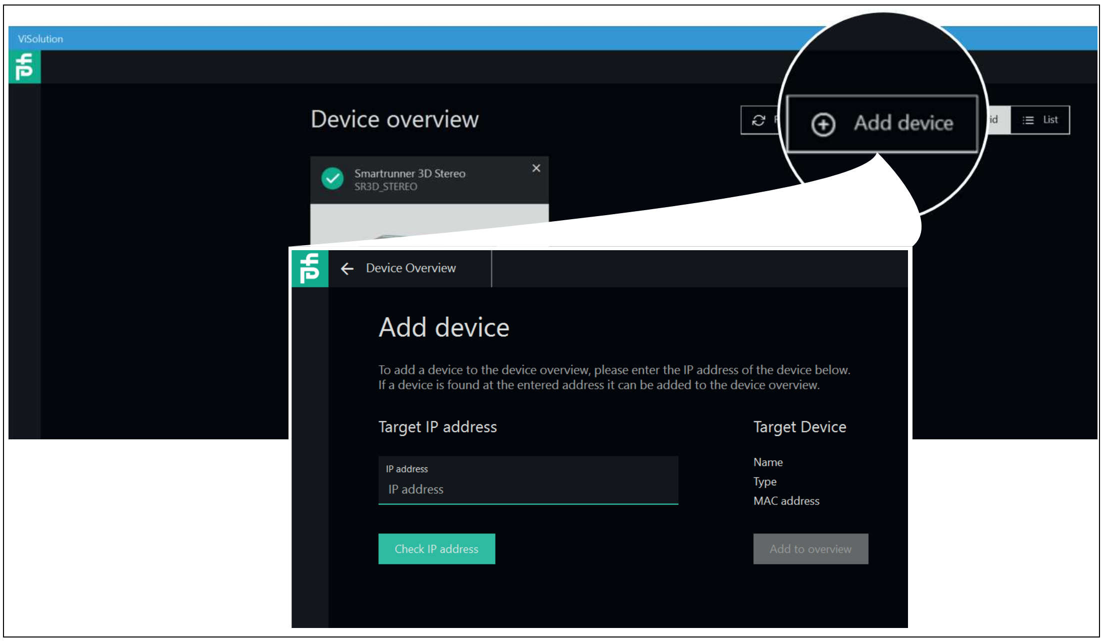
| Position | Designation | Function |
|---|---|---|
| 3 | Grid | You can use the "Grid" button to change the way the sensors found are displayed. If you select "Grid," the sensors found are displayed as tiles. |
| 4 | List | You can use the "List" button to change the way the sensors found are displayed. If you select "List," the sensors found are displayed in a list. |

## Selecting a Sensor

1. Click the button highlighted in green SELECT for the sensor that has been found.

↳ A connection to the sensor is established. The main screen of the software opens.

If your sensor is not recognized in the device overview, connect your sensor as follows:



Add device

To add a device to the device overview, please enter the IP address of the device below.
If a device is found at the entered address it can be added to the device overview.

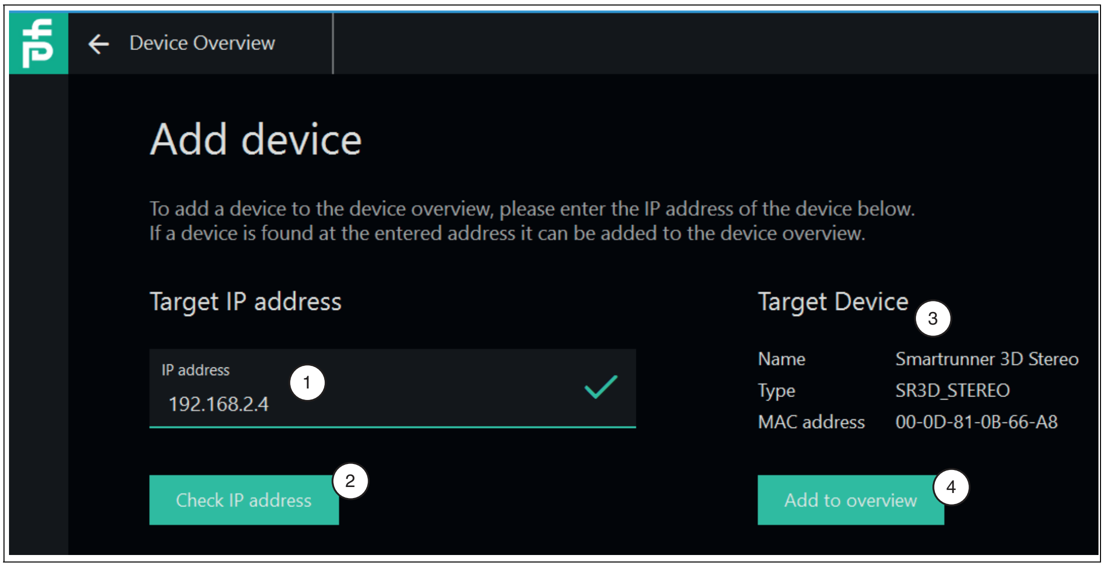
Figure 6.2 Device overview

## Finding a Sensor

This function can be used to search for connected sensors or to establish a connection to a connected device. Proceed as follows:

1. Ensure that the sensor and the PC/laptop are ready for operation and that there is an Ethernet connection.
2. Click the "Add device" button to search for sensors.

   ↳ The "Add device" input window opens.



Device Overview

Add device

To add a device to the device overview, please enter the IP address of the device below.
If a device is found at the entered address it can be added to the device overview.

Target IP address

IP address: 192.168.2.4 ✓

Target Device

Name: Smartrunner 3D Stereo
Type: SR3D_STEREO
MAC address: 00-0D-81-0B-66-A8

Figure 6.3 Entering the IP address

3. Enter the IP address of your sensor in the "Target IP address" input field (1). The default IP address of the sensor can be found on the device.
4. Click the "Check IP address" button (2) to search for the device with the entered IP address.

   ↳ A green check mark in the search field indicates that the device with the entered IP address has been found. In addition, the following information on the sensor is displayed under "Target Device" (3):
   - Name of the sensor
   - Sensor type
   - MAC address
5. Click the "Add to overview" button (4).

   ↳ The sensor is displayed on the overview screen (see "Device Overview" on page 34) of all sensors found.

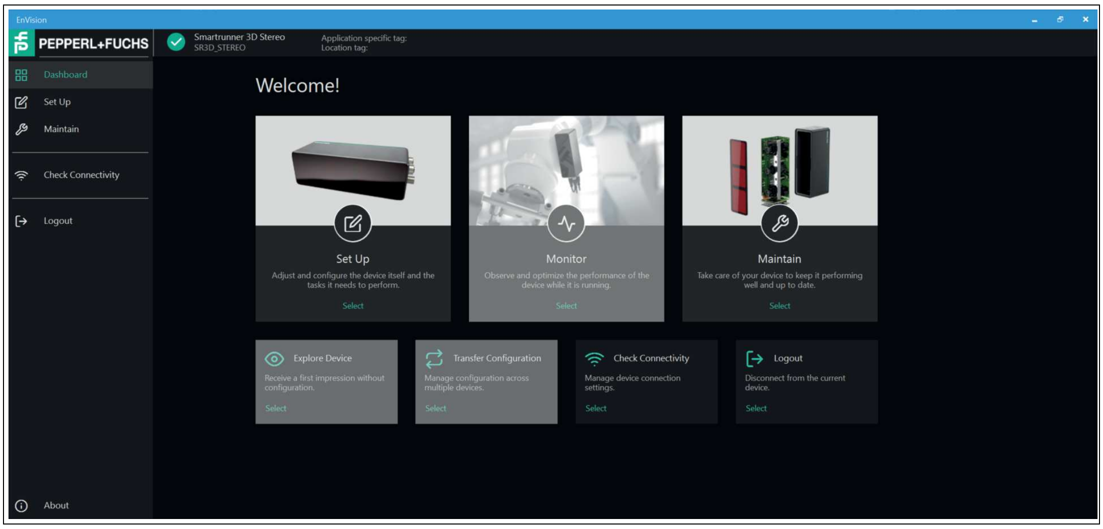
**Note**

If no sensor is detected:

- Check that the sensor is connected properly and ready for operation. See chapter 3.5, .
- Check that Windows network communication is set up correctly between the device and the PC/laptop; see chapter 3.6.

## 6.2 ViSolution Main Screen

The basic ViSolution functions can be selected on the main screen.



Adjust and configure the device itself and the tasks it needs to perform.

Observe and optimize the performance of the device while it is running.

Take care of your device to keep it performing well and up to date.

Receive a first impression without configuration.

Manage configuration across multiple devices.

Manage device connection settings.

Disconnect from the current device.

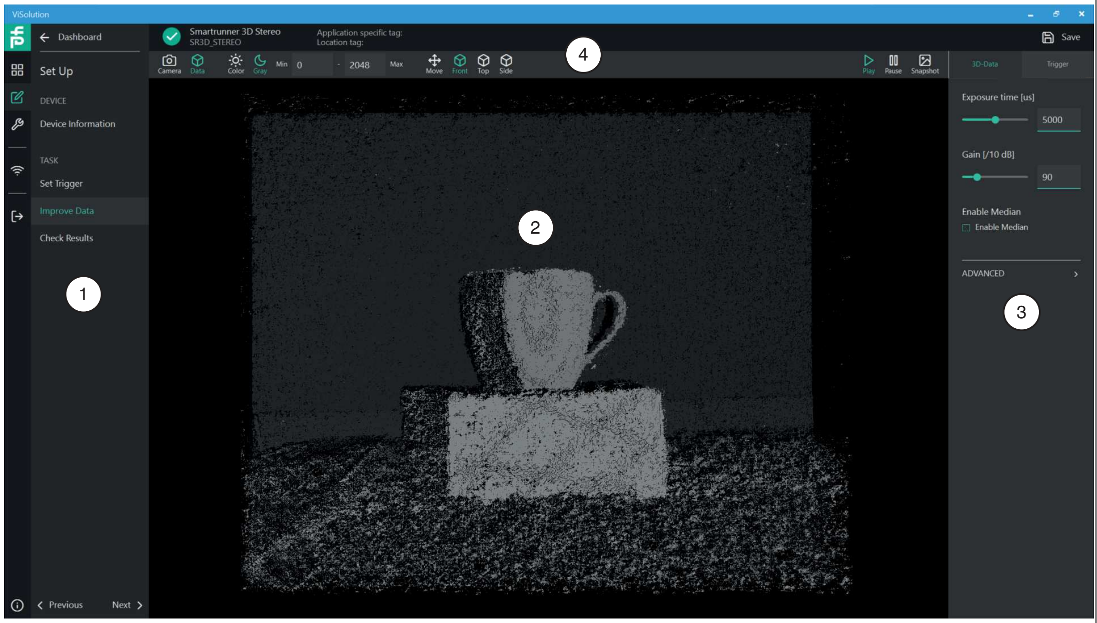
Figure 6.4 Main screen

Basic functions on the main screen:

| Designation | Description |
|---|---|
| Set Up | Manage and configure the sensor and applications. See chapter 6.2.2. |
| Monitor | Monitor and optimize the performance of the device during operation. See chapter 6.2.3. |
| Maintain | You can use the Maintain function to check the status of the sensor and keep it up to date. See chapter 6.2.4. |
| Explore Device | This menu allows you to gain an initial impression of the sensor without configuring the sensor. See chapter 6.2.5. |
| Transfer Configuration | Manage configurations across multiple sensors. See chapter 6.2.6. |
| Check Connectivity | Use this menu to manage connections to the network and display connection data. See chapter 6.2.7. |
| Logout | Use "Logout" to exit the user interface and access the device overview screen. See chapter 6.2.8. |

### 6.2.1 User Interface

The ViSolution user interface is structured as follows:



Min 0 - 2048 Max

Exposure time [us] 5000

Gain [/10 dB] 90

Enable Median

☐ Enable Median

Figure 6.5 Structure of the user interface

| Position | Designation | Function |
|---|---|---|
| 1 | Menu bar | Use the menu bar to access the sensing modes. |
| 2 | Main area | You can display the image data in the main area. |
| 3 | Configuration window | You can set the sensor-specific parameters for the sensor in this area. |
| 4 | Navigation bar | You can select the desired option from the navigation bar. |

## Controls in the Navigation Bar

Different controls are displayed in the navigation bar depending on whether "Camera" or "Data" is selected.

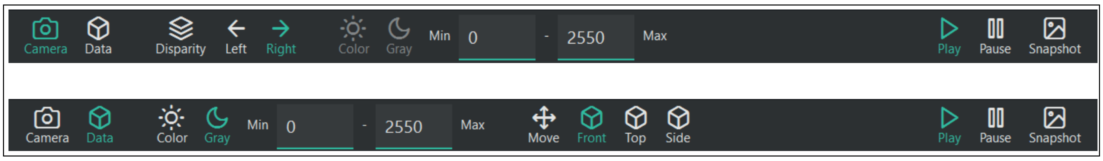
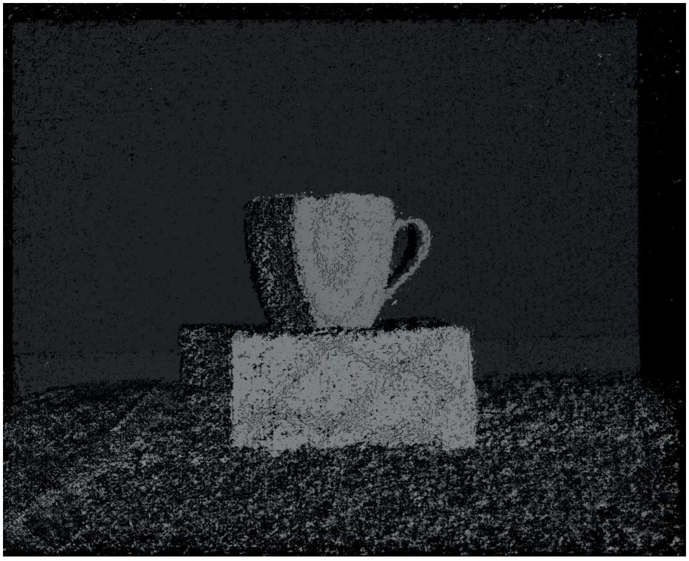
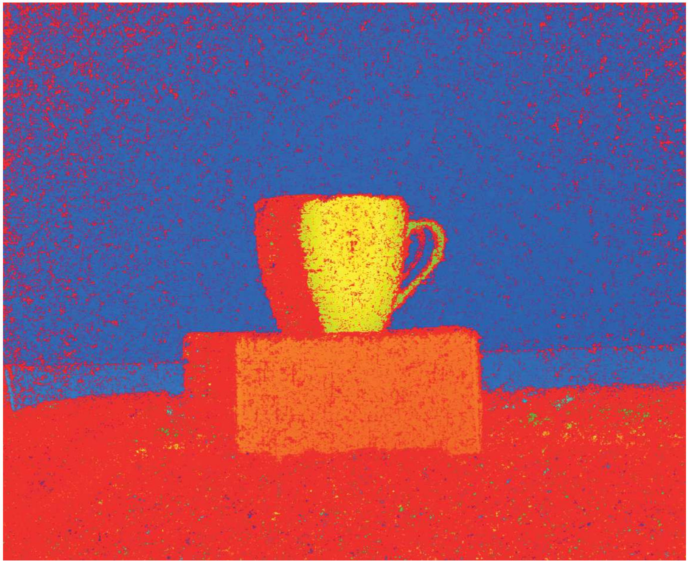
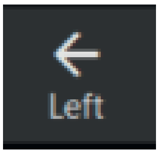
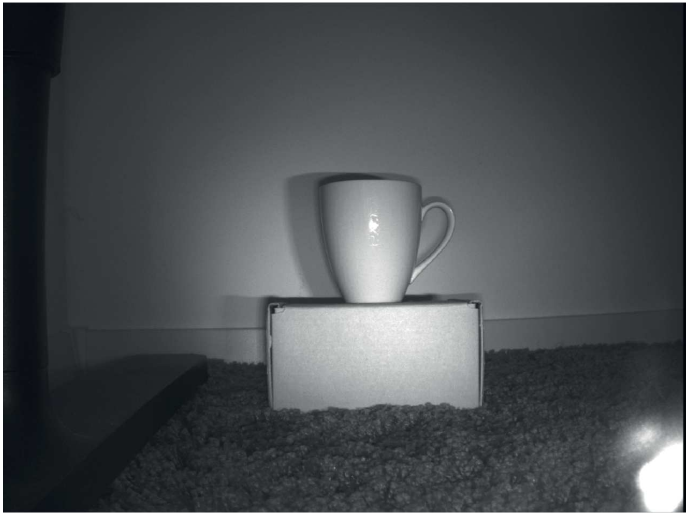
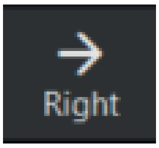
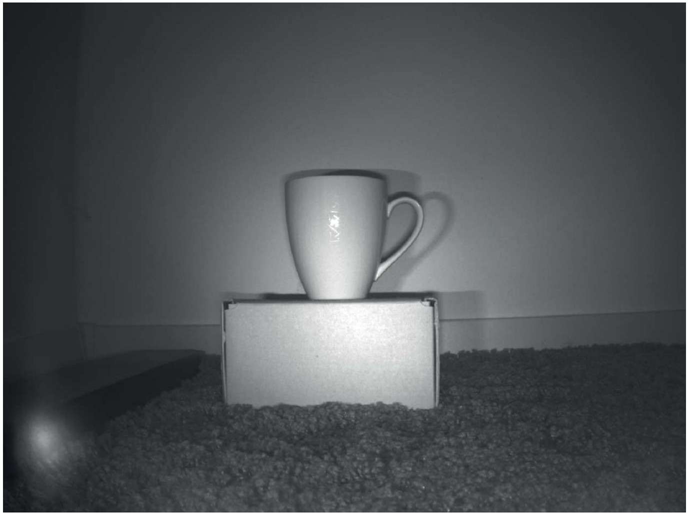
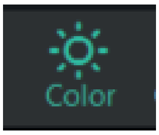
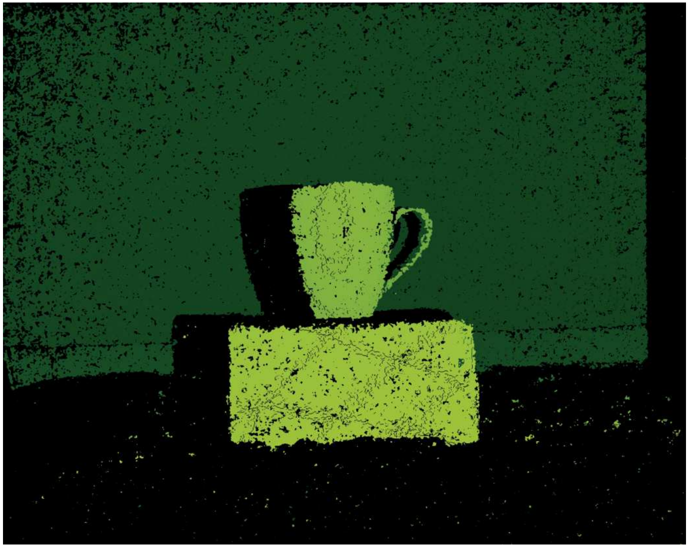
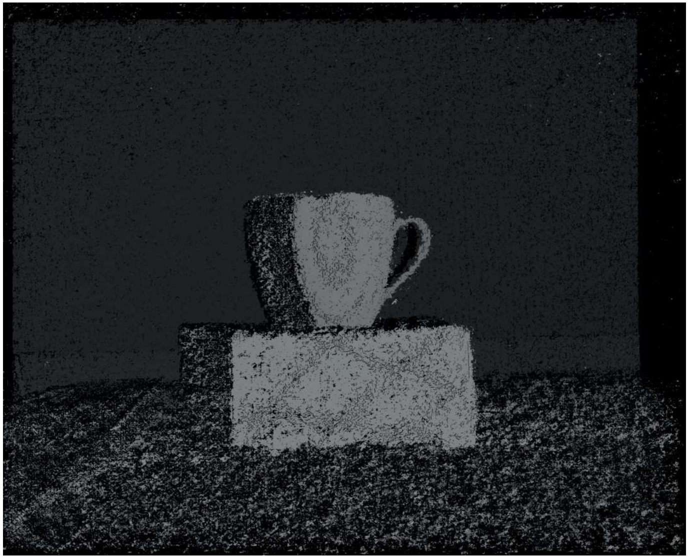
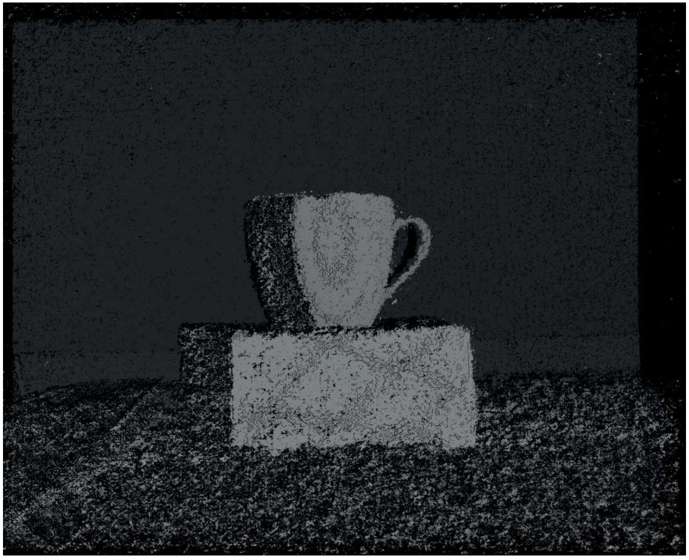


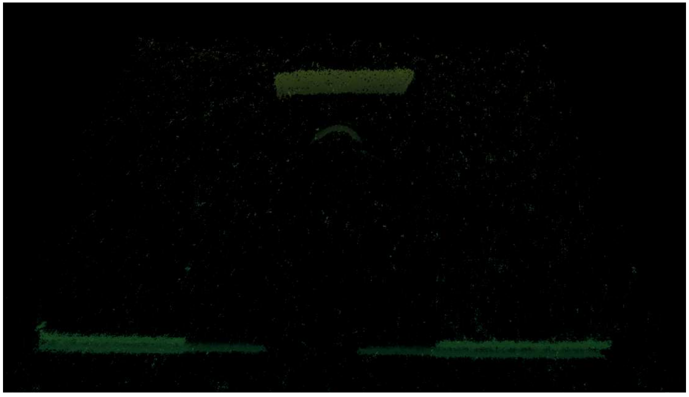
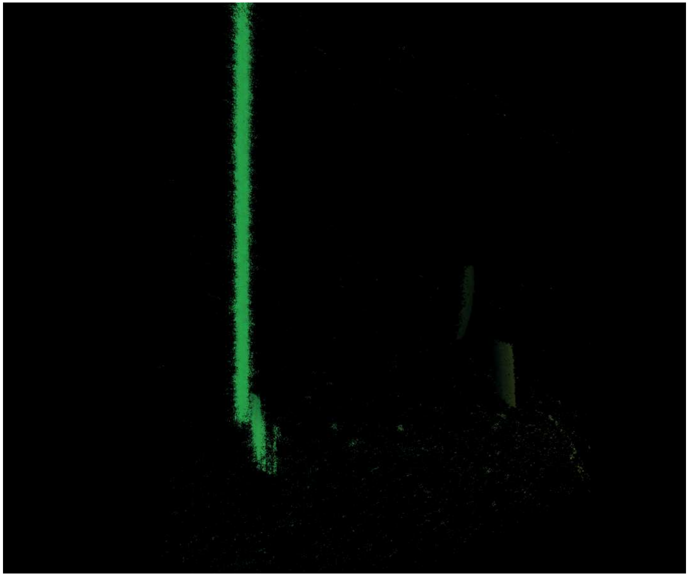
Figure 6.6 Navigation bar

### Controls

| Control | Designation | Function |
|---|---|---|
| Camera | Camera | Switch to camera. Raw images, disparity images, or camera images illuminated with surface lighting can be displayed here. |
| Data | Data | Output of the point cloud<br> |
| Disparity | Disparity | Display of the disparity image<br> |

| Control | Designation | Function |
|---|---|---|
|  | Left | Left image<br> |
|  | Right | Right image<br> |
|  | Color | Switch from grayscale to color image.<br> |

| Control | Designation | Function |
|---|---|---|
| Gray | Gray | Switch from color image to grayscale.<br> |
| Move | Move | You can move the 3-D view freely. |
| Front | Front | 3-D view settings: front view<br> |

| Control | Designation | Function |
|---|---|---|
| Top | Top | 3-D view settings: top view  |
| Side | Side | 3-D view settings: side view  |
| ▷ Play | Play | Click to start a recording. |
| Pause | Pause | Click to stop the recording. |
| Snapshot | Snapshot | Single image capture when the sensor is in the "Pause" position. |
| Save | Save | Non-volatile storage of parameters on the sensor. |

### 6.2.2 Set up

The "Set up" menu allows you to set up and manage the sensor and its tasks.

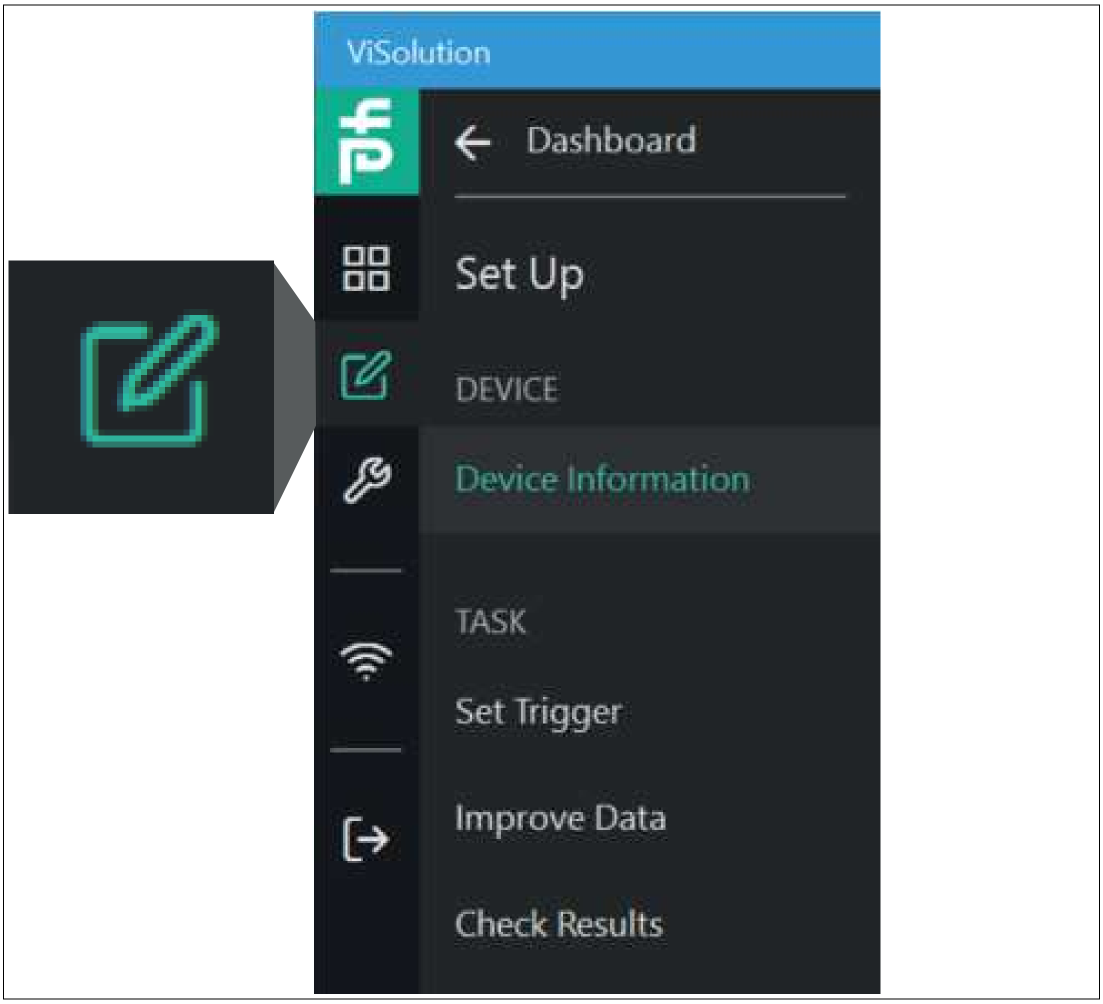


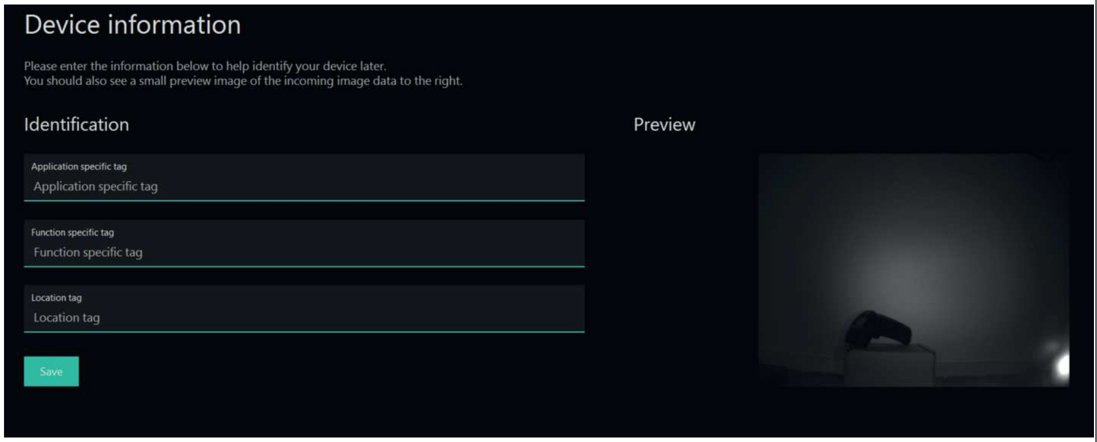
Figure 6.7 Set up

**Link to chapter**

| Area | Function | Link |
|---|---|---|
| DEVICE | Device Information | See chapter 6.2.2.1 |
| TASK | Set Trigger | See chapter 6.2.2.2 |
|  | Improve Data | See chapter 6.2.2.3 |
|  | Check Results | See chapter 6.2.2.4 |

### 6.2.2.1 Device Information

Use "Set up" > "Device information" to access the "Device information" window. A preview image of the camera shot and basic information about the sensor are displayed.



Device information

Please enter the information below to help identify your device later.
You should also see a small preview image of the incoming image data to the right.

Figure 6.8 Device information

Enter the following information to identify your sensor at a later time. Make sure that the preview image of the incoming image data appears on the right.

| Information | Description |
|---|---|
| Application specific tag | Application-specific tag |
| Function specific tag | Function-specific tag |
| Location tag | Location tag |
| Preview | A live preview of the camera shot is displayed in the right window range. |

### 6.2.2.2 Set Trigger

Use "Set up" > "Set Trigger" to access "Set Trigger." The trigger signal starts an event. It triggers the capture and evaluation of images. You can set the trigger mode and trigger frequency in the user interface, under "Set Trigger" in the configuration window.

There are different ways to trigger object detection by the camera. An external signal can be used to trigger the individual objects. The camera can also be configured to capture objects at regular intervals without any external trigger signal.

The sensor distinguishes between different trigger sources:

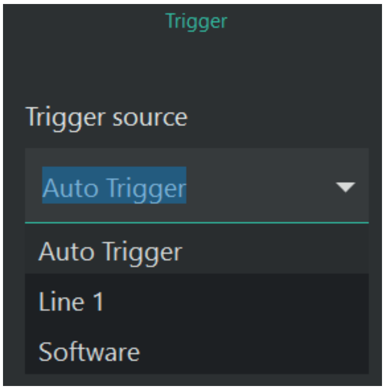
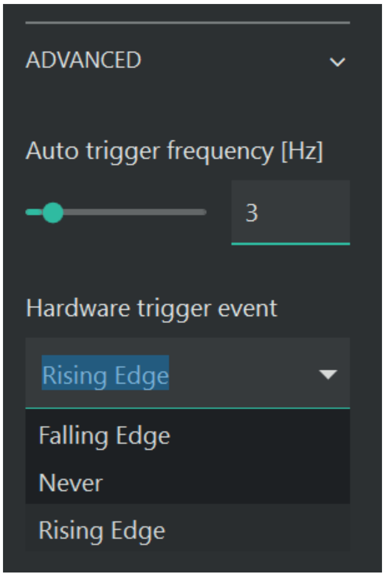


Figure 6.9 Trigger sources

#### Trigger source

| Option | Description |
|---|---|
| Auto Trigger | The sensor captures images at a preset frequency. |
| Line 1 | External trigger via hardware input "IN Trigger." |
| Software | The sensor is triggered via the process interface. |

Additional options are available via the advanced settings:



Auto trigger frequency [Hz]
3

Hardware trigger event
Rising Edge

Figure 6.10 Advanced settings

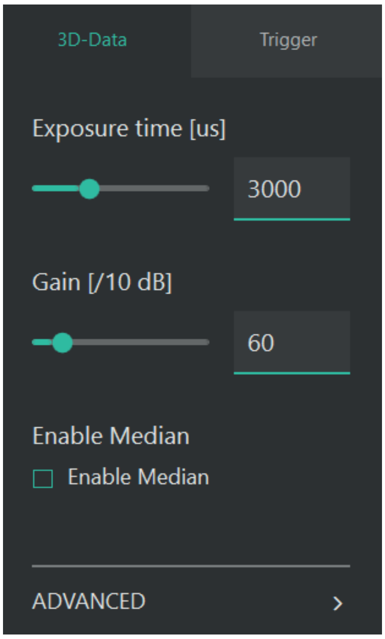
### Advanced settings

| Option | | Description |
|---|---|---|
| Auto trigger frequency [Hz] | | If you select "Auto Trigger" as the trigger source, you can set the frame rate in Hz. The "Auto trigger frequency" slider can be used to change the frame rate or enter it in the display window. The maximum frame rate that can be set is 10 Hz. |
| Hardware trigger event | Falling Edge | The sensor is triggered by hardware via the rising edge of an input signal. |
| | Never | Hardware trigger input is disabled. |
| | Rising Edge | The sensor is triggered by hardware via the falling edge of an input signal. |

### 6.2.2.3 Improve Data

Use "Set up" > "Improve Data" to access the "Improve Data" window.

## Default Settings



Exposure time [us]

3000

Gain [/10 dB]

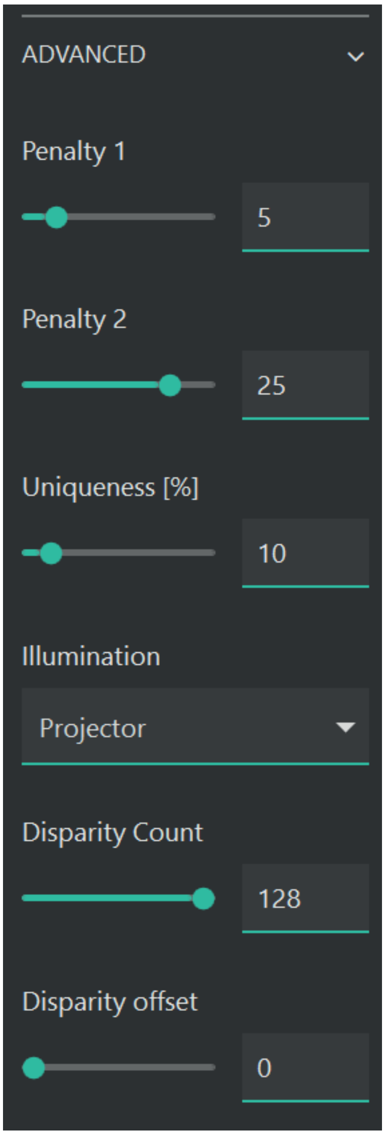
60

Enable Median

□ Enable Median

| Option | Description |
|---|---|
| Exposure Time in µs | Exposure times are displayed in µs. The value for the exposure time can be changed with the slider or entered in the display window. |
| Gain | The gain is used if the image is too dark and if you cannot increase the exposure time further because the object is moving. To increase the image brightness and contrast, the pixel values read out are amplified in a similar manner. The gain is expressed in decibels [/dB]. The higher the gain, the more image noise is present. |
| Enable Median | Point cloud is smoothed by a median filter. |

## Advanced Settings



ADVANCED

Penalty 1
5

Penalty 2
25

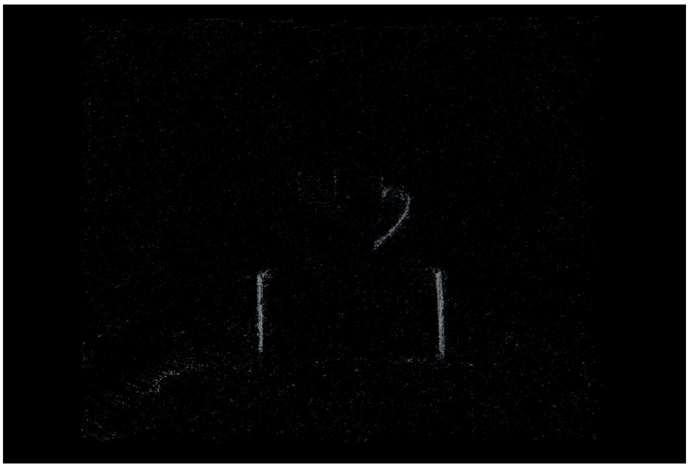
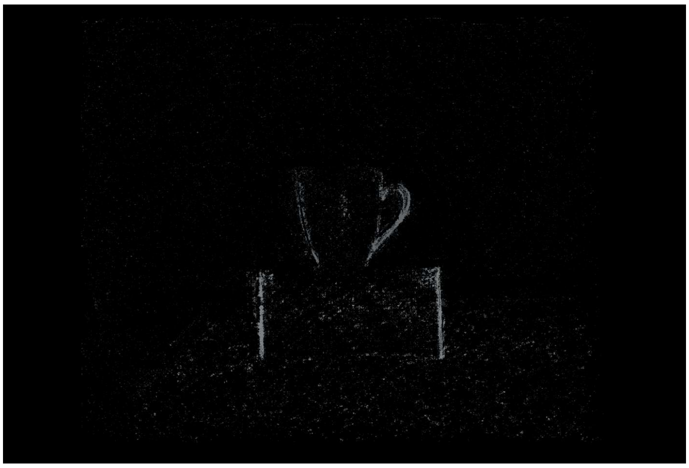
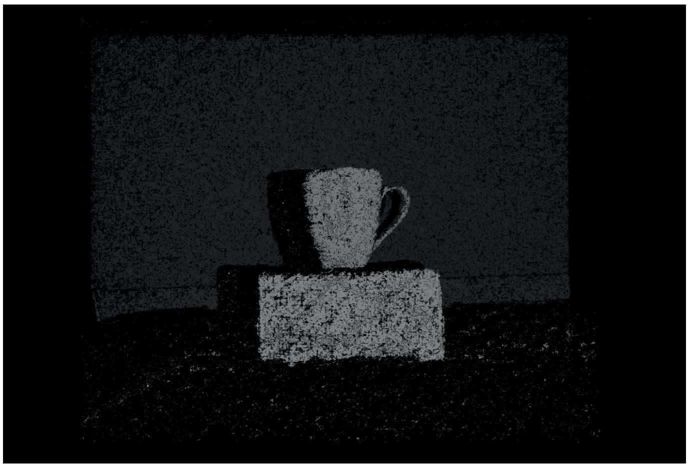
Uniqueness [%]
10

Illumination
Projector

Disparity Count
128

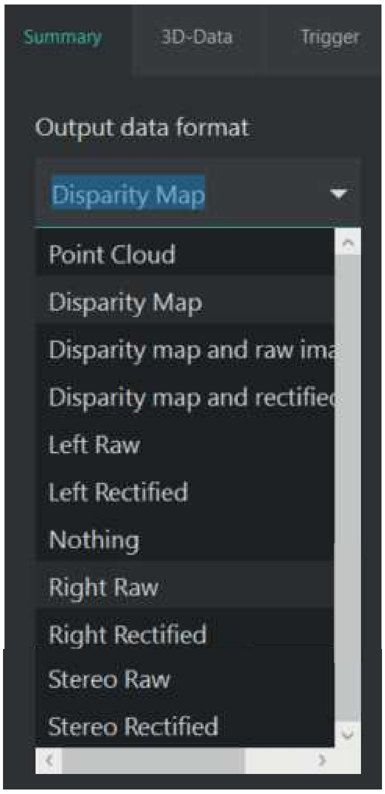
Disparity offset
0

| Option | Description |
|---|---|
| Penalty 1 | Tuning parameter (effort to smooth a disparity jump of 1 in the output) |
| Penalty 2 | Tuning parameter (effort to smooth a disparity jump of more than 1 in the<br>output) |
| Uniqueness | Tuning parameter (maximum relative effort for the best disparity compared to the second best disparity, in percent) |

| Option | | Description |
|---|---|---|
| Illumination | None |  |
| | Onboard |  |
| | Projector |  |

### 6.2.2.4 Check Results

Use "Set up" > "Improve Data" to access the "Check Results" window.



Output data format

Disparity Map

Output data format

| Output data format | Description |
|---|---|
| Point Cloud | Point cloud |
| Disparity Map | Disparity map |
| Disparity map and raw images | Disparity map + raw images |
| Disparity map and rectified images | Disparity map + rectified (corrected) raw images |
| Left Raw | Left camera raw image |
| Left Rectified | Left camera rectified image |
| Nothing | Nothing |
| Right Raw | Right camera raw image |
| Right Rectified | Right camera rectified image |
| Stereo Raw | Right + left raw image |
| Stereo Rectified | Right + left raw image |

### 6.2.3 Monitor

The "Monitor" menu allows you to monitor the performance of the sensor during operation and to optimize it if necessary. This menu is not available in the current software version.

### 6.2.4 Maintain

You can use the Maintain function to check the status of the sensor and keep it up to date.

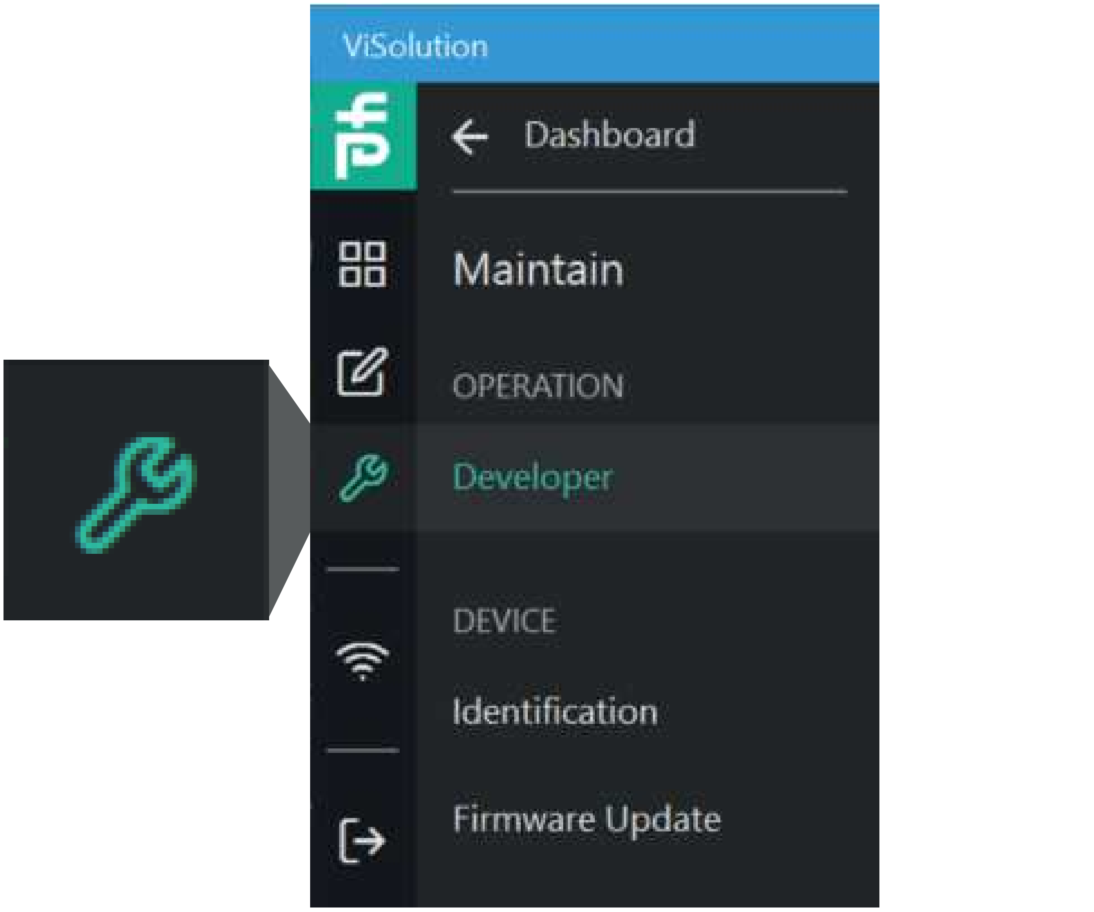


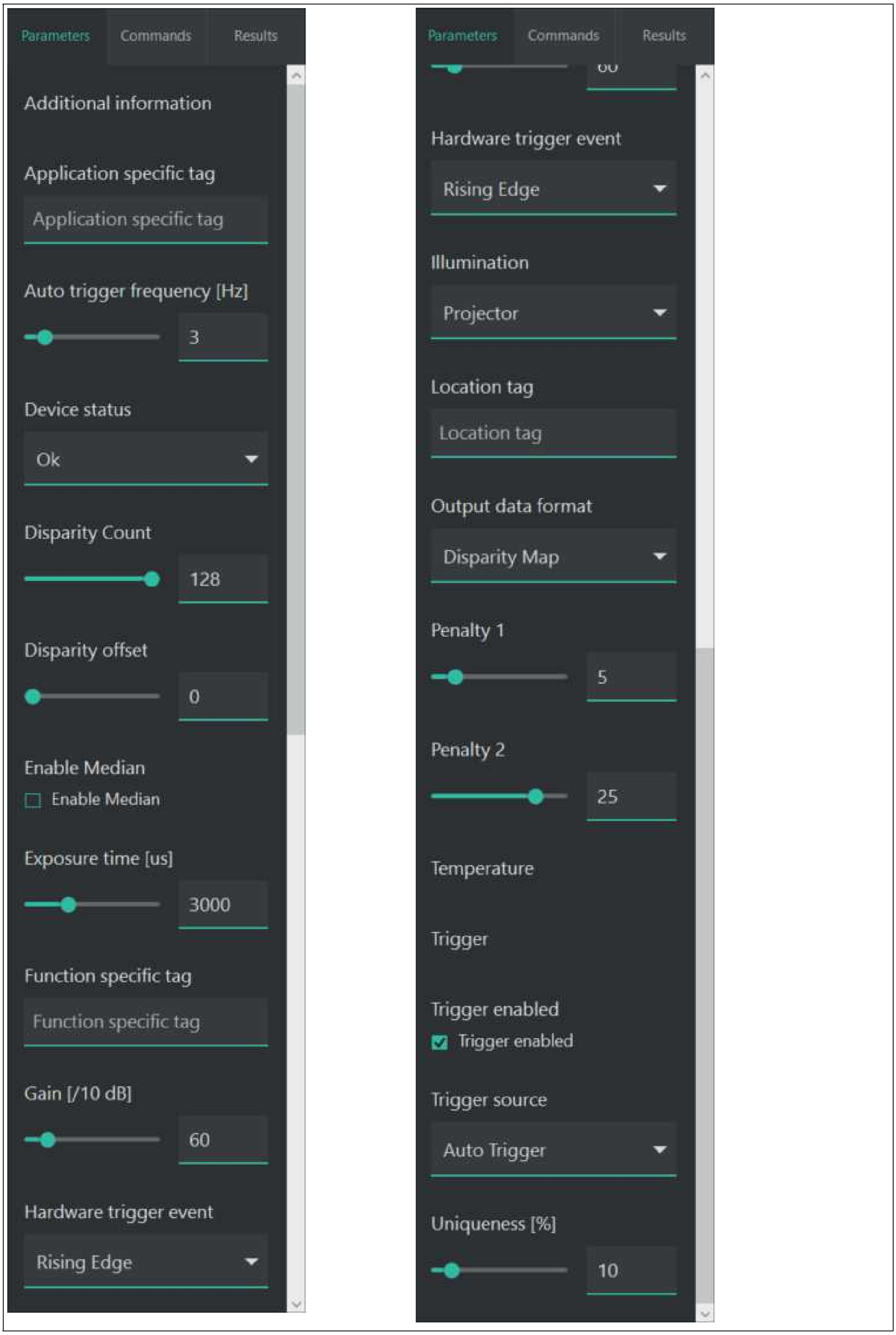
Figure 6.11 Maintain

**Link to chapter**

| Area | Function | Link |
|---|---|---|
| OPERATION | Developer | See chapter 6.2.4.1. |
| DEVICE | Identification | See chapter 6.2.4.2. |
|  | Firmware Update | See chapter 6.2.4.3. |

### 6.2.4.1 Developer

Use "Maintain" > "Developer" to access the "Parameters" overview. This area is specifically designed for developers whose specialist knowledge makes them sufficiently qualified to work on information technology systems.



Additional information

Application specific tag
Application specific tag

Auto trigger frequency [Hz]
3

Device status
Ok

Disparity Count
128

Disparity offset
0

Enable Median
□ Enable Median

Exposure time [us]
3000

Function specific tag
Function specific tag

Gain [/10 dB]
60

Hardware trigger event
Rising Edge

60

Hardware trigger event
Rising Edge

Illumination
Projector

Location tag
Location tag

Output data format
Disparity Map

Penalty 1
5

Penalty 2
25

Temperature

Trigger

Trigger enabled
☑ Trigger enabled

Trigger source
Auto Trigger

Uniqueness [%]
10

Figure 6.12 Parameter overview

| Parameter | Description |
|---|---|
| Application specific tag | Application-specific tag |
| Auto trigger frequency [Hz] | If you select "Auto Trigger" as the trigger source, you can set the frame rate in Hz. The "Auto trigger frequency" slider can be used to change the frame rate or enter it in the display window. The maximum frame rate that can be set is 10 Hz. |
| Device status | Device information |
| Enable Median | Point cloud is smoothed by a median filter. |
| Exposure time [us] | Exposure times are displayed in µs. The value for the exposure time can be changed with the slider or entered in the display window. |
| Function specific tag | Function-specific tag |
| Gain [/10 dB] | The gain is used if the image is too dark and if you cannot increase the exposure time further because the object is moving. To increase the image brightness and contrast, the pixel values read out are amplified in a similar manner. The gain is expressed in decibels [/dB]. The higher the gain, the more image noise is present. |
| Hardware trigger event | The sensor is triggered via the rising/falling/deactivated edge of an input signal. |
| Illumination | Light source to be used |
| Location tag | Location tag |
| Output data format | Output data format, see chapter 6.2.2.4. |
| Penalty 1 | Tuning parameter (effort to smooth a disparity jump of 1 in the output) |
| Penalty 2 | Tuning parameter (effort to smooth a disparity jump of more than 1 in the output) |
| Temperature | Temperature information |
| Trigger | Trigger |
| Trigger enabled | Enable trigger |
| Trigger source | Specifies the internal signal or physical input line to be used as the trigger source. |
| Uniqueness [%] | Tuning parameter (maximum relative effort for the best disparity compared to the second best disparity, in percent) |

### 6.2.4.2 Identification

Use "Maintain" > "Identification" to access the "Identification" window.

| Information | Description |
|---|---|
| Application specific tag | Application-specific tag |
| Function specific tag | Function-specific tag |
| Location tag | Location tag |

**Identification parameters**

| ID card | Description |
|---|---|
| VendorName | Manufacturer |
| VendorURL | Manufacturer's URL |
| ProductName | Product name |
| PartNumber | Product number |
| ProductClass | Product class |
| SerialNumber | Serial number |
| FirmwareVersion | Firmware version |
| HardwareVersion | Hardware version |
| ProtocolStackRevision | Protocol stack revision |
| UniqueProductID | Unique product ID |

### 6.2.4.3 Firmware Update

Use "Maintain" > "Firmware Update" to access the "Firmware Update."

#### Updating Firmware

To update the firmware on your device, proceed as follows:

1. Select a valid update file.
2. Start the update process.

↳ The selected file is uploaded to the sensor. Once it is uploaded, the update will start automatically.

**Note**

The update process may take a few minutes. Do not switch off the sensor during this time.

### 6.2.5 Explore Device

The "Explore Device" menu allows you to gain an initial impression of the sensor without configuring the sensor. This menu is not available in the current software version.

### 6.2.6 Transfer Configuration

The "Transfer Configuration" menu allows you to manage configurations across multiple sensors. This menu is not available in the current software version.

### 6.2.7 Check Connectivity

Manage connections to the network.

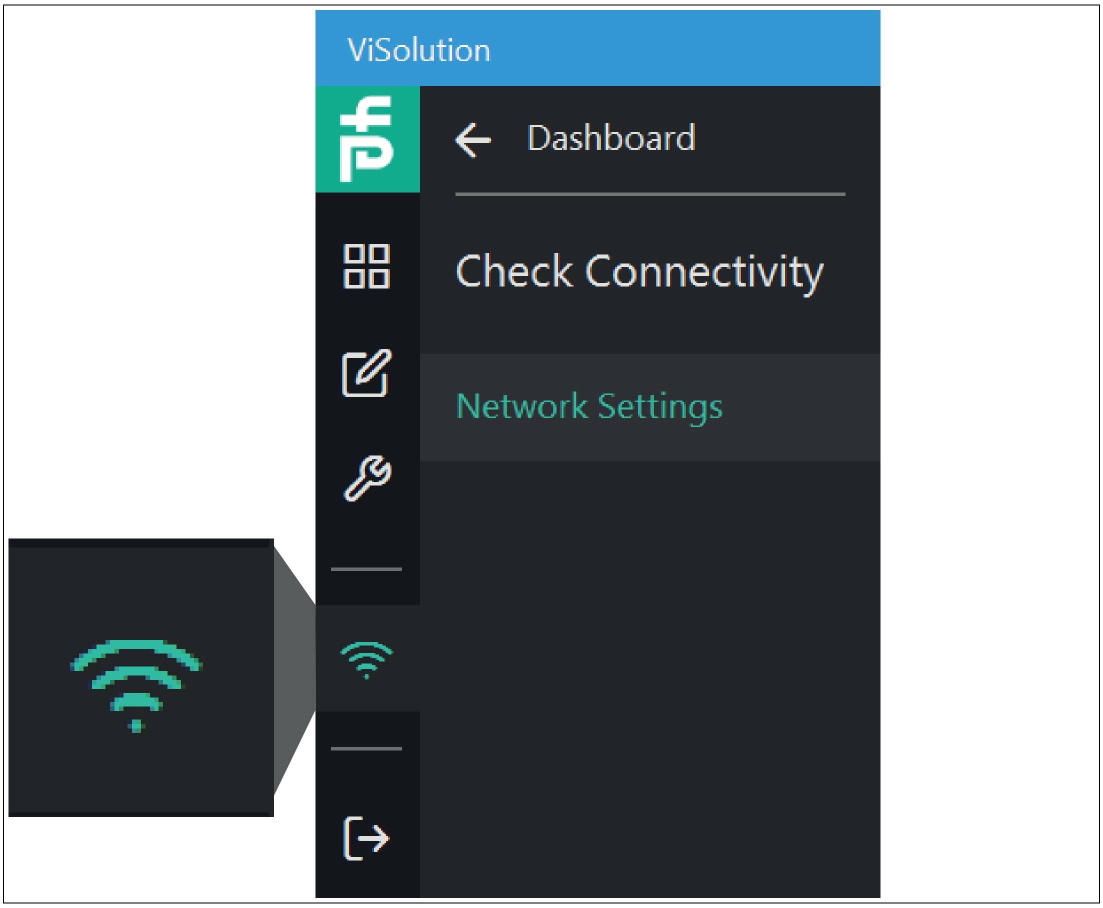


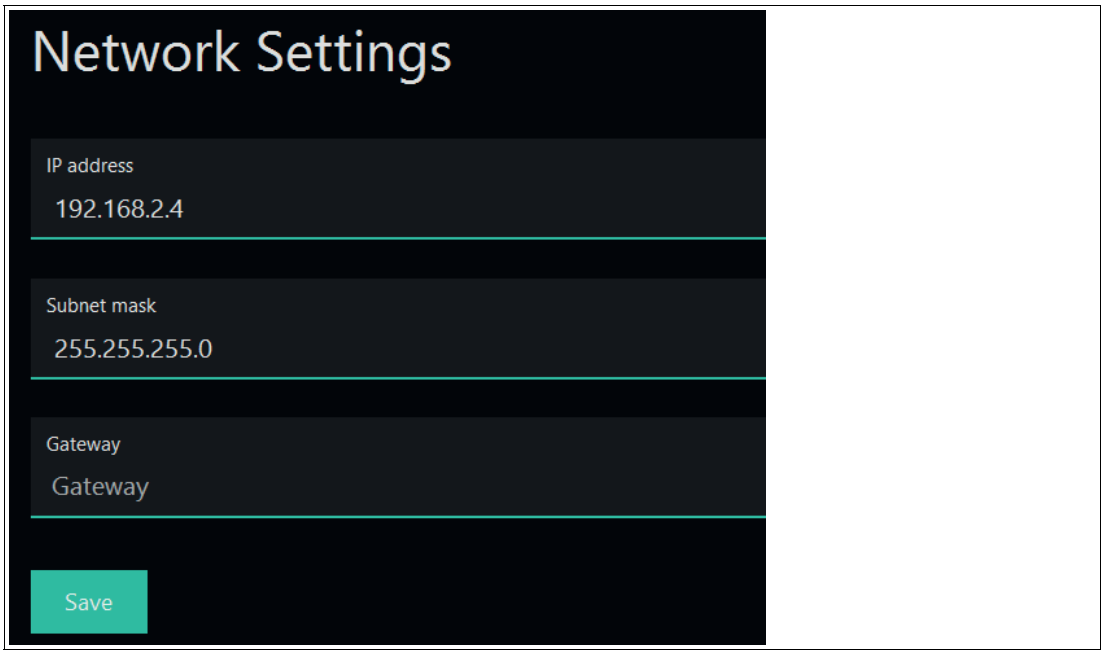
Figure 6.13 Check Connectivity

**Link to chapter**

| Function | Link |
|---|---|
| Network Settings | See chapter 6.2.7.1. |

### 6.2.7.1 Network Settings

Use "Check Connectivity" > "Network Settings" to access the "Network Settings" window.



Network Settings

IP address: 192.168.2.4

Subnet mask: 255.255.255.0

Gateway: Gateway

Figure 6.14 Network settings

| Information | Description |
|---|---|
| IP address | IP address setting |
| Subnet mask | Subnet mask setting |
| Gateway | Gateway setting |

### 6.2.8 Logout

Use "Logout" to exit the user interface and access the device overview screen (see "Device Overview" on page 34).

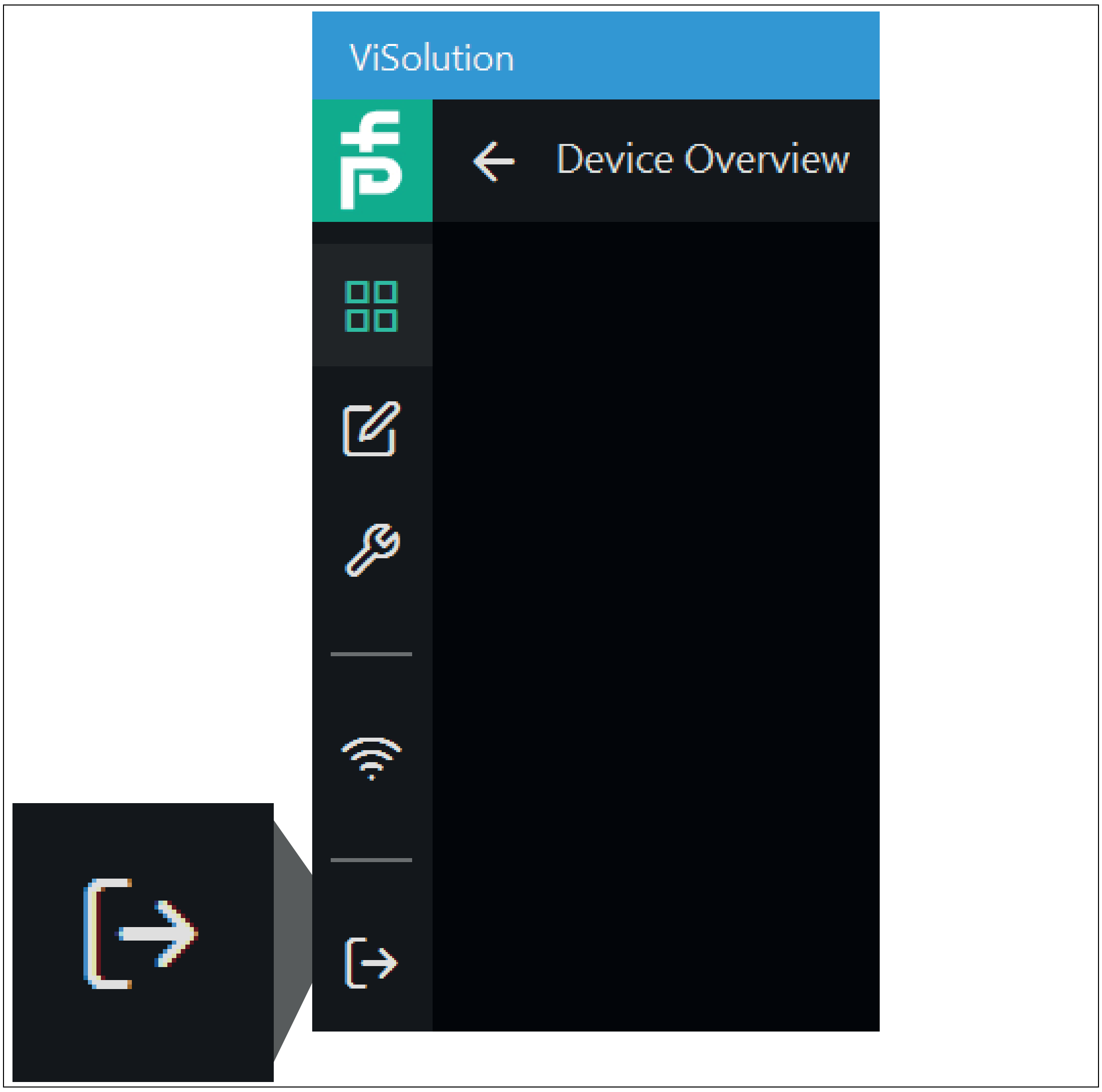


Figure 6.15 Logout

# 7 Maintenance and Repair

## 7.1 Servicing

**Danger!**

Danger to life due to electrical current!

Contact with live parts causes immediate danger to life.

- Allow only qualified electricians to carry out work on the electrical installation.
- Switch off the power supply before carrying out servicing, cleaning, and repairs, and prevent the supply from being switched on again.
- Keep the live parts free from moisture.

The device is maintenance-free. To get the best possible performance out of your device, keep the optical unit on the device clean, and clean it when necessary.

Observe the following instructions when cleaning:

- Do not touch the optical unit with your fingers.
- Do not immerse the device in water. Do not spray the device with water or other liquids.
- Do not use abrasive agents to clean the surface of the device.
- Use a cotton or paper cloth moistened (not soaked) with water or isopropyl alcohol.
- Remove any residual alcohol using a cotton or paper cloth moistened (not soaked) with distilled water.
- Wipe the device surfaces dry using a lint-free cloth.

## 7.2 Repair

The device must not be repaired, changed, or manipulated. In case of failure, always replace the device with an original device.

# 8 Troubleshooting

## 8.1 What to Do in Case of a Fault

Before you have the device repaired, perform the following actions:

- Test the system according to the checklist below.
- Contact our service center to pinpoint the problem.

### Checklist

| Fault | Cause | Remedy |
|---|---|---|
| "PWR" LED does not light up | The power supply is switched off | Check whether there is a reason why the power supply is switched off (installation or maintenance work, etc.). Switch on the power supply if necessary. |
| | Wiring error, cable break | Check the wiring carefully and repair any faults with the wiring. Check the cables to ensure that they are functioning properly. See chapter 3.5. |
| Control panel receiving no measurement data | Connection cable not connected | Connect the connection cable. |
| | Incorrect connection cable used | Use the appropriate connection cable only. See chapter 2.6. |
| | Sensor incorrectly parameterized | Make sure that you have set the correct parameterization. See chapter 6. |
| Test object not detected | Incorrect operating distance | Set the operating distance; see chapter 3.3 |
| | Front panel dirty | Clean the front panel. See chapter 7.1 . |
| | Reflections | Avoid reflections. |
| | Foreign exposure | Avoid foreign exposure. |
| Measurement errors | Test object outside the sensing range | Better positioning of the test object. |
| | Incorrect distance to the test object | Observe the distance values; see chapter 3.3 |
| | Housing incorrectly mounted | Mount the housing correctly; |
| No connection to the sensor | Alternating-current voltage or supply voltage too high | Connect the sensor to direct-current voltage (DC) only. Ensure that the supply voltage level is within the specified sensor range. |
| | Network settings | Set up Windows network communication between the sensor and a PC/laptop; see chapter 3.6 |
| | | Due to the large amount of data supplied by the sensor per second, the sensor(s) must be connected to the PC via a Gigabit Ethernet network. |

- If none of the above remedies the problem, please contact our service center. Please have the error images and the firmware version number available. The firmware version number can be found at the top right of the user interface.

# Your automation, our passion.

## Explosion Protection

- Intrinsic Safety Barriers
- Signal Conditioners
- FieldConnex® Fieldbus
- Remote I/O Systems
- Electrical Ex Equipment
- Purge and Pressurization
- Industrial HMI
- Mobile Computing and Communications
- HART Interface Solutions
- Surge Protection
- Wireless Solutions
- Level Measurement

## Industrial Sensors

- Proximity Sensors
- Photoelectric Sensors
- Industrial Vision
- Ultrasonic Sensors
- Rotary Encoders
- Positioning Systems
- Inclination and Acceleration Sensors
- Fieldbus Modules
- AS-Interface
- Identification Systems
- Displays and Signal Processing
- Connectivity

**Pepperl+Fuchs Quality**
Download our latest policy here:

**www.pepperl-fuchs.com/quality**

PEPPERL+FUCHS

**www.pepperl-fuchs.com**