SENSOPART

# VISOR®

## User Manual

Software version 2.2

## Copyright (English)

No part of this document may be reproduced, published, or stored in databases or information retrieval systems in any form – even in part – nor may illustrations, drawings, or the layout be copied without prior written permission from SensoPart Industriesensorik GmbH.

We accept no responsibility for printing errors or mistakes which occurred in drafting these document. Subject to delivery and technical alterations.

First publication 01 / 2019

SensoPart Industriesensorik GmbH
Nägelseestr. 16
79288 Gottenheim

IND. CONT. EQ.
77HA
For use in class 2 circuits
For use in NFPA 79 Applications only

## Open Source Licenses

The VISOR® software makes use of third party software packages that come under various licenses. This section is meant to list all these packages and to give credit to those whose code helped in the creation of the VISOR® software.

For components that reference the GNU General Public License (GPL) or the GNU Lesser General Public License (LGPL), please find these licenses in this software installation in \SensoPart\VISOR Vision Sensor\Eula\OpenSourceLicenses.

For at least three years from the date of distribution of the applicable product or software, we will give to anyone who contacts us using the contact information provided below, for a charge of no more than our cost of physically distributing, a complete machine-readable copy of the corresponding source code for the version that we may have distributed to you.

The packages containing the source code and the licenses for all of the open-source software are available upon request.
Contact information for requesting source code:
Email: Open.Source@sensopart.de

### VISOR® Firmware version bigger than V2.0

This software uses Linux Version 4.14 (Website: www.kernel.org), which is distributed under the GNU GPL version 2

This software is based on pugixml library (http://pugixml.org). pugixml is Copyright © 2006-2018 Arseny Kapoulkine.

This software makes use of Nlohmann JSON library, which is distributed under the MIT License. Copyright © 2013-2018 Niels Lohmann.

This software makes use of libcrypto, which is distributed under the openssl license. Copyright OpenSSL © 2018.

This software makes use of libcurl, which is distributed under the MIT license. Copyright (c) 1996 - 2018, Daniel Stenberg, daniel@haxx.se, and many contributors.

This software makes use of busybox, which is distributed under the GNU GPL version 2.

This software makes use of e2fsprogs which is distributed under the GNU GPL version 2 License, GNU LGPL version 2.1 License, BSD License and MIT License.

This software makes use of eudev, which is distributed under the GNU GPL version 2 License and GNU LGPL version 2.1 License or later.

This software makes use of glibc, which is distributed under the GNU LGPL version 2.1 License or later.

This software makes use of GNU MP library, which is distributed under the GNU GPL version 2 License.

This software makes use of libcap, which is distributed under the GNU LGPL version 2.1 License or later.

This software makes use of libidn, which is distributed under the GNU LGPL version 2.1 License or later.

This software is based in part on the work of the Independent JPEG Group. This software makes use of libjpeg-turbo, which is distributed under the Modified (3-clause) BSD License.

This software makes use of libnl, which is distributed under the GNU LGPL version 2.1 License or later.

This software makes use of libssh2, which is distributed under the BSD license.

This software makes use of libssh2, which is distributed under the openssl license.

This software makes use of libstdc++, which is distributed under the GPL-3.0-with-GCC-exception license.

This software makes use of libxml2, which is distributed under the MIT license.

This software makes use of lighttpd, which is distributed under the BSD 3-clause license. Copyright (c) 2004, Jan Kneschke, incremental.

This software makes use of mtd-utils, which is distributed under the GNU GPL version 2 License.

This software makes use of net-snmp-libs, which is distributed under the BSD-License License. Copyright (c) 2001-2003, Networks Associates Technology, Inc

This software makes use of netbase, which is distributed under the GNU GPL version 2 License.

This software makes use of OpenSSH, which is distributed under the OpenSSH License.

This software makes use of OpenSSL, which is distributed under the OpenSSL License.

This software makes use of GnuPG Version 1.4.10 ( Website: https://www.gnupg.org/ ),which is distributed under the GNU GPL version 3 or higher.

This software makes use of netbase, which is distributed under the GNU GPL version 2 License U-Boot exception 2.0 license.

### VISOR® Firmware version less than V2.0

The VISOR® firmware makes use of Linux Version 2.6.33 (Website: www.kernel.org), which is distributed under the GNU GPL version 2.

The VISOR® firmware makes use of x-loader, an initial program loader for Embedded boards based on OMAP processors (Website: http://arago-project.org/git/projects/?p=x-load-omap3.-git; a=summary) which is distributed under the GNU GPL version 2 or higher.

The VISOR® firmware makes use of u-boot, an initial program loader for Embedded boards based on OMAP processors (Website: http://arago-project.org/git/projects/?p=x-load-omap3.-git; a=summary) which is distributed under the GNU GPL version 2 or higher

The VISOR® firmware makes use of spike Version 0.2,a SPI-driver (Website: https://-github.com/scottellis/spike/blob/master/spike.c ), which is distributed under the GNU GPL version 2 or higher.

The VISOR® firmware makes use of Busy-Box Version 1.18.1 ( Website: http://www.busy-box.net/ ), which is distributed under the GNU GPL version 2 or higher

The VISOR® firmware makes use of vsftpd Version 2.0.3 ( Website: https://se-curity.appspot.com/vsftpd.html ), which is distributed under the GNU GPL version 2 or higher.

The VISOR® firmware makes use of mtd-utils Version 1.5.0 ( Website: http://www.linux-mtd.in-fradead.org/doc/general.html ), which is distributed under the GNU GPL version 2 or higher.

The VISOR® firmware makes use of Boa Webserver Version 0.94.13 ( Website: http://www.-boa.org/ ), which is distributed under the GNU GPL version 2 or higher.

The VISOR® firmware makes use of Procps Version 3.2.8 ( Website http://-procps.sourceforge.net/download.html), which is distributed under the GNU GPL version 2 or higher and GNU LGPL version 2.1 or higher.

The VISOR® firmware makes use of GnuPG Version 1.4.10 ( Website: https://www.gnupg.org/ ), which is distributed under the GNU GPL version 3 or higher.

The VISOR® firmware makes use of glibc, which is distributed under GNU LGPL version 2.1 or higher.

The VISOR® firmware makes use of Dropbear - a SSH2 server Version 2012.55 ( Website: https://matt.ucc.asn.au/ dropbear/dropbear.html ). The Dropbear SSH2 server is distributed under the terms of the Dropbear License, which is a MIT/X Consortium style open source license. Please find this license in this software installation in \SensoPart\VISOR Vision-Sensor\Eula\OpenSourceLicenses

### VISOR® PC Software

SensoConfig software is based in part on the work of the Qwt project (http://qwt.sf.net).

SensoFind, SensoConfig, SensoRescue software is based in part on the work of the Qt-project (https://doc.qt.io/qt-5/licenses-used-in-qt.html).

SensoCalc uses icons from material-design-icons (https://material.io/tools/icons), which are distributed under the Apache License Version 2.0.

SensoCalc uses NodeJS, which is distributed under the MIT license. Copyright (C) 2012-2018 by various contributors.

SensoCalc uses electron and electron-store, which is distributed under the MIT license. Copyright (c) 2013-2019 GitHub Inc.

SensoCalc uses AngularJS, which is distributed under the MIT license. Copyright (c) 2010-2018 Google, Inc. http://angularjs.org.

SensoCalc uses fs-jetpack, which is distributed under the MIT license.

SensoCalc uses asar, which is distributed under the MIT license. Copyright (c) 2014 GitHub Inc.

SensoCalc uses clipboard.js, which is distributed under the MIT license. Copyright © 2019 Zeno Rocha <hi@zenorocha.com>

SensoCalc uses roboto font, which is distributed under the MIT license.

SensoCalc uses gulp, which is distributed under the MIT license. Copyright (c) 2013-2018 Blaine Bublitz <blaine.bublitz@gmail.com>, Eric Schoffstall <yo@contra.io> and other contributors.

SensoCalc uses rcedit, which is distributed under the MIT license.

SensoCalc uses q, which is distributed under the MIT license. Copyright 2009–2017 Kristopher Michael Kowal and contributors.

SensoFind uses libgpg-error, which is distributed under the GNU LGPL version 2.1 License or later. Copyright (C) 2012-2017 g10 Code GmbH.

SensoFinduses Libgcrypt, which is distributed under the GNU LGPL version 2.1 License or later. Copyright (C) 1989,1991-2017 Free Software Foundation, Inc. Copyright (C) 2012-2017 g10 Code GmbH. Copyright (C) 2013-2017 Jussi Kivilinna.

SensoConfig uses Mesa library, which is licensed according to the terms of the MIT license. Copyright (C) 1999-2007 Brian Paul. All Rights Reserved.

## Table of contents

# 1 Information on this document

## 1.1 Explanation of symbols

### Warnings

**CAUTION / WARNING / DANGER**
This symbol is used to indicate a potentially hazardous situation that, if not avoided, could result in death or serious injury.

**WARNING**
This symbol is used to indicate potentially hazardous situations arising from laser beams.

**ATTENTION:**
This symbol is used to indicate text that must be observed without fail. Failure to do so may result in bodily injury or property damage.

**NOTE:**
This symbol is used to highlight useful tips and recommendations, as well as information intended to help ensure efficient operation.

### Detectors

- Pattern matching
- Contour
- Contrast
- Brightness
- Gray
- Caliper
- BLOB
- Contour 3D
- Barcode
- Datacode
- OCR
- Color Value
- Color List
- Color Area
- Result processing

### Alignment

Alignment
Includes the position detectors: Contour matching, Pattern matching, and Edge detector

## 1.2 Additional documents

The following documents for the VISOR® vision sensor are available for download in the Download area of the SensoPart website.

- VISOR® User Manual
- VISOR® Communications manual
- VISOR® Operating manual

Furthermore, these documents are part of the software installation and can be found in the subfolder "\Documentation\", as well as via the Windows Start menu.

## 1.3 Document version

This manual describes the VISOR® software version 2.2.

Documents for the previous software versions (< 2.2) can be found in the download area of the SensoPart homepage (www.sensopart.com).

# 2 Safety instructions

**WARNING**
The vision sensor is not a safety component pursuant to the EU Machinery Directive. It is strictly prohibited to use it in applications in which the safety of people depends on device functions.

Comply with all applicable local accident prevention regulations and general safety regulations.

Follow all safety instructions and other instructions in the operating manual and in this user manual.

The connection should be made exclusively by trained qualified personnel.

Do not tamper with or make alterations to the unit!

For use with any listed, configured cable cables (CYJV).

**WARNING**
Vision sensors with a laser belong to laser class 1 as defined in IEC 60825-1:2014.
Wavelength: 655 nm, frequency: 9 kHz, pulse width: 2.6 µs, pulse limit: 11 mW.

# 3 Intended use

The VISOR® vision sensor is an optical sensor and is used for the non-contact acquisition /identification of objects. The vision sensor features a number of different evaluation methods (detectors), with the specific methods depending on the specific model sensor. The product is designed for industrial use only. In residential areas, it is possible that additional measures for noise suppression must be taken. The vision sensor is not suitable for use outdoors.

**WARNING**
The vision sensor is not a safety component pursuant to the EU Machinery Directive It is strictly prohibited to use it in applications in which the safety of people depends on device functions.

## 3.1 Field of application

The VISOR® vision sensor is a cost-effective alternative to conventional image processing systems.

**Object:**
The VISOR® vision sensor detects defective parts and parts in the wrong position, angular position, order, or any combination of them with unrivalled precision and accuracy. For testing tasks and evaluations, different detectors are available: e.g. Pattern matching, Contour, Brightness, Gray, Contrast, Caliper or BLOB. The Advanced version of the VISOR® vision sensor features Alignment on top of this. In this way, it is also possible to reliably detect the features which do not appear repetitively in the taught-in position. All evaluations are made relative to the current part position and angular position, without you having to define your own characteristic for each possible position.
The Advanced version also offers calibration for correcting distortion, e.g. for simple measurement tasks.

**Object color:**
The VISOR® Object color offers powerful object detection combined with color detection. This makes it possible to increase the stability of many applications in which there are too few differences in the gray image. In addition, e.g. self-luminous parts such as colored LEDs and "non-colors" such as white and black are detected.

**Code reader:**
The identification of products, components, or packaging on the basis of printed or directly marked – nailed or lasered – codes or plain text is common today in many areas of industry. The Code readerSensoPart of recognizes at a glance what part of it is in front of you: It effortlessly reads barcodes of many types as well as printed and directly marked ECC-200 standard Data matrix codes – and this from any carrier material (metal, plastic, paper, glass). The sensor can even routinely decipher applied codes on oblique, distorted, or convex, reflective, or transparent

surfaces. The code reader evaluates the quality of printed and directly marked Data Matrix codes based on standardized ISO and AIM quality parameters. This makes it possible to take corrective measures early on, preventing scrap resulting from illegible codes. In addition, the sensor can also read directly printed fonts with the detector plain text reading.

**Solar:**
The VISOR® Solar features an optimized inspection algorithm that can be used for the comprehensive quality control of sensitive silicon wafers while they are being produced. The functions relevant to wafer and cell inspection – from wafer geometry detection, location, and position to defect localization, processing speed adjustment, and test accuracy – are pre-configured so that the sensor is ready to go after a few mouse clicks.

**Allround:**
The VISOR® Allround features all the functions of the VISOR® Object, Code reader and Object color combined in a single device. The Professional version also offers the "Multishot" function to detect minute surface defects.

**Robotic**:
All VISOR® Robotic functions are available in the VISOR® Object, as are added robot-specific functions.

## 3.2 Requirements for use

In order to configure the VISOR® vision sensor, a standard PC / notebook (at least 1 GHz processor with support for SSE2 and 1 GB RAM, with Microsoft Windows 8, Windows 8.1 or Windows 10 operating system) with a network connection with RJ-45 port and a network with TCP-IP protocol is required. We recommend a screen resolution of at least 1024 x 768 pixels.

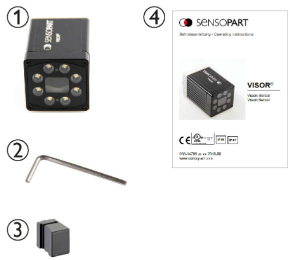
The default network settings for the VISOR® vision sensor are 192.168.100.100 for the IP address, 255.255.255.0 for the subnet mask, and 192.168.100.1 for the gateway.

The VISOR® vision sensor does not need a PC or PLC to run. A PC / laptop is required only in order to configure the VISOR® vision sensor.

Attention must be paid to sufficient and constant object illumination to ensure reproducible results and avoid malfunction. Light reflections or changing extraneous light can distort evaluation results. If necessary, use an external light source and / or light-protection devices to protect against extraneous light / ambient light.

# 4 Scope of delivery and software

## 4.1 Scope of delivery: VISOR® Vision Sensor




The scope of delivery includes:

① VISOR® vision sensor

② Allen key

③ Mounting bracket MK 45

④ Operating manual

Upon receiving the delivery, check it immediately for any transit damage and make sure that it is complete. If there is any transit damage, inform the shipping agent. If returning the sensor, always make sure to pack it in sufficiently sturdy packaging.

## 4.2 VISOR® software

### Download

The VISOR® software Setup is available for download at www.sensopart.com (Download / Software... ).

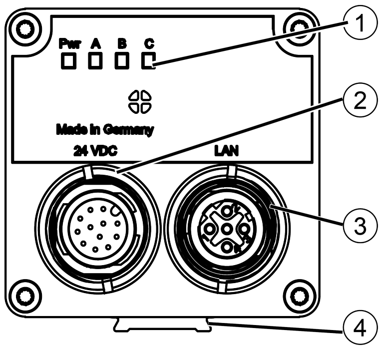
### USB drive

The VISOR® software setup is also available from SensoPart Industriesensorik GmbH on a USB drive with part number 651-01000 (subject to a charge).

# 5 Product identification

## 5.1 Product description

### Overview




① LED display
② M12 connector for supply voltage (24 VDC) and digital I/O.
③ M12 connector socket for Ethernet connection.
④ Dovetail guide

### LED display

All LEDs are driven without taking into account any delays used.


*Fig. 1: LED display*

| Name | Color | Meaning | |
|---|---|---|---|
| Pwr. (Power) | Green | Operating voltage | No errors |
| | Red / Yellow | | No PROFINET connection |
| | Yellow | | No jobset available |
| | Red | | Error / Starting device |
| A | Yellow | Result 1 | |
| B | Yellow | Result 2 | |
| C | Yellow | Result 3 | |

## 5.2 Type key

**V 20 - ALL - P 3 - W - M D - M 2 - L**

① ② ③ ④ ⑤ ⑥ ⑦ ⑧ ⑨ ⑩ ⑪

① **V = VISOR®**

② **Hardware / resolution**

- V10 / V10C: SVGA (800x600), QSVGA (400x300), QSVGA Zoom 2 (400x300)
- V20 / V20C: HDV2 (1440x1080), WGA (720x540), WGA Zoom 2 (720x540)
- V50 / V50C: QSXGA (2560 x 1936), SXVGA (1280 x 968), SXVGA Zoom 2 (1280 x 968)

③ **Sensor type**
ALL = Allround
OB = Object
CR = Code reader
RO = Robotic
SO = Solar

④ **Variant**
S = Standard
A = Advanced
P = Professional

⑤ **Version**

⑥ **Lighting**
W = White LEDs
R = Red LEDs
I = Infrared LEDs

⑦ **Objective**
C = C-Mount
W = Wide
M = Medium
N = Narrow

⑧ **Depth of focus**
" "= Depth of field: Normal
D = Depth of field: Increased

⑨ **Focal point (optional)**
M = Motorized focal point

⑩ **Connections (optional)**
2 = Two connections (1 x I/O, 1 x Ethernet)

⑪ **Laser**

## 5.3 Overview of functions

### VISOR® Object

| | VISOR® Object | |
|---|---|---|
| | **Standard** | **Advanced** |
| **Applications** | Presence, completeness, measuring, color, position control | |
| **Resolution** | | |
| V10 (800 x 600): Mono \| Color | ✓ \| ✓ | |
| Frames per second: Mono \| Color | 75 \| 50 | |
| V20 (1440 x 1080): Mono \| Color | – \| – | ✓ \| ✓ |
| Frames per second: Mono \| Color | – \| – | 40 \| 20 |
| V50 (2560 x 1440): Mono \| Color | – \| – | ✓ \| ✓ |
| Frames per second: Mono \| Color | – \| – | 22 \| 8 |
| **Lighting** | White; for Mono only: Red, Infrared | |
| Multishot (Mono) | – | – |
| Target laser | – | ✓ |
| **Objective** | | |
| V10 Wide \| Medium \| Narrow \| C-Mount | ✓ \| ✓ \| ✓ \| – | ✓ \| ✓ \| ✓ \| ✓ |
| V20 Wide \| Medium \| Narrow \| C-Mount | – \| – \| – \| – | ✓ \| ✓ \| ✓ \| ✓ |
| V50 Wide \| Medium \| Narrow \| C-Mount | – \| – \| – \| – | – \| – \| – \| ✓ |
| **Interfaces** | Ethernet \| EtherNet/IP \| PROFINET | |
| Inputs \| Outputs \| Selectable | 2 \| 2 \| 4 | 2 \| 2 \| 6 |
| Encoder input | – | ✓ |
| Ethernet \| EtherNet/IP | ✓ \| ✓ | ✓ \| ✓ |
| PROFINET \| SensoWeb | ✓ \| ✓ | ✓ \| ✓ |
| Service port | – | ✓ |
| **Job \| Detectors** | | |
| Number of jobs (max.) | 8 | 255 |
| Number of detectors per job (max.) | 32 | 255 |
| **Calibration** | | |
| Scaling (Measurement) | ✓ | ✓ |

| | VISOR® Object | |
|---|---|---|
| | Standard | Advanced |
| Calibration plate (Measurement) | – | ✓ |
| Point pair list (Robotics) | – | – |
| Calibration plate (Robotics) | – | – |
| Hand-Eye calibration (Robotics) | – | – |
| Base-Eye calibration (Robotics) | – | – |
| **Pre-processing** | | |
| Pre-processing filter | – | ✓ |
| Repeat mode | – | ✓ |
| Shutter variation | – | ✓ |
| Free-form search region | ✓ | ✓ |
| **Alignment** | | |
| Contour matching (translation, rotation 360°) | ✓ | ✓ |
| Pattern matching (translation, rotation 360°) | – | ✓ |
| Edge detector (translation, rotation) | – | ✓ |
| **Object detection** | | |
| Contour (translation, rotation 360°)<br>Multiple objects | ✓<br>– | ✓<br>✓ |
| Contour 3D (translation, rotation 360°)<br>Multiple objects | –<br>– | –<br>– |
| Pattern matching (translation, rotation 360°)<br>Multiple objects | ✓<br>– | ✓<br>✓ |
| Gray | ✓ | ✓ |
| Contrast | ✓ | ✓ |
| Brightness | ✓ | ✓ |
| Caliper | ✓ | ✓ |
| BLOB | – | ✓ |

| | VISOR® Object | |
|---|---|---|
| | **Standard** | **Advanced** |
| **Identification** | | |
| Barcode \| Barcode Advanced | – \| – | – \| – |
| Datacode \| Datacode Advanced | – \| – | – \| – |
| OCR | – | – |
| **Robotic functions** | | |
| Result offset Image \| 2D \| 3D | – \| – \| – | – \| – \| – |
| Gripping space check | – | – |
| **Color detectors for Color variants** | | |
| Color Area | ✓ | ✓ |
| Color Value | – | ✓ |
| Color List | – | ✓ |
| Color distance \| binarization | – | ✓ |
| **Result processing** | | |
| Result processing: Text | – | – |
| Result processing: Math | – | ✓ |

**VISOR® Code reader**

| | VISOR® Code reader | | |
|---|---|---|---|
| | **Standard** | **Advanced** | **Professional** |
| **Applications** | Reading barcodes, data codes, text | | |
| **Resolution** | | | |
| V10 (800 x 600): Mono \| Color | ✓ \| – | | |
| Frames per second: Mono \| Color | 75 \| – | | |
| V20 (1440 x 1080): Mono \| Color | ✓ \| – | | |
| Frames per second: Mono \| Color | 40 \| – | | |
| V50 (2560 x 1440): Mono \| Color | – \| – | ✓ \| – | |
| Frames per second: Mono \| Color | – \| – | 22 \| – | |
| **Lighting** | White; for Mono only: Red, Infrared | | |
| Multishot (Mono) | – | – | – |
| Target laser | ✓ (V20 only) | ✓ | ✓ |
| **Objective** | | | |
| V10 Wide \| Medium \| Narrow \| C-Mount | ✓ \| ✓ \| ✓ \| – | ✓ \| ✓ \| ✓ \| ✓ | |
| V20 Wide \| Medium \| Narrow \| C-Mount | ✓ \| ✓ \| ✓ \| – | ✓ \| ✓ \| ✓ \| ✓ | |
| V50 Wide \| Medium \| Narrow \| C-Mount | – \| – \| – \| – | – \| – \| – \| ✓ | |
| **Interfaces** | Ethernet \| EtherNet/IP \| PROFINET | | |
| Inputs \| Outputs \| Selectable | 2 \| 2 \| 4 | 2 \| 2 \| 6 | |
| Encoder input | – | ✓ | ✓ |
| Ethernet \| EtherNet/IP | ✓ \| ✓ | ✓ \| ✓ | ✓ \| ✓ |
| PROFINET \| SensoWeb | ✓ \| ✓ | ✓ \| ✓ | ✓ \| ✓ |
| Service port | – | ✓ | ✓ |
| **Job \| Detectors** | | | |
| Number of jobs (max.) | 8 | 255 | |
| Number of detectors per job (max.) | 2 | 255 | |
| **Pre-processing** | | | |
| Pre-processing filter | – | ✓ | ✓ |
| Repeat mode | – | ✓ | ✓ |
| Shutter variation | – | ✓ | ✓ |
| Free-form search region | – | – | ✓ |

| | VISOR® Code reader | | |
|---|---|---|---|
| | **Standard** | **Advanced** | **Professional** |
| **Alignment** | | | |
| Contour matching (translation, rotation 360°) | – | – | ✓ |
| Pattern matching (translation, rotation 360°) | – | – | ✓ |
| Edge detector (translation, rotation) | – | – | ✓ |
| **Object detection** | | | |
| Contour (translation, rotation 360°)<br>Multiple objects | –<br>– | –<br>– | –<br>– |
| Contour 3D<br>Multiple objects | –<br>– | –<br>– | –<br>– |
| Pattern matching (translation, rotation 360°)<br>Multiple objects | –<br>– | –<br>– | ✓<br>– |
| Gray | – | – | ✓ |
| Contrast | – | – | ✓ |
| Brightness | – | – | ✓ |
| Caliper | – | – | – |
| BLOB | – | – | – |
| **Identification** | | | |
| Barcode \| Barcode Advanced | ✓ \| – | ✓ \| ✓ | ✓ \| ✓ |
| Datacode \| Datacode Advanced | ✓ \| – | ✓ \| ✓ | ✓ \| ✓ |
| OCR | – | – | ✓ |
| **Robotic functions** | | | |
| Result offset Image \| 2D \| 3D | – \| – \| – | – \| – \| – | – \| – \| – |
| Gripping space check | – | – | – |
| **Result processing** | | | |
| Result processing: Text | – | ✓ | ✓ |
| Result processing: Math | – | – | – |

### VISOR® Robotic

| | VISOR® Robotic | |
|---|---|---|
| | **Advanced** | **Professional** |
| **Applications** | Robotics, localization, presence, completeness, measurement, position control | |
| **Resolution** | | |
| V10 (800 x 600): Mono \| Color | ✓ \| – | – |
| Frames per second: Mono \| Color | 75 | – |
| V20 (1440 x 1080): Mono \| Color | ✓ \| ✓ | |
| Frames per second: Mono \| Color | 40 \| 20 | |
| V50 (2560 x 1440): Mono \| Color | – | ✓ \| ✓ |
| Frames per second: Mono \| Color | – | 22 \| 8 |
| **Lighting** | White; for Mono only: Red, Infrared | |
| Multishot (Mono) | – | – |
| Target laser | ✓ | ✓ |
| **Objective** | | |
| V10 Wide \| Medium \| Narrow \| C-Mount | ✓ \| ✓ \| ✓ \| ✓ | – \| – \| – \| – |
| V20 Wide \| Medium \| Narrow \| C-Mount | ✓ \| ✓ \| ✓ \| ✓ | |
| V50 Wide \| Medium \| Narrow \| C-Mount | – \| – \| – \| – | – \| – \| – \| ✓ |
| **Interfaces** | Ethernet \| EtherNet/IP \| PROFINET | |
| Inputs \| Outputs \| Selectable | 2 \| 2 \| 6 | |
| Encoder input | ✓ | ✓ |
| Ethernet \| EtherNet/IP | ✓ \| ✓ | ✓ \| ✓ |
| PROFINET \| SensoWeb | ✓ \| ✓ | ✓ \| ✓ |
| Service port | ✓ | ✓ |
| **Job \| Detectors** | | |
| Number of jobs (max.) | 255 | |
| Number of detectors per job (max.) | 255 | |
| **Calibration** | | |
| Scaling (Measurement) | ✓ | ✓ |
| Calibration plate (Measurement) | ✓ | ✓ |
| Point pair list (Robotics) | ✓ | ✓ |

| | VISOR® Robotic | |
|---|---|---|
| | **Advanced** | **Professional** |
| Calibration plate (Robotics) | ✓ | ✓ |
| Hand-Eye calibration (Robotics) | – | ✓ |
| Base-Eye calibration (Robotics) | – | ✓ |
| **Pre-processing** | | |
| Pre-processing filter | ✓ | ✓ |
| Repeat mode | ✓ | ✓ |
| Shutter variation | ✓ | ✓ |
| Free-form search region | ✓ | ✓ |
| **Alignment** | | |
| Contour matching (translation, rotation 360°) | ✓ | ✓ |
| Pattern matching (translation, rotation 360°) | ✓ | ✓ |
| Edge detector (translation, rotation) | ✓ | ✓ |
| **Object detection** | | |
| Contour (translation, rotation 360°)<br>Multiple objects | ✓<br>✓ | ✓<br>✓ |
| Contour 3D<br>Multiple objects | –<br>– | ✓<br>✓ |
| Pattern matching (translation, rotation 360°)<br>Multiple objects | ✓<br>✓ | ✓<br>✓ |
| Gray | ✓ | ✓ |
| Contrast | ✓ | ✓ |
| Brightness | ✓ | ✓ |
| Caliper | ✓ | ✓ |
| BLOB | ✓ | ✓ |
| **Identification** | | |
| Barcode \| Barcode Advanced | – \| – | ✓ \| ✓ |
| Datacode \| Datacode Advanced | – \| – | ✓ \| ✓ |

| | VISOR® Robotic | |
|---|---|---|
| | **Advanced** | **Professional** |
| OCR | – | ✓ |
| **Robotic functions** | | |
| Result offset Image \| 2D \| 3D | ✓ \| ✓ \| ✓ | |
| Gripping space check | ✓ | ✓ |
| **Color detectors for Color variants** | | |
| Color Value | ✓ | ✓ |
| Color Area | ✓ | ✓ |
| Color List | ✓ | ✓ |
| Color distance \| Binarization | ✓ | ✓ |
| **Result processing** | | |
| Result processing: Text | – | ✓ |
| Result processing: Math | ✓ | ✓ |

**VISOR® Allround**

| | VISOR® Allround | |
|---|---|---|
| | **Advanced** | **Professional** |
| **Applications** | Presence, completeness, measurements, color Reading of barcodes, data codes, text, multishot, position control | |
| **Resolution** | | |
| V10 (800 x 600): Mono \| Color | ✓ \| ✓ | – \| – |
| Frames per second: Mono \| Color | 75 \| 50 | – \| – |
| V20 (1440 x 1080): Mono \| Color | ✓ \| ✓ | |
| Frames per second: Mono \| Color | 40 \| 20 | |
| V50 (2560 x 1440): Mono \| Color | – \| – | ✓ \| ✓ |
| Frames per second: Mono \| Color | – \| – | 22 \| 8 |
| **Lighting** | White; for Mono only: Red, Infrared | |
| Multishot (Mono) | ✓ | ✓ |
| Target laser | ✓ | ✓ |
| **Objective** | | |
| V10 Wide \| Medium \| Narrow \| C-Mount | ✓ \| ✓ \| ✓ \| ✓ | – \| – \| – \| – |
| V20 Wide \| Medium \| Narrow \| C-Mount | ✓ \| ✓ \| ✓ \| ✓ | |
| V50 Wide \| Medium \| Narrow \| C-Mount | – \| – \| – \| – | – \| – \| – \| ✓ |
| **Interfaces** | Ethernet \| EtherNet/IP \| PROFINET | |
| Inputs \| Outputs \| Selectable | 2 \| 2 \| 6 | |
| Encoder input | ✓ | ✓ |
| Ethernet \| EtherNet/IP | ✓ \| ✓ | ✓ \| ✓ |
| PROFINET \| SensoWeb | ✓ \| ✓ | ✓ \| ✓ |
| Service port | ✓ | ✓ |
| **Job \| Detectors** | | |
| Number of jobs (max.) | 255 | |
| Number of detectors per job (max.) | 255 | |
| **Calibration** | | |
| Scaling (Measurement) | ✓ | ✓ |
| Calibration plate (Measurement) | ✓ | ✓ |

| | VISOR® Allround | |
|---|---|---|
| | **Advanced** | **Professional** |
| Point pair list (Robotics) | – | ✓ |
| Calibration plate (Robotics) | – | ✓ |
| Hand-Eye calibration (Robotics) | – | ✓ |
| Base-Eye calibration (Robotics) | – | ✓ |
| **Pre-processing** | | |
| Pre-processing filter | ✓ | ✓ |
| Repeat mode | ✓ | ✓ |
| Shutter variation | ✓ | ✓ |
| Free-form search region | ✓ | ✓ |
| **Alignment** | | |
| Contour matching (translation, rotation 360°) | ✓ | ✓ |
| Pattern matching (translation, rotation 360°) | ✓ | ✓ |
| Edge detector (translation, rotation) | ✓ | ✓ |
| **Object detection** | | |
| Contour (translation, rotation 360°)<br>Multiple objects | ✓<br>✓ | ✓<br>✓ |
| Contour 3D<br>Multiple objects | –<br>– | ✓<br>✓ |
| Pattern matching (translation, rotation 360°)<br>Multiple objects | ✓<br>✓ | ✓<br>✓ |
| Gray | ✓ | ✓ |
| Contrast | ✓ | ✓ |
| Brightness | ✓ | ✓ |
| Caliper | ✓ | ✓ |
| BLOB | ✓ | ✓ |
| **Identification** | | |
| Barcode \| Barcode Advanced | ✓ \| ✓ | ✓ \| ✓ |

| | VISOR® Allround | |
|---|---|---|
| | **Advanced** | **Professional** |
| Datacode \| Datacode Advanced | ✓ \| ✓ | ✓ \| ✓ |
| OCR | ✓ | ✓ |
| **Robotic functions** | | |
| Result offset Image \| 2D \| 3D | – \| – \| – | ✓ \| ✓ \| ✓ |
| Gripping space check | – | ✓ |
| **Color detectors for Color variants** | | |
| Color Value | ✓ | ✓ |
| Color Area | ✓ | ✓ |
| Color List | ✓ | ✓ |
| Color distance \| Binarization | ✓ | ✓ |
| **Result processing** | | |
| Result processing: Text | ✓ | ✓ |
| Result processing: Math | ✓ | ✓ |

# 6 Installation

## 6.1 Mechanical installation

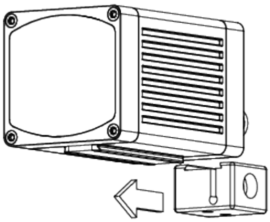
In order to optimize the evaluations, the vision sensor must be protected from vibrations. Secure the supply and I/O cables with cable binders to prevent slipping or crushing. The vision sensor’s position must ensure that disrupting effects, such as permissible deviations in the position of the object being measured and variations in the surrounding lighting, will not have a significant impact.

## 6.1.1 Installing the bracket

For installation, use the Mounting bracket MK 45 (543-11000) or the MG 3A mounting joint (543-11024) only.

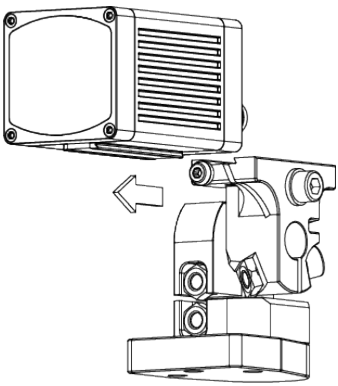
**NOTE:**
The MG 3A mounting joint is not included.
It is available under part number 543-11024 from SensoPart Industriesensorik GmbH.

#### Installing VISOR® on Mounting bracket MK 45



1. Slide the mounting bracket onto the sensor's dovetail guide.
2. Use the Allen key to tighten the socket cap screw in the mounting bracket's cross-hole.
3. Now install the mounting bracket on a suitable fixture.

#### Installing VISOR® on MG 3A mounting joint

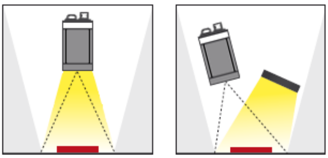


1. Slide the mounting joint's dovetail onto the sensor's dovetail guide.
2. Use the Allen key to tighten the socket cap screw in the mounting joint's cross-hole.
3. Now install the mounting joint on a suitable fixture.

## 6.1.2 Sensor and illumination configuration

The terms "bright field illumination," "dark field illumination," and "diffuse illumination" are used to distinguish between three sensor and illumination configurations.

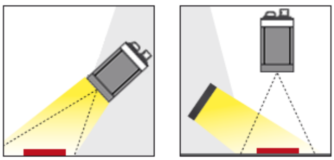
### Arrangement for bright-field illumination



When using bright field illumination, the positions of the sensor, object, and illumination are chosen in such a way that the light will be reflected directly from the object surface towards the sensor.

Smooth object surfaces will appear bright, while bumps and depressions will appear dark.

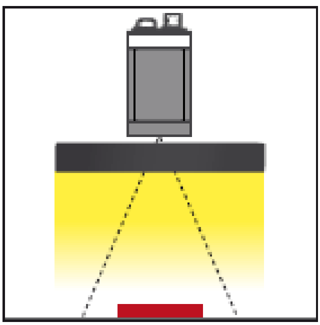
### Arrangement for dark-field illumination



When using dark field illumination, the positions of the sensor, object, and illumination are chosen in such a way that the light will not be reflected directly from the object surface towards the sensor.

Smooth object surfaces will appear dark, while bumps and depressions will appear bright.

### Diffuse illumination configuration



Diffuse illumination is only possible with an external source of lighting.
Diffuse illumination is used everywhere where highly reflective, curved, or, above all, irregularly shaped object surfaces are concerned (e.g. aluminum foil on blister packs, etc.). Such objects can only be illuminated with diffuse illumination (i.e. uniform illumination coming from all directions), and not spot-shaped illumination. Diffuse illumination is also known as "cloudy day" illumination, i.e. uniform light from behind the cover of clouds as a light source rather than from direct sunlight.

**NOTE:**
External sources of light are available from SensoPart Industriesensorik GmbH.
Additional information: Accessories (Page 391)

### Fine adjustment

It will not be possible to carry out a fine adjustment on the vision sensor until after electrical installation and the initial setup (VISOR® software installation).

## 6.1.3 Blocking ambient light

### Physical enclosure

Ambient light from windows or skylights that disrupts the scene only temporarily on certain days / seasons of the year can often be blocked with mechanical elements.

### Version with infrared illumination

Another option for becoming more independent of ambient light is to use the corresponding VISOR® variant with IR illumination. Here, the test scene is illuminated with built-in, powerful IR illumination. The receiver is equipped with appropriate filters that only let light within this spectrum get through to the receiver, i.e., the sensor operates within a narrow wavelength range and, to the greatest extent possible, only with the light it emits itself.

Another advantage of using infrared illumination consists of the fact that the flashes are invisible and accordingly will not inconvenience any people working in the area.

## 6.1.4 Alignment for vertical illumination

In order to ensure that the VISOR® is perfectly perpendicularly aligned with the object surface, place a piece of reflective foil or a mirror on the object and start the VISOR® operating software as a test. For an image that is continually updated, select the trigger mode "Free run" and under Trigger / Image update "Continuous". Now align the sensor as perpendicular as possible to the reflective / mirror surface until the integrated illumination LEDs dazzle directly into the image of the user interface. Arrangement see Figure in Chapter Sensor and illumination configuration

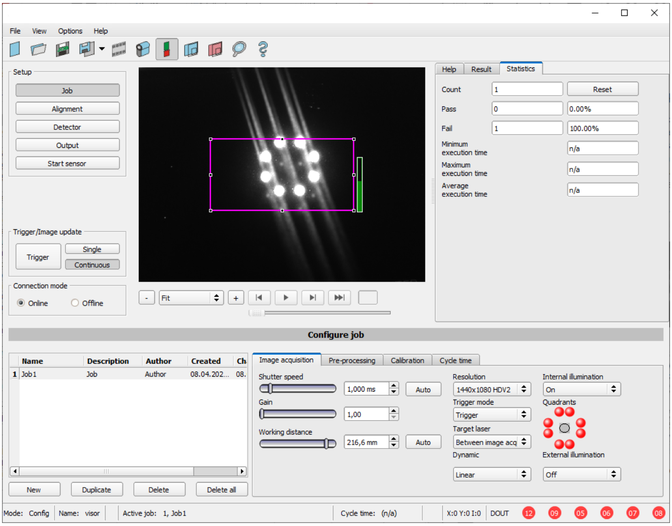
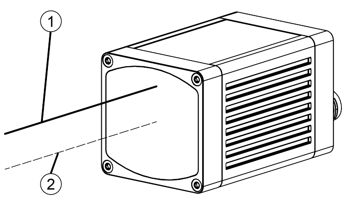



*Fig. 2: Figure for vertical illumination*

## 6.1.5 Target laser

The laser can be configured in the VISOR® software in the SensoConfig module under Job / Image acquisition / Target laser.




The target laser's laser beam runs parallel to the optical axis at a distance of 12.5 mm and can be used to align the sensor.

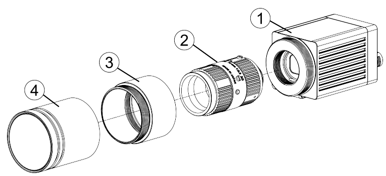
① Laser beam

② Optical axis

## 6.1.6 C-Mount objective and protective casing

Various objectives with different focal lengths are available for C-Mount VISOR® vision sensors. The objectives can be protected with a protective casing. Objectives and protective casings are available from SensoPart.

**Installing the C-Mount objective and protective casing on the vision sensor**



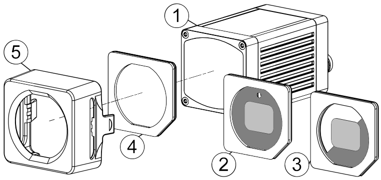

1. Screw the objective into the C-Mount vision sensor's internal thread.
2. Screw the protective casing's extension onto the VISOR® vision sensor's external thread.
3. Screw the protective casing onto the extension's external thread.

① C-Mount VISOR® vision sensor
② Objective
③ Extension for the protective casing
④ Protective casing

## 6.1.7 Polarizing filters and spark protection guard

**ATTENTION:**
If you use the spark protection guard, the maximum operating temperature is lowered to 45 °C.

Various polarizing filters and a spark protection guard are available for the VISOR® vision sensor.




① **VISOR® vision sensor**

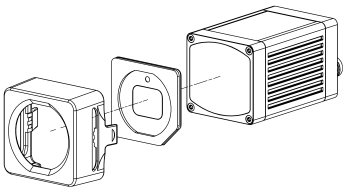
② **Polarizing filter for spark protection with 100% coverage.**

③ **Polarizing filter for spark protection with 50% coverage.**
Individual quadrants of the internal illumination can be turned on and off in the VISOR® software. If the only quadrants that are turned on are the ones within the 50% polarizing filter cover, polarized light will be emitted. If the quadrants that are turned on are the ones in the clear half of the polarizing filter, unpolarized light will be emitted instead.

④ **Spark protection guard**

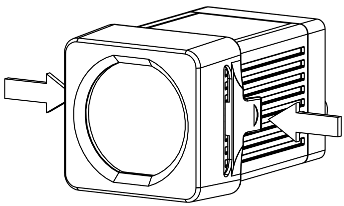
⑤ **Spark protection**

### Installing the polarizing filter and spark protection guard



1. Insert the polarizing filter, aligned at the chamfer with the foam side facing towards the front of the sensor, into the spark protection guard.
2. Slide the polarizing filter disk and spark protection guard onto the front of the sensor until the guard locks into place on the sensor case and you hear two clicks.

### Removing the polarizing filter and spark protection guard



1. Squeeze the two tabs on the side of the spark protection guard at the same time. The spark protection guard will come loose from the sensor case and you will be able to remove it.

## 6.2 Electrical installation

**WARNING**
The connection should be made exclusively by trained qualified personnel.
All live components must be de-energized when performing the electrical installation work.

**ATTENTION:**
When using the unit on a network, it is necessary to ensure that the vision sensor's default network address (IP address), 192.168.100.100 / 24, is free and that it is not being used by any other device connected to the network. If necessary, the vision sensor's IP address must be changed. For more information, please refer to "Network settings."
For error-free operation, the length of the connecting cables must not exceed 30 m. Failure to do this may cause malfunction.

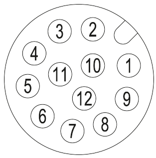
For stand-alone operation (independent of PC / PLC), only connection 24 V DC is required after startup.

### 6.2.1 24 V DC connection

M12 connection socket for voltage supply and digital I/O.

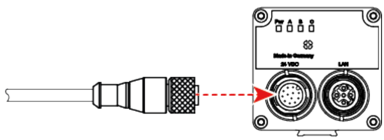
**ATTENTION:**
Use shielded cables exclusively, and terminate the shield across a large area.
Tightening torque for connector: 0.6 – 1 Nm.

### I/O Mapping

| | PIN | Color [3] | Signal |
|---|---|---|---|
|  | 1 | BN | + $V_B$ (24 VDC) |
| | 2 | BU | GND |
| | 3 | WH | IN (external trigger) |
| | 4 | GN | READY (ready for next external trigger) |
| | 5 [1] | PK | IN/OUT (encoder B+) |
| | 6 | YE | IN/OUT, (external illumination south)[4] |
| | 7 | BK | IN/OUT, (external illumination west)[4], LED B [2] |
| | 8 | GY | IN/OUT, (external illumination north)[4], LED C [2] |
| | 9 | RD | IN/OUT, (external illumination, external illumination east)[4] |
| | 10 [1] | VT | IN (encoder A+) |
| | 11 | GY/PK | VALID (indicator for valid results) |
| | 12 | RD/BU | IN/OUT (ejector), LED A [2] |

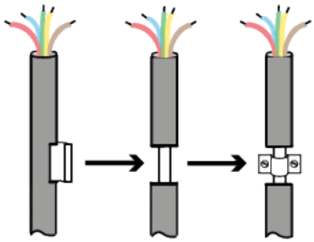
[1] Not available on all standard models
[2] All LEDs are driven without taking into account any delays used
[3] Colors match the SensoPart power cables. If other cables are used, there might be differences.
[4] Only if Multishot is active

### Connecting the power cable



1. Connect the power cable to the 24 VDC connector on the vision sensor.
2. Fasten the connector and tighten it with a torque of 0.6 to 1 Nm.

### Terminating the shield



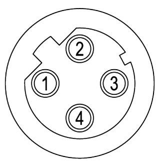
1. Strip a section of the power cable (remove a section of the jacket). The power cable's shield will be exposed.
2. Place a shielding clamp or a similar component over the stripped cable section and secure this shielding clamp to a shielding plate.

## 6.2.2 LAN connection

M12 connector for Ethernet connection.

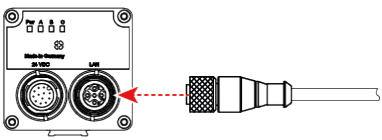
**ATTENTION:**
Use only the correct network cables.
Tightening torque for connector: 0.6 – 1 Nm.

### I/O Mapping

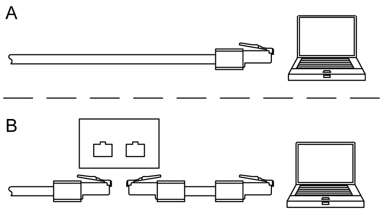
<table>
<tr><th></th><th>PIN</th><th>Signal</th></tr>
<tr><td rowspan="4"></td><td>1</td><td>TxD+</td></tr>
<tr><td>2</td><td>RxD+</td></tr>
<tr><td>3</td><td>TxD-</td></tr>
<tr><td>4</td><td>RxD-</td></tr>
</table>

### Connecting the Ethernet cable

The vision sensor can be connected either directly to a PC (the preferred option) or to a PC through a network.



1. Connect the Ethernet cable to the vision sensor's LAN connector.
2. Fasten the connector and tighten it with a torque of 0.6 to 1 Nm.




3. Use the RJ45 connector to connect the Ethernet cable either
   A) Directly to your PC (the preferred option).
   B) To your PC through a network.

### 6.2.3 Exemplary connection plan

Exemplary connection plan for the following configuration:

- Power supply
- Trigger
- 1x digital switch output
- Encoder
- Ethernet to PC or PLC

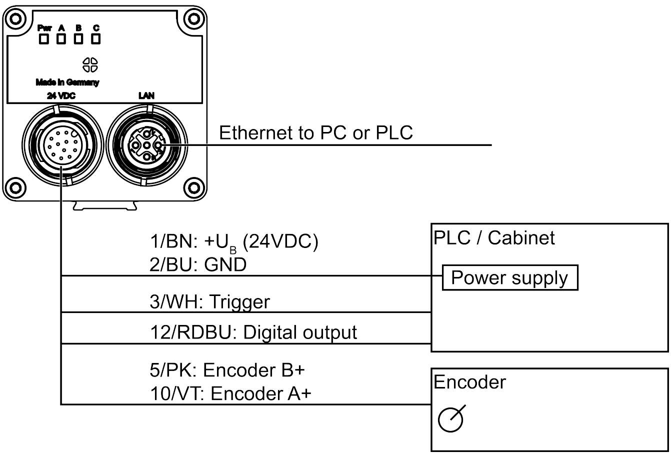



*Fig. 3: Exemplary connection plan*

## 6.2.4 Electrical connection Supply voltage with shield

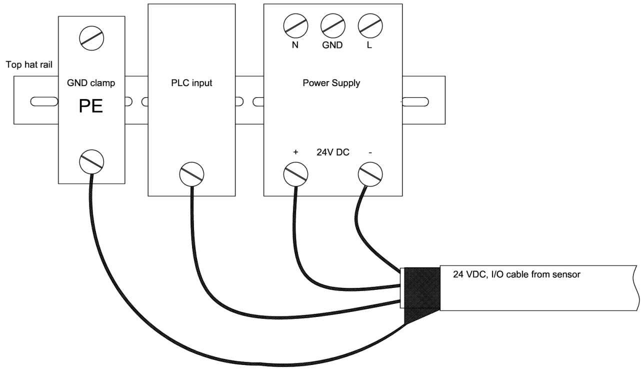



*Fig. 4: Power supply 24 V DC in the control cabinet with shield*

## 6.2.5 Electrical connection PNP / NPN

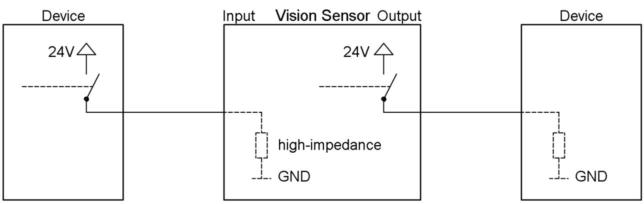



*Fig. 5: Connection example for VISOR® in PNP mode. Inputs/outputs switch to +24 V*

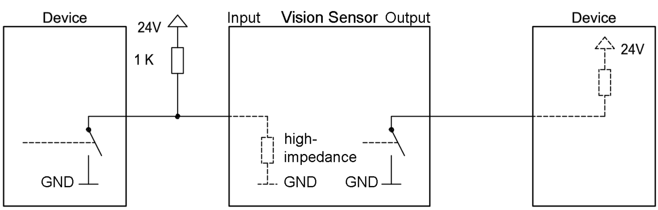



*Fig. 6: Connection example for VISOR® in NPN mode*

The inputs are ground-referenced. Accordingly, an additional pull-up resistor may be needed in certain cases to ensure that the input voltage will be increased to 24 V in an unswitched state. The outputs switch to ground.

## 6.3 Network settings Short guide

The following instructions explain how to modify the network configuration for the PC and the VISOR® vision sensor. If incorrect settings are used, the network connections in the computer may be lost. To be on the safe side, note the former settings and reuse them if required. Following this procedure, it may be necessary to restart the system. In order to determine which IP address is allowed in your network or locally on your PC, and to carry out the necessary settings on your PC, please contact the responsible system administrator or administrator beforehand. The illustrations, dialogues and menus used are taken from Microsoft Windows 10 operating system. The illustrations are similar in other operating systems.

### 6.3.1 Basic PC and VISOR® vision sensor settings

Prerequisite for configuring the VISOR® vision sensor with a PC: PC with network adapter and an installed TCP/IP LAN connection (even if the PC is not connected to a network). The VISOR® supports the automatic detection of the Ethernet transmission rate, but a maximum of 100 MBit. The internet protocol IPv4 must be activated. There are two ways to configure the VISOR® vision sensor.

See also chapter Network connection

1. Direct connection – setting the IP address of the PC

2. Network Connection – setting the IP Address of the VISOR® vision sensor

### 6.3.2 Direct connection - Setting the IP address of the PC

To connect the VISOR® vision sensor to a computer via Ethernet, the IP address settings of the two devices must correspond to each other. The default setting for the VISOR® vision sensor's IP address is 192.168.100.100 / 24 with a subnet mask of 255.255.255.0. For direct connection, the PC must be set to a fixed IP address suitable for the sensor, as follows:

1. Clicking on Start / Control Panel / Network Connection / LAN Connection / Properties opens the dialog window "Local Area Connection Properties".
2. In the list "This connection requires the following elements", select the option "Internet Protocol (TCP/IP)" and click the button "Properties".
3. In the following window, set the desired IP address and subnet mask of the PC.
4. Confirm entries with OK.

Flowchart: Network connection: Direct connection (Page 370)

Example:

The VISOR® vision sensor comes with its IP address set to 192.168.100.100 and its subnet mask set to 255.255.255.0. In this case, the IP address may be set to any value between 192.168.100.1 and 192.168.100.254, with a subnet mask 255.255.255.0, with the exception of the sensor IP address (192.168.100.100).
To alter the sensor's IP address, see Sensor network settings (Page 76). Do not use the network addresses .0 and .255 as device addresses, as these are mostly reserved for network infrastructure, such as servers, gateways, etc.

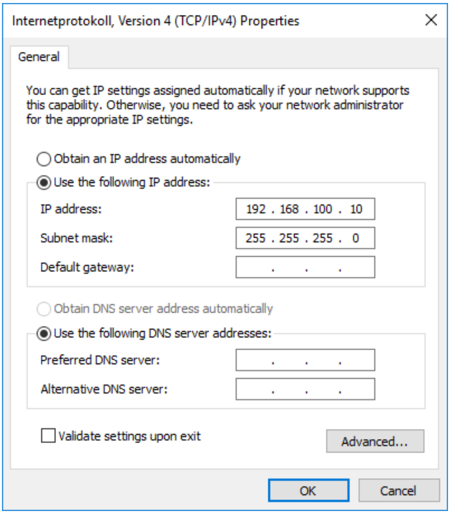



*Fig. 7: PC IP Setup*

## 6.3.3 Network connection - Setting the IP address of the VISOR® vision sensor

Before connecting the sensor to the network, check with the network administrator whether the sensor's address has already been assigned (default: 192.168.100.100 with subnet mask 255.255.255.0). This can otherwise cause network failure. The configured IP address should be written down on the enclosed label for the VISOR® vision sensor. The label must then be stuck on the sensor in a clearly visible place after installation.

**Network connection speed:**

Especially when using the VGA resolution and SensoView, the sensor should be operated with 100 Mbit /full-duplex only.

**Sensor IP still free:**

Connect the sensor to the network and then set the sensor's IP to match the administrator's specifications, as follows, beginning with Point 2.

**Sensor IP already assigned:**

1. First connect sensor and PC directly and set an authorized IP address in the sensor.
2. Connection via the network can then be carried out. First ensure electrical connection and installation of PC software has been completed. To set the IP address on the VISOR® vision sensor, carry out the following steps in the PC software:
   a. Start SensoFind.
   b. Select the VISOR® vision sensor you want from the list of active sensors.
   c. Set sensor's new IP address with the "Set" button. The IP address is assigned by your system administrator. The PC's IP address is shown in the status bar under the buttons. Note:: Certain PCs may have more than one Ethernet connection, i.e., wireless and wired LAN connections.
   d. Select the sensor and connect via SensoConfig or SensoView.

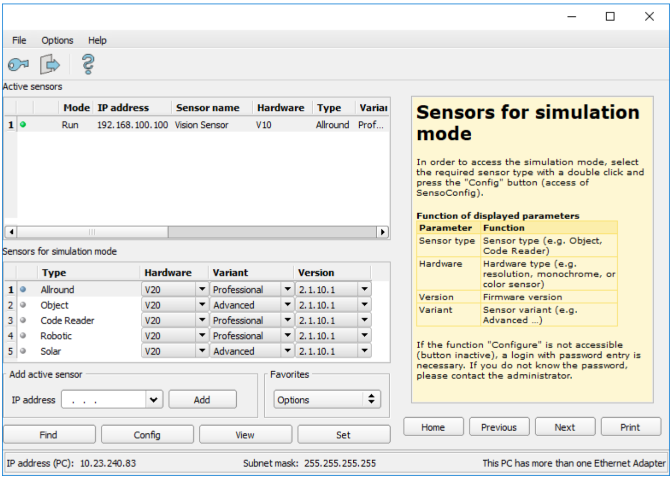



*Fig. 8: SensoFind*

Modification of the standard gateway enables operation in different sub-networks. Only alter this setting after consultation with your administrator. Automatic integration of a new computer or sensor into the existing network without manual configuration is possible through DHCP. Normally, only the automatic reference of the IP address must be set at the sensor on the client. When the sensor is started on the network, it can obtain the IP address, net mask, and gateway from a DHCP server. Activation of the DHCP mode is carried out via the "Set" button by activating the checkbox "DHCP". Since this means that the exact same VISOR® may have different IP addresses at different times, a sensor name must be assigned when enabling DHCP. If there are multiple VISOR® sensors on a network, each one must be assigned its own unique name.

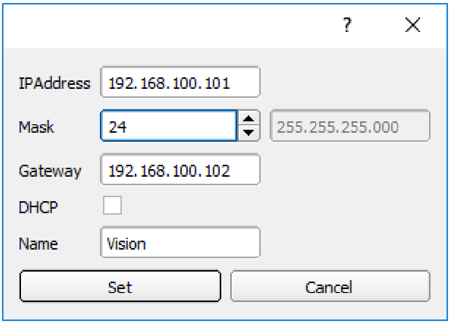



*Fig. 9: VISOR® IP setup*

If a VISOR® with DHCP is turned on on a network without a DHCP server, the VISOR® will automatically set its IP address to 0.0.0.0. This can occur, for example, in the event of a power or server failure or in the event that the system is restarted. The reason for this is that the DHCP server may boot more slowly than the VISOR®. Make sure that the VISOR® is turned on only after the DHCP server is available.

Flowchart: Network connection: Connection via network (Page 371)

# 7 VISOR® software – Overview and Quick Start Guide

The following describes the basic structure of the VISOR® software. The structure of the individual modules (SensoFind, SensoConfig and SensoView) is explained and short instructions are given for each module. This Quick Start Guide uses the example of an object sensor to explain the exemplary procedure for setting an inspection task on the VISOR® vision sensor.

## 7.1 Structure of the VISOR® software

The VISOR® software software is made up of the following three modules:

- **SensoFind**
  With this module, you can select the sensor to be configured or a sensor simulation, and start the applications SensoConfig or SensoView. Here, system settings such as IP addresses or firmware updates can be modified, and passwords and user rights can be managed.
- **SensoConfig**
  This module contains comprehensive functions for setting up sensors and configuring inspection tasks (jobs). If password protection is activated, you need the authorization of the user group administrator for the configuration.
- **SensoView**
  This module displays images and results. You can use it to monitor / check sensors and analyze measurement results. In addition, extensive archiving functions are available. Compared to SensoConfig, it only offers limited configuration options. If password protection is activated, the authorization of the user group administrator or worker is needed.

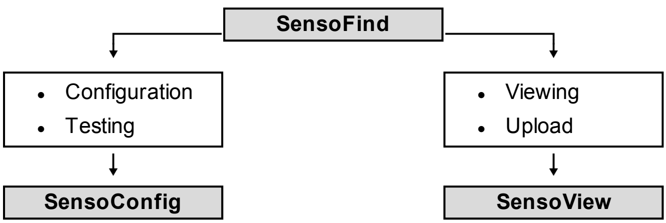



*Fig. 10: Software structure*

You can download free, up-to-date versions of the VISOR® software at www.sensopart.com.

## 7.2 Start the VISOR® software

Click on the "VISOR Vision Sensor" desktop icon to start the VISOR® software.


*Fig. 11: VISOR® software Icon*

## 7.3 SensoFind

### 7.3.1 SensoFind - Overview

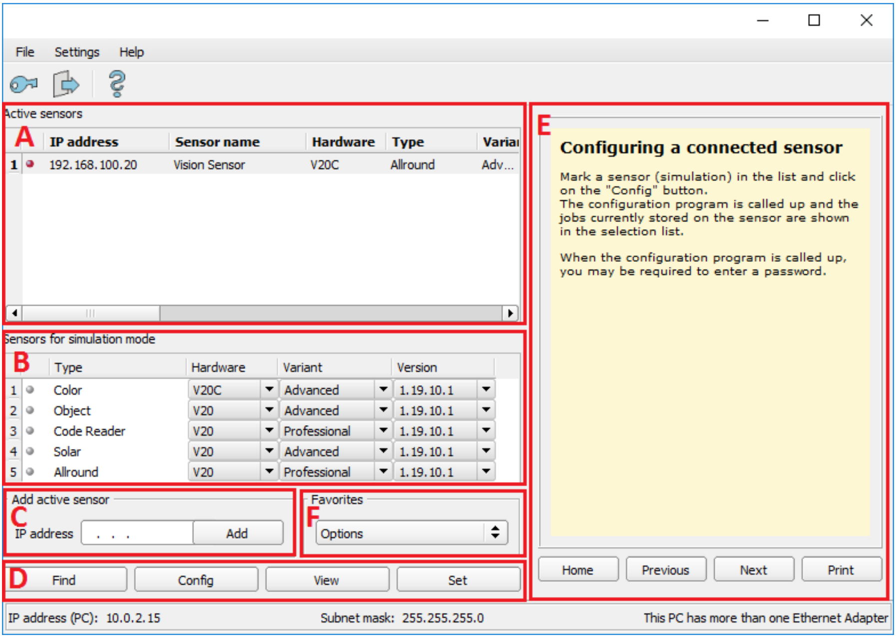



*Fig. 12: SensoFind Overview*

**A: Active sensors**

This list shows all the VISOR® vision sensors that can be reached from the PC.

**B: Sensors for simulation mode**

Shows all the VISOR® vision sensors available for the offline simulation.

**C: Add sensors via IP address**

Sensors that do not appear in the list "Active sensors" after the software has started or after

"Find" (triggering a further search run), but are definitely in the network (possibly behind a gateway, for example) and whose IP address is known, can be entered here with their IP address. By clicking the "Add" button, such sensors, if found, are also entered in the "Active sensors" list and can now be edited.

**D: Functions**

- **Find**
  Triggers another search
- **Configure**
  Configuration of a connected sensor or sensor simulation = SensoConfig
- **View**
  Displays image or result data from a connected sensor = SensoView
- **Settings**
  Edits network settings such as the sensor's IP address, etc.

**E: Context help**
Context-sensitive help for the current topic

**F: Favorites**
VISOR® vision sensors can be saved as favorites. These favorites can be used to quickly access and manage VISOR® vision sensors.

Additional information: VISOR® Software – SensoFind

## 7.3.2 SensoFind – Quick Start Guide

In this program, you can select a sensor or a sensor simulation for configuration or display (monitoring) and carry out different basic settings.

### 7.3.2.1 Open sensors or sensor simulations

#### 7.3.2.1.1 Configuring or displaying sensors

In order to open a sensor for configuration or display, select with a single left mouse click the required sensor in the "Active sensors" list. Then, with a click on the **"**Config**"** button, the "SensoConfig" module starts. With the button "View", the "SensoView" module starts.

#### 7.3.2.1.2 Sensor simulation

To open a sensor for offline simulation, highlight the desired sensor in the "Sensors for simulation mode" list. Then, with a click on the "Config" button, the "SensoConfig" module starts.

### 7.3.2.2 Passwords

#### 7.3.2.2.1 Set up passwords

At the first start after the installation, the password input is completely deactivated and the auto-login as administrator is preset. If parameter settings should be protected against unauthorized access, passwords for the password levels "Admin" and "User" should be assigned. This can be accessed via the menu bar File / User administration or via the button with the key symbol in the toolbar.

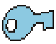


*Fig. 13: Password button*

#### 7.3.2.2.2 Password levels

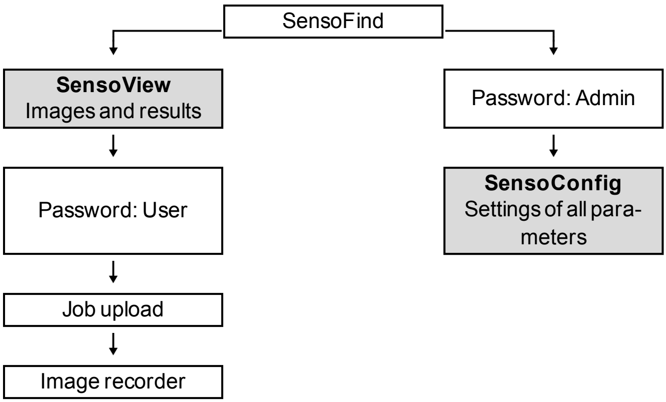



*Fig. 14: Password levels*

| Password level | SensoFind | SensoConfig | SensoView |
|---|---|---|---|
| Administrator password | All functions | All functions | All functions |
| Worker password | All functions except<br>• Configurating<br>• Settings<br>• Update | none | All functions, including Job Upload and Image Recorder |

| Password level | SensoFind | SensoConfig | SensoView |
| --- | --- | --- | --- |
| User<br>(without password) | All functions except<br>• Configurating<br>• Settings<br>• Update | none | Only display of images, inspection results, and statistics |

In order to be able to use the "Config" function after assigning passwords, a login is now necessary: To login, click on the Login button in the toolbar and enter the previously assigned password.

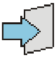


*Fig. 15: Login button*

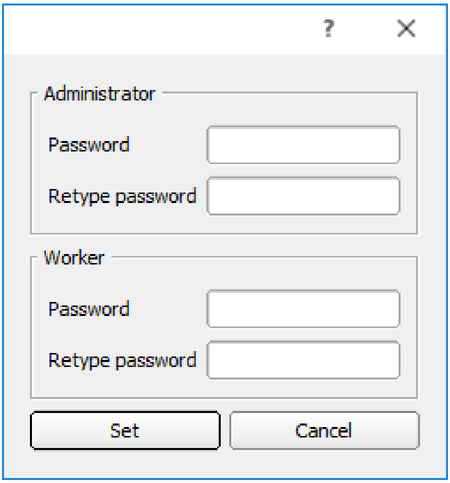



*Fig. 16: Password input*

By assigning an empty password, the query can be acknowledged again without further input. By activating the checkbox "Deactivate password query" the query will be deactivated permanently.

If passwords have been assigned and then forgotten, the software can be reset to the delivery status by reinstalling the software.

## 7.4 SensoConfig

### 7.4.1 SensoConfig - Overview

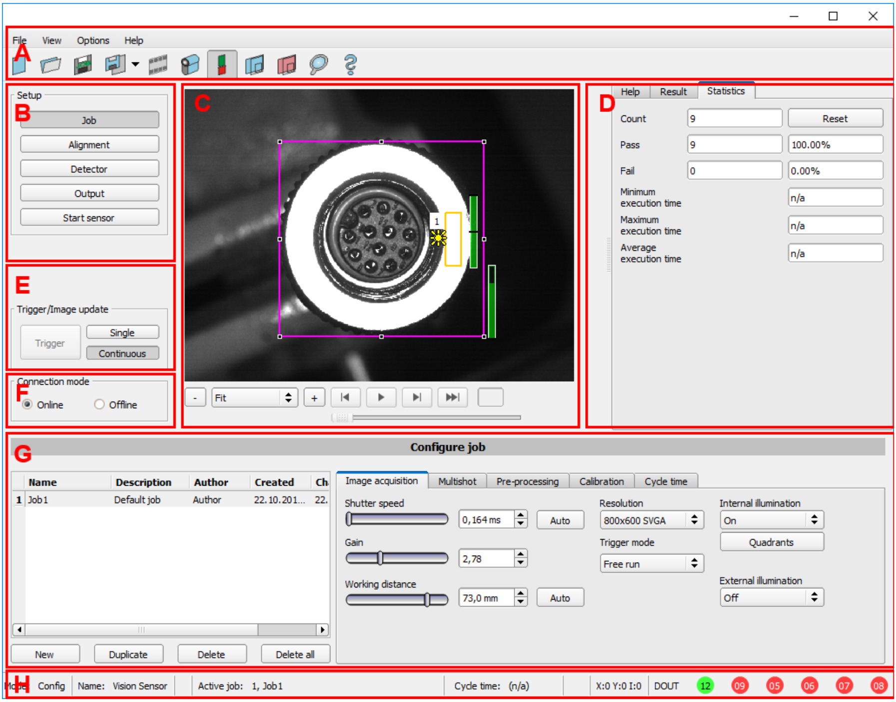



*Fig. 17: SensoConfig Overview*

**The different areas are:**

**A: Menu and toolbar**

**B: Setup steps**: VISOR® Software – SensoConfig (Page 83)

**C: Image**

Image output with graphically adjustable working and search ranges as well as zoom function also filmstrip navigation

**D: Help, Result, Statistics**

- Help: Context-sensitive help for the current topic
- Result: Detector results for selected parameters

- Statistic: Display statistic on evaluation and execution time

**E: Image acquisition settings**
Switchover between continuous mode and single-frame mode and software trigger

**F: Connection mode**
Switchover between online and offline mode (sensor present or simulation without sensor)

**G: Configuration window**
Variable, content changes to the corresponding action to set the associated parameters.

**H: Status bar**
Various status information, including: Mode / name of VISOR® / active job. In the Run mode: Cycle time, xy position of the cursor and pixel intensity / individual I/O on/off status (as configured below in "Output / Digital output").

Additional information: VISOR® Software – SensoConfig

### 7.4.2 SensoConfig – Quick Start Guide

You can use this program to configure your VISOR® for one or more jobs in five setup steps.

#### 7.4.2.1 Configuring a job

To configure a job: Edit the job entry under Setup / Job or create a new job.

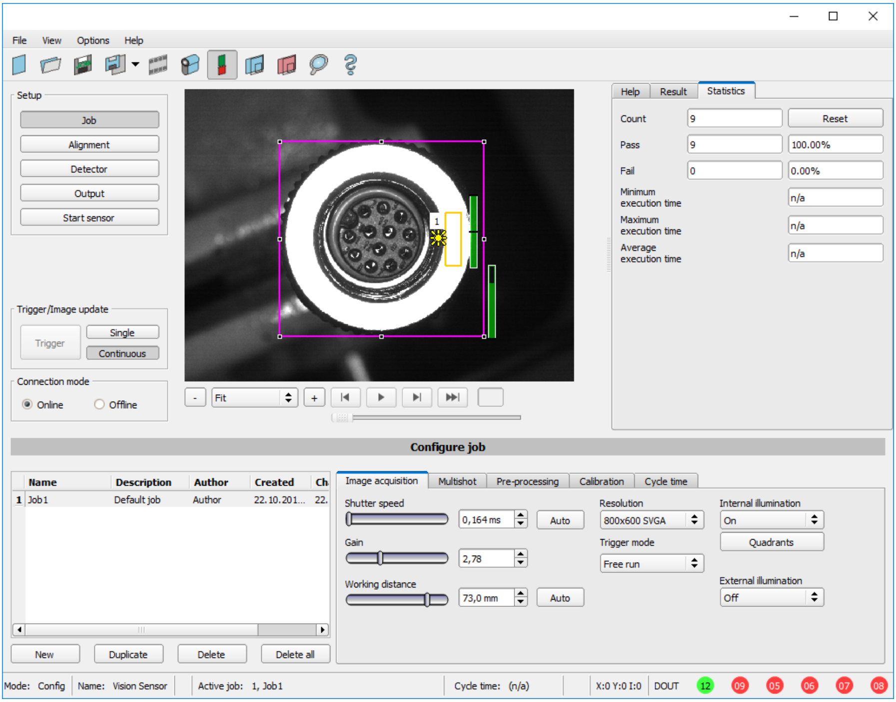



*Fig. 18: SensoConfig Job*

Here, new jobs are created and the jobs are managed. In addition, this is where all global settings that are valid for the entire inspection task, such as shutter speed, gain, illumination settings, etc. are carried out.

A job contains all the settings and parameters required to perform a specific inspection task.

- The following basic image settings should first be made to ensure a high-contrast and sharp image:
  - Image brightness: Adjust Shutter speed or Gain (see setup step Job/ Image acquisition tab)

  - Sharpness: Adjust the sharpness of the image by adjusting the "Working distance" control until a sharp image is visible (see chapter Job / Image acquisition tab).
- In the delivery state, the settings are Trigger mode = "Free run" (see Job / Image acquisition) and "Trigger / Image update" = "Continuous". Thus, a new image is permanently fed into the focus and brightness setting, thus permanently updating the display.
- The subsequent adjustment of the Alignment and the detectors is preferably carried out in single image mode, since all settings are then based on a master image and the image input is not permanently executed. For this purpose set the Trigger mode = Trigger in the Image acquisition tab.
- Within a job, Alignment and (depending on the sensor type) 32 or 255 different detectors to solve the inspection task can be defined below.

It is possible to save a job as a template. To do this, right-click on the job in the job list and select "Save as template". For each new job, the settings and detectors are then copied from the job template. In the job list, the job template is identified with a "T" (Template). The job template cannot be edited. To remove the job template, right-click on the template and select "Remove".

#### 7.4.2.2 Configuring Alignment

For objects or features whose position varies in the image, Alignment may be useful or necessary.

Alignment is optional and is available with the methods Pattern matching, Edge detector and Contour matching.

First select the appropriate Alignment method. Then graphically set the position and size of the corresponding work areas on the screen to the characteristic that is to be used to determine the position. The associated parameters are displayed on the bottom right-hand side and can also be adjusted there.

Alignment affects all detectors defined below in this job. In the example here, the upper left corner of the rectangular component, which only varies in position translatorically in X and Y direction, is used to determine the position of the component. Therefore the left and the upper outer edge and their intersection point are determined. If the angular position of the component can also vary, the "Contour matching" method should be used for Alignment.

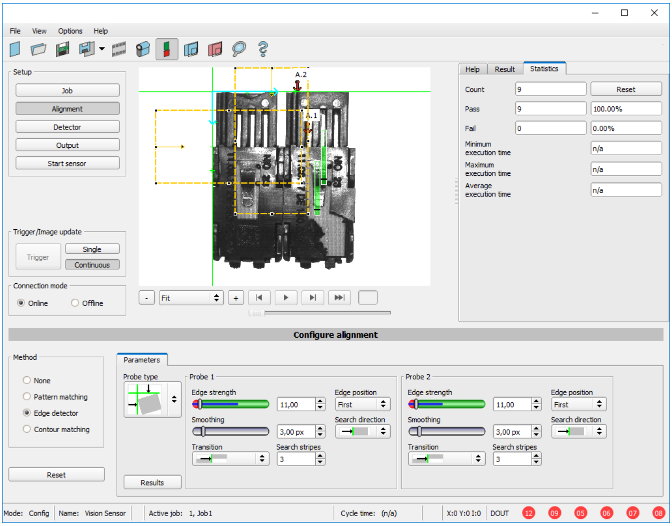



Fig. 19: SensoConfig, Alignment Edge detector

### 7.4.2.3 Configuring detectors

In the setup step Detectors, detectors can be selected and set to solve an inspection task.

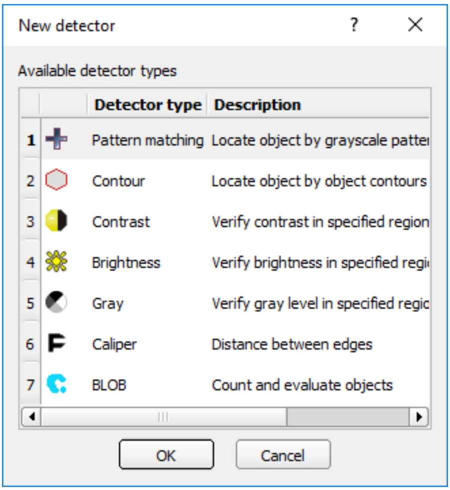



Fig. 20: Detector selection list, example: Object sensor

First select a suitable detector from the dialog shown above. Then set the work and search areas graphically in the image. If there are teach-in areas, they are taught-in immediately upon completing the setting. At the bottom left, all the detectors defined in this job are displayed in the detector list. At the bottom right, the parameters of the currently selected detector are displayed and can be adjusted there.

If additional features should be tested on the same part, you can use "New" to create any number of additional detectors, analogous to those described above. In the example, two brightness detectors were defined to check the presence of contacts in the test piece.

- Detector 1 finds a contact (brightness value is within the required range, as there is a metallically shiny, i.e. highly reflective contact) and therefore reports a positive result.
- Detector 2 finds no contact (brightness value is outside the required range, as there is hardly any reflection from dark plastic housing) and therefore reports a negative result.

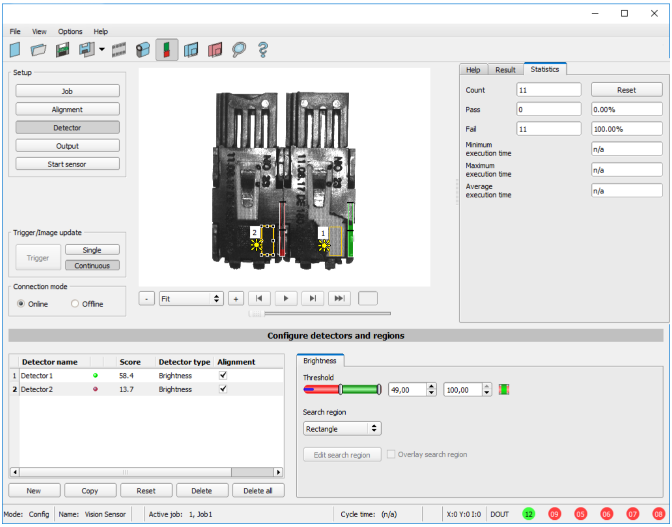



*Fig. 21: Set detector*

### 7.4.2.4 Output, I/O and data output

The setup step Output enables different settings of digital inputs / outputs and data output.

The interfaces can be selected and activated in the various tabs. Detector results can be logically linked and assigned to the existing I/Os.

The desired interface is also selected for the output of result data and the data string is compiled.

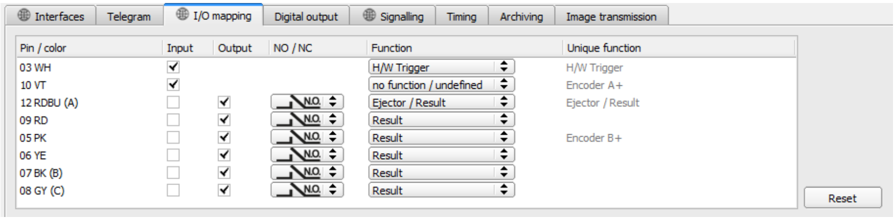



*Fig. 22: Output, Digital signals, and Data*

**Setting possibilities in the different tabs:**

- **Interfaces**
  Selection, Setting, and Activation of the individual interfaces.
- **Telegram**
  Used to configure the data output string via Ethernet or PROFINET.
- **I/O mapping**
  Used to select and map digital switching inputs and outputs.
- **Digital output**
  Assignment of a logical link using the Boolean results of all detectors.
  Definition of complex logical links via table or via input of a logical formula.
  A separate logical link can be assigned to each existing digital output.
- **Signalling**
  Settings for statistics and for digital outputs.
- **Timing**
  Used to configure delays: Trigger delay, result delay, result duration
- **Archiving**
  Used to configure data archiving.
- **Image transmission**
  Used to configure image transmission via image recorder or RAM drive.

Selection of: Binary or ASCII protocol, start / trailer, standard content / flexibly configurable, special individual data of the individual detectors.
Any number of individual results of all defined detectors can be freely arranged in the output string.

**NOTE:**
The settings in the "Interfaces", "I/O mapping" and "Signalling" tabs (indicated by the "globe" symbol) apply to the entire job set. Changes made in one job are applied to all other jobs.

### 7.4.2.5 Starting the sensor

When this function is activated, all settings are transferred to the sensor, stored in the flash, and, depending on the settings, made e.g. in the free run or triggered mode. All displays in the detector list, in the result field or under "Statistic", are updated here. With a click on "Start sensor" the transferred parameters are permanently stored and the corresponding hardware outputs are also set during execution.

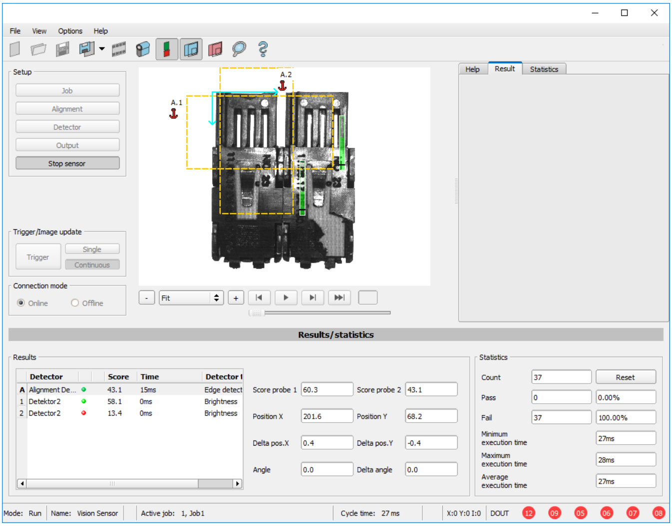


*Fig. 23: Start sensor*

## 7.5 SensoView

### 7.5.1 SensoView - Overview

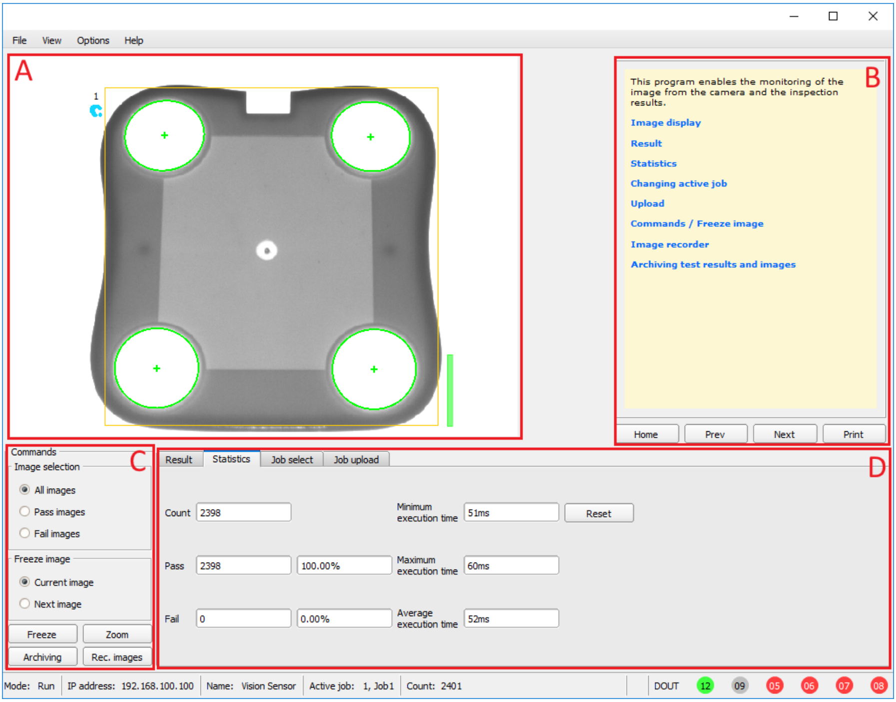



*Fig. 24: SensoView Overview*

**A: Image display**

**B: Context help**
Context-sensitive help for the current topic

**C: Commands**
Commands for displaying, transferring, and archiving images.

**D: Job and result display**
These tabs can display (statistical) results, switch jobs, and load jobs / job sets from SensoView to the sensor.

Additional information: VISOR® Software – SensoView

## 7.5.2 SensoView – Quick Start Guide

### Display images and results

This program is used to monitor / check connected sensors, to analyze inspection results, as well as to archive inspection results and images.

Nach Klick auf den Button „**View**" in SensoFind startet das Modul SensoView.

The current image is displayed with overlays of the Alignment and detectors (if "Image Transfer = Active" is enabled in the configuration module under Job / General).

- Im Reiter „**Result**“ werden die einzelnen Detektoren mit Ihren Ergebnissen, und das Gesamtergebnis dargestellt.
- Im Reiter „**Statistic**“ werden weitere statistische Ergebnisse angezeigt.
- With "**Freeze image**", event-driven (e.g. bad part) images can be captured in the display.
- With "**Zoom**", displayed images can be enlarged.
- With "**Archive images**", images and result data can be archived on the hard disk of the connected PC, as previously set under File / "Configure archiving", with or without numerical result data.
- Mit „**Rek. Bilder**" kann der Bildrekorder ausgelesen werden.
- Im Reiter „**Job**“ können auf dem Sensor vorhandene Jobs umgeschaltet werden.
- Im Reiter „**Upload**“ können weitere zuvor definierte Jobs oder ganze Jobsätze vom Viewer aus auf den Sensor geladen werden.

## 7.6 Context help

Context-sensitive help pages are available for all important program functions: As soon as you select a certain function on the program interface, you will receive the appropriate information in the help window at the top right (Help tab).

To view all available help pages, select "Help" from the menu or click the button with the "?" symbol or double-click in the context help window. There, you also can also search for terms or keywords. In comparison to the context-sensitive help, the size of this help window can be enlarged to view longer text more clearly.

Used open-source software: Open Source Licenses (Page 2)

# 8 VISOR® Software – SensoFind

In this program, you can select a sensor or a sensor simulation for configuration, or for display (monitoring), as well as various basic settings:


- Active sensors (Page 70)
- Sensors for simulation mode (Page 71)
- Add / find active sensor (Page 72)
- Configuring a connected sensor (Page 76)
- SensoView – Quick Start Guide (Page 67)
- Sensor network settings (Page 76)
- Firmware update (file) (Page 78)
- User administration / Passwords (file) (Page 77)
- Favorites (Page 72)
- Autostart file (file) (Page 79)

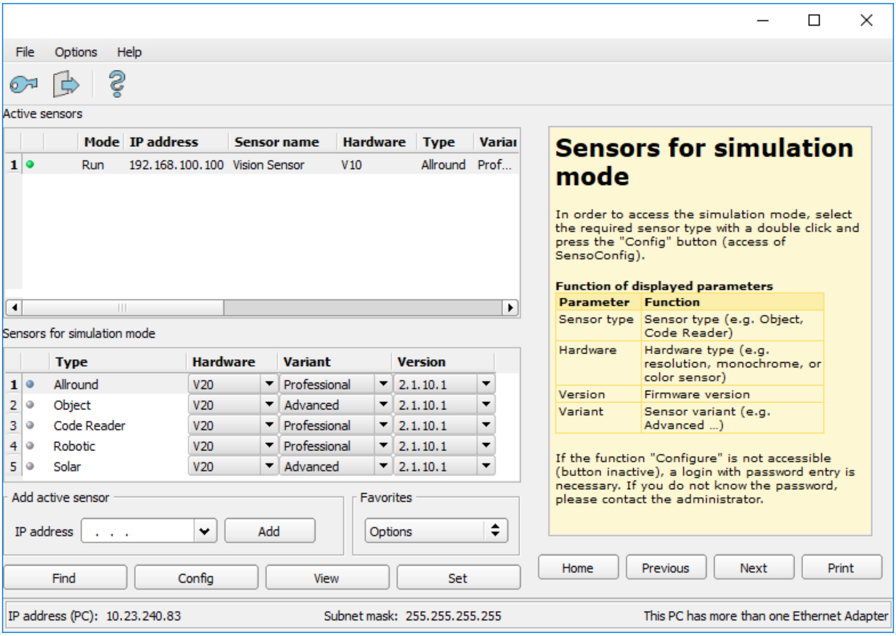




*Fig. 25: SensoFind*

If the function "Config" is not accessible (button inactive), a login with password entry is necessary. Click on the button with the door symbol and arrow:

If you do not know the password, please contact the administrator.

## 8.1 Active sensors

All sensors available in the connected network are displayed in the drop-down list Active sensors. In the first column, an LED indicates the operating mode of the VISOR®. Green: Device is in run mode, yellow: Device is in configuration mode, red: Error/device start

**NOTE:**

- If no entries are shown in the list, even though a sensor is connected, you can enter it using the "Find" button or the "Add" button.
- If no sensors are connected, the Sensors for simulation mode list will show available simulations for various sensor applications.

Clicking on the Details button (at the right end of the "Active sensors" parameter list) will open an even more detailed list of the VISOR® parameters.

Sensor properties ? ×

| Property | Setting |
|---|---|
| **Sensor name** | vision sensor |
| **Type code** | V20-RO-P3-R-M-M2-L |
| **Hardware** | V20 |
| **HW Minor Version** | 1 |
| **Sensor type** | Robotic |
| **Variant** | Professional |
| **Function restrictions** | - |
| **Firmware version** | 2.1.11.1 |
| **Mode** | Config |
| **Error Status** | No Error |
| **Status (out)-Port** | Open |
| **Status (in)-Port** | Status not available |
| **MAC address** | 00-19-6F-0C-5D-58 |
| **IP address** | 192.168.178.45 |
| **Subnet mask** | 255.255.255.0 |
| **Gateway** | 192.168.178.1 |
| **DHCP** | Enabled |
| **Origin IP address** | DHCP |
| **Ethernet** | Not selected |
| **Industrial Ethernet** | n/a |
| **Status fieldbus** | Off |
| **Lens ID** | 27 (12.0 mm) |
| **LED type** | 1 (Red) |
| **Target laser** | Available |

Copy OK

*Fig. 26: Sensor properties*

**Right-click on Active sensors in SensoFind:**

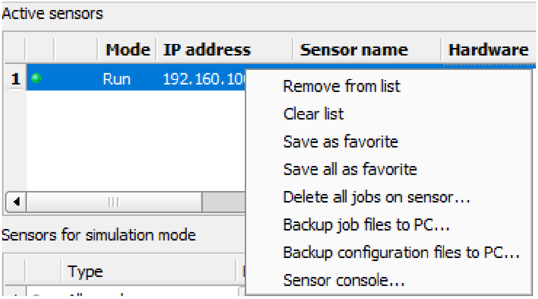



*Fig. 27: Right-click on Active sensor*

| Parameter | Function |
|---|---|
| Remove from list | Removes the selected sensor from the "Active sensors" list. |
| Clear list | Clears the complete list "Active sensors". |
| Save as favorite | Saves the selected sensor or all sensors in the list as favorite(s). Additional information: Favorites |
| Save all as favorite | |
| Delete all jobs on sensor... | Deletes all jobs on the sensor. Information cannot be recovered. A restart is required after deletion. |
| Backup job files to PC... | Saves job files on the PC in the specified directory. |
| Backup configuration files to PC... | Saves the configuration files on the PC in the specified directory. |
| Sensor console... | Opens a console (ssh) to the selected device. |

Additional information:

Configuring a connected sensor (Page 76)SensoConfig (Access of SensoConfig)

Display images and result data (Page 76)SensoView (Access of SensoView)

## 8.2 Sensors for simulation mode

In order to access the simulation mode, select the required sensor type with a double click or press the "Config" button (access of SensoConfig).

**Function of displayed parameters**

| Parameter | Function |
| --- | --- |
| Sensor type | Sensor type (e.g. Object, Code reader, ) |
| Hardware | Hardware type (e.g. resolution, monochrome, or color sensor) |
| Version | Firmware version |
| Variant | Sensor variant (e.g. Advanced …) |

If the function "Config" is not accessible (button inactive), a login with password entry is necessary. If you do not know the password, please contact the administrator.

## 8.3 Add / find active sensor

If no entries are displayed in the "Active sensors" drop-down list, even though a sensor is connected, proceed as follows:

**Search Find / sensor:**

To search for sensors which are connected to the PC, or which are available in the network, click the button "Find".

**Add active sensor:**

If you know the IP-address of a sensor, please enter it in the field IP-address and click the button "Add".

Now the sensor appears in the list and can be configured (button "Config") or displayed (button "View").

If the function "Config" is not available (button not active / grayed out), a login with password input is necessary. If you do not know the password, please contact your system administrator.

## 8.4 Favorites

The favorites are used for quick access and management of the VISOR® vision sensors. The following parameters can be selected for the favorites (by right-clicking or in the "Favorites" area in SensoFind.

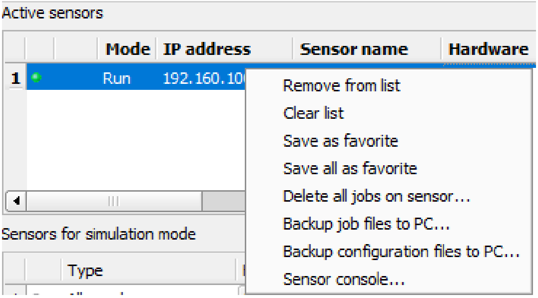



*Fig. 28: "Favorites" parameters by right-clicking on an active sensor*

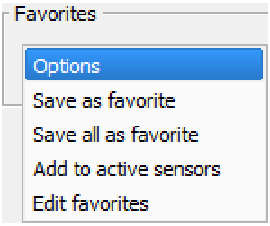



*Fig. 29: Favorites options*

**Parameter description:**

| Parameter | Function |
|---|---|
| Save as favorite | Opens the "Save as favorite" window where a desired destination can be selected in the tree structure in which the sensor from the "Active sensors" list should be saved as a favorite. |
| Save all as favorite | Opens the "Save all as favorite" window where a desired destination can be selected in the tree structure in which all the sensors from the "Active sensors" list should be saved as a favorite. |
| Add to active sensors | Opens the "Add to active sensors" window where a sensor/ sensor group can be selected that should be added to the "Active sensors" list. |
| Edit favorites | Opens the "Edit favorites" window in which the sensor groups can be managed / edited. |

**Edit favorites - create groups**

In the left window area, the sensors are divided into groups via a tree structure, e.g. according to production sites and production lines. In the right window area, the sensors below a selected group are listed in tabular form, e.g. the group "Favorites" shows all sensors.

To create a group, right click on "Favorites" or an existing group / "Add group".

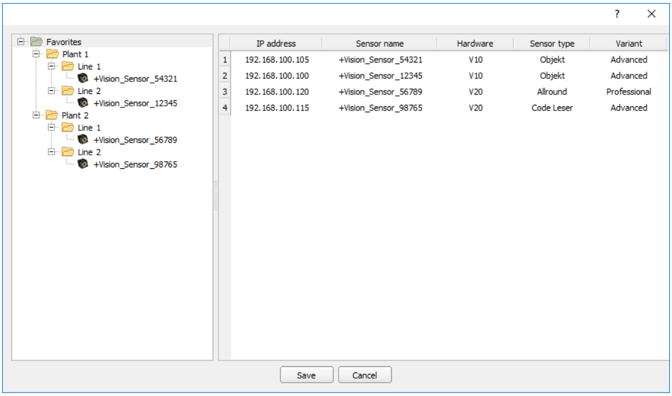



*Fig. 30: Group configuration*

Favorites are saved as an XML file in the installation path for the VISOR® vision sensor on the PC. The file is located in the directory "SensoPart/VISOR Vision Sensor/SensoFind/Data". An exchanged between different PCs can take place.

Examples for using favorites:

**Example 1:**

VISOR® vision sensors that are integrated into various networks can be viewed and managed locally in SensoFind (please refer to the following figure as well). The sensors can be added to the "Active sensors" list by entering the IP address in the field "Add active sensor". The sensors are subsequently managed via the favorites. The sensors can be added to favorites by "SensoFind/Favorites/Saves as favorite". Within the favorites, the sensors can be assigned to different groups.

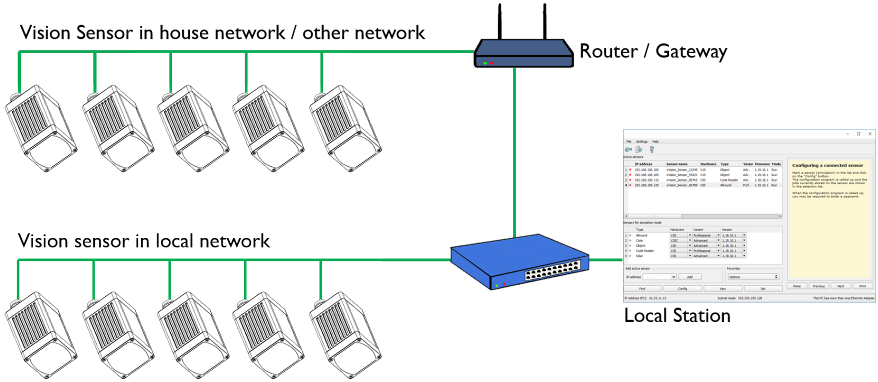



*Fig. 31: Example 1 - VISOR® on various networks*

**Example 2:**

Multiple stations are in the same local network. All users have access to all VISOR® vision sensors, even though they only need some of these for their work (please refer to the following figure as well). When using the autostart function (please refer to Autostart file (file) (Page 79) as well), you can choose for only a specific selection of VISOR® vision sensors (favorites) to be shown. The sensors must therefore be added to the favorites and divided into groups. A group of favorites can then be selected in the Autostart file. Users will then only have access to the relevant sensors when opening SensoFind via the Autostart file.

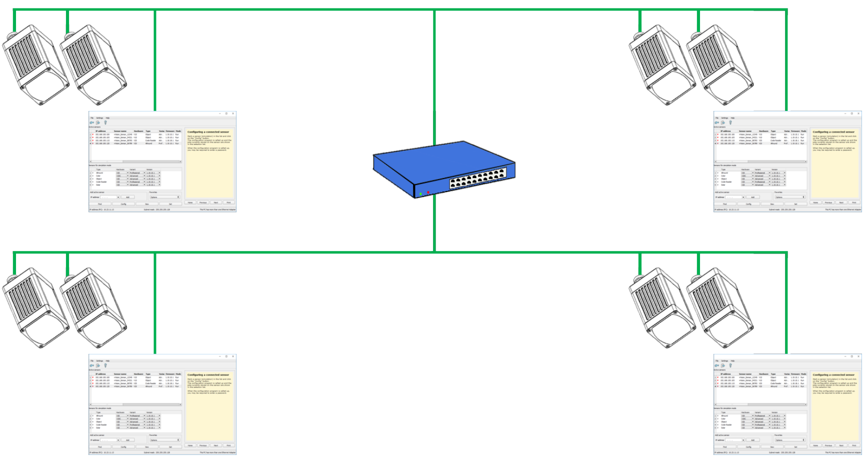


*Fig. 32: Example 2 - Favorites in the Autostart file*

## 8.5 Configuring a connected sensor

Mark a sensor (or simulation) in the list and click on the "Config" button. The configuration program SensoConfig is accessed and any jobs stored on the sensor are displayed in the drop-down list. When SensoConfig is accessed, you may be required to enter a password. See User administration / Passwords (file) (Page 77) for defining passwords.

See Chapter: VISOR® Software – SensoConfig

## 8.6 Display images and result data

Select a sensor in the list and click on the " View " button. The SensoView program is started and images and measurement results from the active job are displayed. SensoView is not available for sensors in simulation mode.

**NOTE:**
Accessing SensoView does not affect the operation of the selected sensor.
See Chapter: VISOR® Software – SensoView

## 8.7 Sensor network settings

You can change the network settings of the selected sensor with the "Set" button. The IP address, subnet mask, standard gateway, DHCP, and sensor name can be set here. The PC's IP address and subnet mask are displayed below in the SensoFind status bar.

To connect the sensor to the PC, the address spaces must match. If necessary, set the IP address, etc. of the sensor accordingly here. Please contact your administrator to set network parameters. For more information, please refer to sections Network settings Short guide and Network connection.

If "DHCP = active" is selected, a name must be assigned for the sensor, since the IP address can then be reassigned every time the sensor is started and can thus change, i.e. is no longer unique. You require administrator rights for these functions (see User administration).

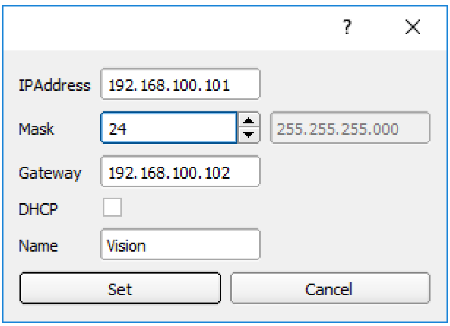



*Fig. 33: SensoFind IP setup*

See Chapter: Network settings Short guide and Network connection

## 8.8 User administration / Passwords (file)

The VISOR® configuration software distinguishes between three user groups with different authorizations: (button in the upper left corner with key symbol)

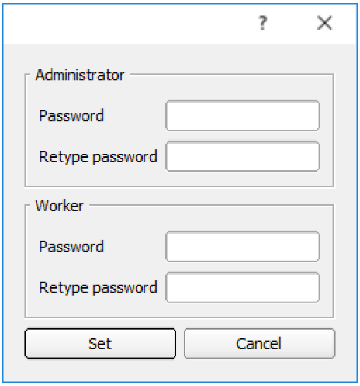



*Fig. 34: SensoFind, Passwords*

| Password level | SensoFind | SensoConfig | SensoView |
|---|---|---|---|
| Administrator password | All functions | All functions | All functions |
| Worker password | All functions except<br>• Configurating<br>• Settings<br>• Update | none | All functions, including Job Upload and Image Recorder |
| User (without password) | All functions except<br>• Configurating<br>• Settings<br>• Update | none | Only display of images, inspection res-ults, and statistics |

After installing the software, the login is automatically executed immediately when the application is called without password prompt. No passwords are assigned.

### Define passwords

Select User Administration in the File menu or click on the button with the key symbol in the toolbar to assign or change passwords for the user categories Administrator and Worker. Once a password has been entered, a logout is automatically carried out, i.e. input of the new password is now necessary. When assigning an "empty" password, the entry can be simply confirmed with OK.

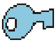


*Fig. 35: Password button*

### Login / Logout

After setting passwords, login is necessary, e.g. for configuring a sensor. To do this, click on the login button in the toolbar, enter the specified password, and confirm with "OK". If the checkmark is set to "Disable password query", the password will no longer be requested the next time the application is started. To log out of the user group, click on the logout button.

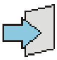


*Fig. 36: Login button*

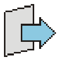


*Fig. 37: Logout button*

## 8.9 Firmware update (file)

You can update the firmware of the selected sensor through the menu item "SensoFind/File/Firmware Update" (see following figure). For this, the corresponding firmware update file must first be obtained from the SensoPart homepage or from SensoPart support.

In the dialog that is opened, select the appropriate firmware file and follow the instructions. Do not disconnect the power to the sensor during this process unless prompted to do so by a screen instruction.

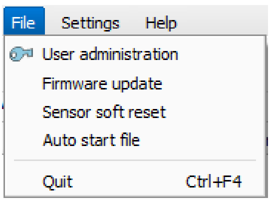



*Fig. 38: SensoFind, firmware update*

**ATTENTION:**
Before executing the firmware update, please create a current backup!
To do this, save the jobsets via the menu item "SensoConfig/File/Save jobset as..."

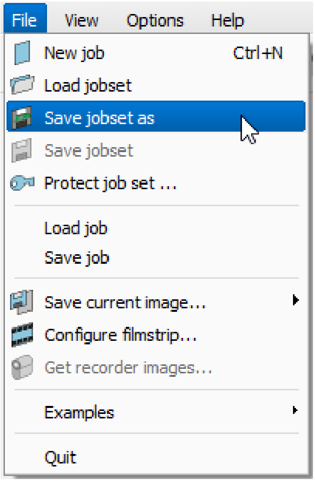



*Fig. 39: Create backup, Save job set under*

## 8.10 Autostart file (file)

Autostart makes it possible to start the VISOR® software software automatically. For this purpose, a batch file is created which can be stored in the Windows system folder "Autostart" so that it can be accessed automatically every time the PC is started. The Autostart window is divided into the areas: mode, window settings, and user.

**Procedure**

1. Open the Autostart file in the SensoFind module with the file path: SensoFind/File/Autostart File
2. You can define which VISOR® software software module should be automatically started in the "Mode" section.
3. In the window settings, select the view of the module: Normal or full screen without title bar in the panel PC mode.
4. In the "User" area, the user for the Autostart file is specified. For more information on user role permissions, please refer to User administration / Passwords (file) (Page 77)
5. Select the "Save" button and save the batch file (.bat) to the desired destination. For an automatic start when the PC boots, the file must be stored in the Windows system folder "Startup".
6. Close the VISOR® software.
7. Execute the batch file. The VISOR® software software will start as specified in the configured settings.

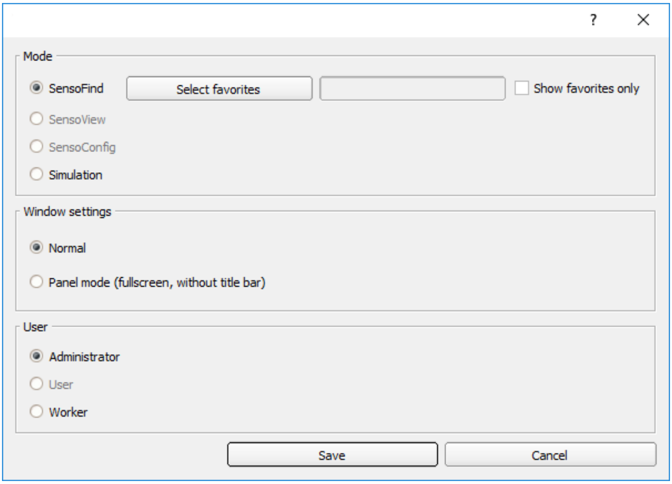



*Fig. 40: Autostart file*

The following parameters can be set in the Autostart window:

| **Mode** | |
|---|---|
| **Parameter** | **Function** |
| SensoFind | VISOR® software software modules that should be opened automatically in the autostart file.<br>For the start of the simulation mode, the model variant which is currently selected in SensoFind (marked in blue) is used. |
| SensoView | |
| SensoConfig | |
| Simulation | |
| Select favorites | This parameter can be used to add a favorites group to the Autostart file. |
| Show favorites only | If the "Select favorite list" parameter is selected, the "Active sensors" list will be emptied and then only filled with the selected favorites. |

| **Window settings** | |
|---|---|
| **Parameter** | **Function** |
| normal | In the autostart file, the selected VISOR® software module will be opened normally with the title bar. |
| Panel PC mode (Full-screen without title bar) | Das ausgewählte VISOR® software Modul wird in der Autostart-Datei im Vollbild ohne Titelleiste geöffnet. Typical application for touchscreen panel PCs. |

| **User** | |
|---|---|
| **Parameter** | **Function** |
| Administrator | The selection of the user depends on the rights that the user will have within the Autostart file. For more information on user role permissions, please refer to User administration / Passwords (file) (Page 77) |
| User | |
| Worker | |

# 9 VISOR® Software – SensoConfig

You can use this program to configure your vision sensor, in six setup steps, for one or more jobs:

- Setup Job (Inspection tasks) (Page 83)
- Setup Alignment (Page 144)
- Setup Detectors (Page 170)
- Setup step Output (Page 311)
- Setup Start sensor (Page 340)

**Other program functions:**

- Trigger / Image update (Page 341)
- Connection mode (Page 342)
- Simulation mode: Simulation of jobs (offline mode) (Page 354) using series of images (filmstrip)
- Filmstrips (file) (Page 348) for analysis or simulation purposes.
  The execution of SensoConfig may require the entry of a password (user group: Administrator). Please refer to: User administration / Passwords (file)
- Image recorder (Page 360)

## 9.1 Setup Job (Inspection tasks)

A job contains all the settings and parameters required to perform a specific inspection task.

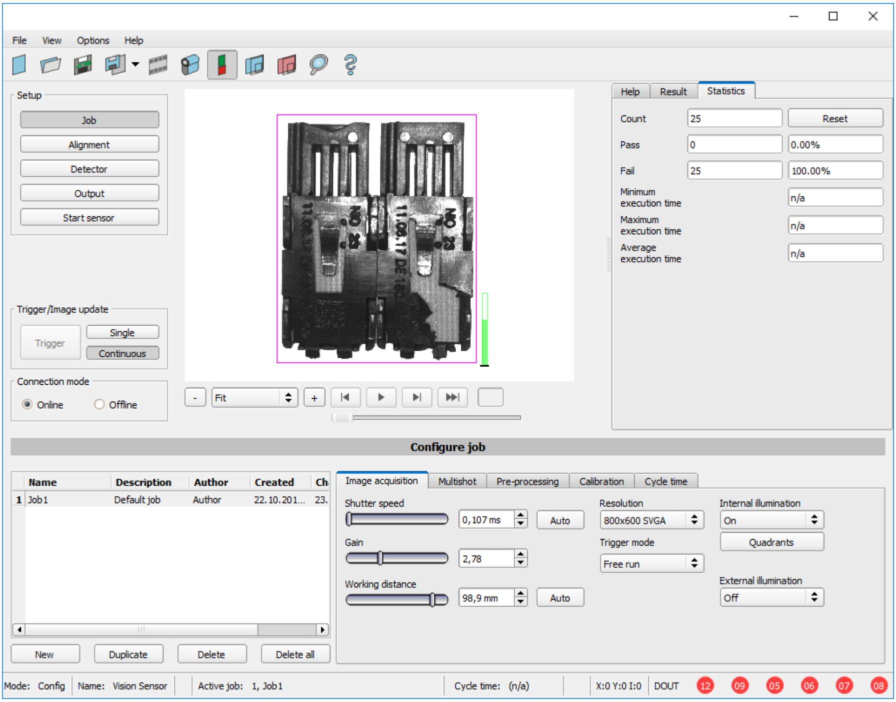



*Fig. 41: SensoConfig, setup step Job*

## 9.1.1 Creation, modification, and administration of jobs

You can edit a selected job (marked in the list on the bottom left) by entering parameters in the tabs of the configuration window (right, bottom).

If there is no job entry in the list, you must create a new job first.

### Creating a new job

1. Click on the button "New" underneath the job drop-down list. A new job entry appears in the list.
2. Edit the entry with a double click on the respective line (Name, Description, Author)

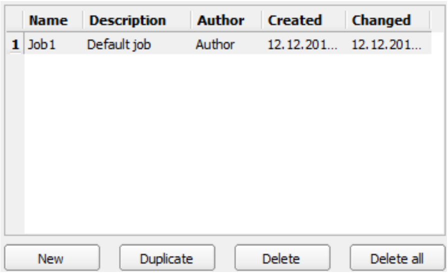



*Fig. 42: SensoConfig Job list*

### Further functions

| Function | Description |
| --- | --- |
| New | Defines a new job |
| Duplicate | Adds a copy of the selected job to the job set. |
| Delete | Deletes the selected job from the list |
| Delete all | Deletes all the jobs in the list |

If the capacity of the sensor memory is exhausted and no further jobs can be loaded onto the sensor, the color of the remaining memory indicator in the status line (below) changes to red.

### Job templates

It is possible to save a job as a template. To do this, right-click on the job in the job list and select "Save as template". For each new job, the settings and detectors are then copied from the job template. In the job list, the job template is identified with a "T" (Template). The job template cannot be edited. To remove the job template, right-click on the template and select "Remove".

When creating new jobs from the job template, the job set parameters are not changed.

### Copy job parameters

By right clicking on a job, job parameters such as Image acquisition or calibration settings can be copied to another job. The respective parameters and the target jobs can be selected in the dialog.

Additional information:

Open and save job or jobset (file) (Page 343)

Protect jobset (file) (Page 345)

## 9.1.2 Image acquisition tab

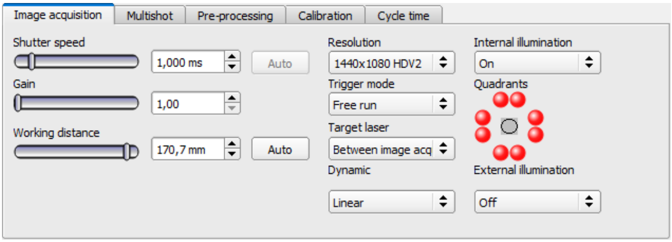



*Fig. 43: Image acquisition tab*

In the Image acquisition tab, you define the basic parameters of image acquisition.

| Parameter | Functions and setting possibilities |
|---|---|
| Shutter speed | Parameters for controling the image brightness. Image brightness should preferably be set with the shutter speed. Only in the second step, if necessary, adjust the gain (default gain = 1). With moving objects, a slower shutter speed can cause motion blur in the image.<br><br>Auto: With the button "Auto" the exposure can be set automatically.<br>The maximum shutter speed that can be configured is 100 ms. The duration of the internal exposure pulse is limited to 8 s. Shutter speeds longer than 8 ms only make sense in cases in which external lighting (or both internal and external lighting) is used. |
| Gain | Parameters for controlling the image brightness. The image brightness should preferably be adjusted with the shutter speed; only adjust the gain in the second step if necessary (default gain = 1) |
| Working distance | Parameters for configuring the working distance.<br>Auto: With the button "Auto" the approximate working distance can be set automatically. Fine adjustment is possible using the slider or value adjustment (Additional information: Focusing / Working distance (Page 87)). |

| Parameter | Functions and setting possibilities |
|---|---|
| Resolution | Available resolutions:<br>V10 / V10C: SVGA (800x600), QSVGA (400x300), QSVGA Zoom 2 (400x300)<br>V20 / V20C: HDV2 (1440x1080), WGA (720x540), WGA Zoom 2 (720x540)<br>V50 / V50C: QSXGA (2560 x 1936), SXVGA (1280 x 968), SXVGA Zoom 2 (1280 x 968)<br>For time-critical applications or for compatibility reasons, a lower resolution can be selected.<br>**Attention:** Bei Änderung der Auflösung werden alle bereits definierten Detektoren gelöscht! |
| Zoom | By selecting a resolution level with zoom, different image sections with different image sizes can be achieved. |
| Trigger mode | Selection option that can be used to define whether the vision sensor should be operated in trigger mode or in free run mode.<br>Trigger: In the triggered mode, the pin 03 WH trigger input or one of the interfaces can be used to trigger an image acquisition.<br>Free run: In free run mode, the vision sensor will continuously capture images and run evaluations. |
| Target laser | The target laser is used to align the sensor.<br>Target laser options: Off / During image acquisition / Between image acquisition operations |
| Internal illumination | Switch for internal illumination (on / off)<br>The internal illumination is limited to 50 ms. If a longer shutter speed is set, the internal illumination switches off at 50 ms. |
| Quadrants | By clicking on the LED quadrants, individual quadrants of the lighting can be switched on / off (shown as two red dots). This function can suppress reflections at low working distances. |
| External illumination | External lighting options: Off / On / Permanent. The external lighting is switched using pin 09 RD. |

In order to get a continuously updated live image without triggers, configure the following settings as shown:

1. In the Job setup step, open the Image acquisition tab and set the Trigger mode to Free run.

2. Under Trigger / Image update, select "Continuous".

#### Focusing / Working distance

The parameter is Working distanceused to set the working distance at which the image is focused. You can use the slider, or edit the values, to do a fine adjustment.

| Parameter | Function |
|---|---|
| Auto | With the button "Auto" the approximate working distance can be determined automatically. The violet search area is used for the determination (see Working distance (Page 88)). If several possible sharp layers are found in the search area, the dialog "Layer selection list" appears. The corresponding working distance can be selected here. The value "Score" indicates a measure for the sharpness of the image (greater = sharper). The corresponding working distance is accepted by clicking on it. |

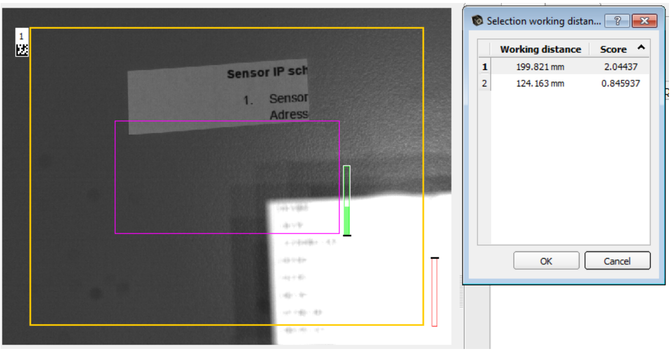



*Fig. 44: Working distance*

Different jobs can be set to different working distances. The time required for the job change can be extended by up to 2 seconds by moving to the working distances. Approx. 1 job change per minute is possible.

### 9.1.3 Multishot tab

**NOTE:**
The Multishot function is available only in Allround and Professional versions and not for color sensors.

When using the Multishot function, each sequence involves acquiring 4 images of an object. Each time an image is taken, the object is illuminated from a different direction. The four images are then combined into one image. Due to the different reflections, a "virtual height image" can be calculated which contains information that is not visible in the individual images. This way, the finest depressions or elevations in the considered surface can be detected.

**This technology is especially suitable for:**

- Detecting defects on flat surfaces e.g. scrapes or scratches
- Reading raised or imprinted fonts using OCR

- Nailed Data matrix codes
- Detection of details on high-resolution surfaces
- Detection of Braille dots

**This technology is not suitable for:**

- Imaging of moving objects
- Heavily curved surfaces
- 3D applications
- Detection of details that are shaded by other parts of the component and thus cannot be illuminated from all 4 sides.

The following **conditions** must be considered:

1. Use ring light and place object in its focus. If no ring light is used, place the light source as far away from the measurement object as possible.
2. Optimize image acquisition: Avoid overexposed or dark areas, shadows and blurred areas in all four images.
3. The test object must be still during the image acquisition operations (four images in a single sequence), i.e., it must be stationary relative to the sensor. Exception: movement of constant speed in x-direction (see Image offset X axis (Page 91)).

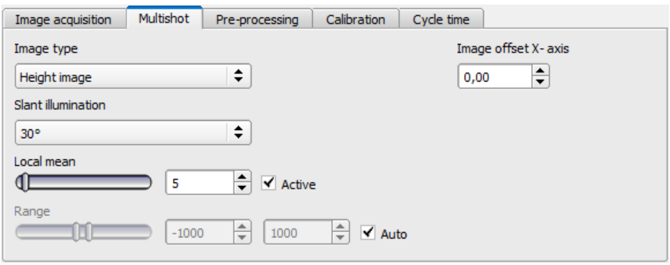



*Fig. 45: "Multishot" function, parameters*

### 9.1.3.1 Image types

| Image type | Description |
|---|---|
| Image curvature | Shows virtual (estimated) curvature values (positive and negative) mapped to gray values. The curvature values indicate how much the tilt of the surface changes at a given point. |
| Curvature, amount | Shows virtual (estimated) curvature values, but here only the absolute amount, mapped to gray values. |

| Image type | Description |
|---|---|
| Height image | Shows virtual (estimated) height values, scaled to gray values.<br>**NOTE:**<br>Selecting "Height image" causes longer execution time. |
| Albedo image | Shows virtual (estimated) reflectivity values mapped to gray values. |
| Mean image | Mean of four single images |
| Combined image, quadrants | All four single images combined in one quadrant image. This function can be used to adjust the lighting. Avoid overexposed or dark areas, shadows and blurred areas. |
| Combined image, horizontal | All four single images arranged horizontally on top of each other combined in one image. If the test object was moved uniformly during image acquisition, the position offset of the individual images in the X-direction can be made visible and compensated with the function "Image offset X-axis".<br>The image order from top to bottom is: East, West, North, South. |
| East | Single image, illumination from east |
| North | Single image, illumination from north |
| West | Single image, illumination from west |
| South | Single image, illumination from south |

**Image type parameters**

| Parameter | Function |
|---|---|
| Image type | Select Image type (see above) |
| Slant illumination | Angle of illumination with respect to the surface of the object (0 ° = flat from the side; 90 ° = perpendicular from above) |
| Background flattening | If the sensor is not exactly perpendicular to the object or the illumination is inhomogeneous, the calculated height image may appear strongly tilted. The local smoothing of the height differences helps to correct the tilting. The smoothing is carried out via the set number of pixels. |
| Range | Value range of virtual height and angle values. The set range is mapped to a gray value image 0...255. With this range selection it is possible to obtain an optimal gray value spread of the area of interest. With "Auto", this value is automatically calculated from the minimum and maximum values found in the image. |

| Parameter | Function |
|---|---|
| Image offset X axis | Generally, the object should rest during the 4 image acquisitions. However, a movement of constant speed in x-direction can be compensated by the parameter "Image offset X-axis". This parameter specifies by how many pixels the object is shifted in the X direction in the successive images.<br>If the inspection object was moved uniformly during image acquisition, the position offset of the individual images can be compensated for one another together with the image type "Combined image, horizontal". |

### 9.1.3.2 Multishot Illumination

Correct illumination is important for use. The object must be illuminated from all four sides. The VISOR® sensor automatically controls the image acquisition sequence. For ease of illustration, the four illumination directions are referred to below analogously to the four directions (north top).

**The result is the following arrangement of illumination:**

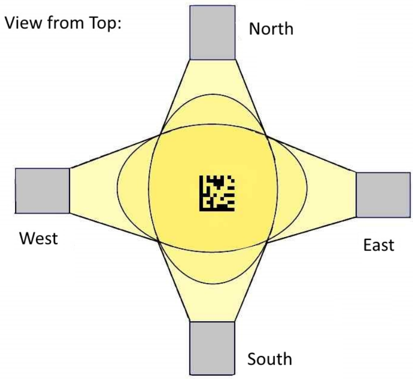



*Fig. 46: Multishot, Orientation of illumination*

**Assignment of the illuminations to the VISOR® connections:**

| Direction | Output pin (old) | Output pin (new) |
|---|---|---|
| East | 09 | 09 |
| South | 07 | 06 |
| West | 06 | 07 |
| North | 05 | 08 |

The correct connection of the illumination can be checked by using the function "Combined image, quadrants", which displays all 4 single images together in one image. Place an object in the image that casts a clear shadow (e.g. a screw). The images are combined according to the following scheme:

| | |
|---|---|
| **Top left:**<br>Illumination from north,<br>shadow to south | **Top right:**<br>Illumination from east,<br>shadow to west |
| **Bottom left:**<br>Illumination from west,<br>shadow to east | **Bottom right:**<br>Illumination from south,<br>shadow to north |

**In the image it looks like this:**

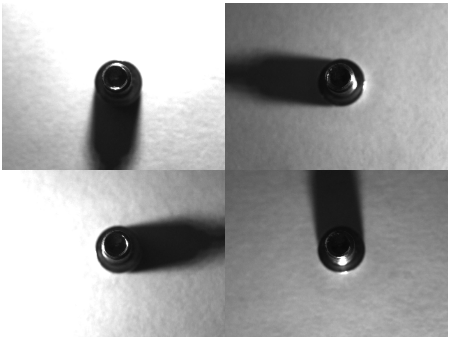


*Fig. 47: Multishot, Single images*

**Further advice for illumination:**

- Avoid both overdriven areas of the image and dark shadows
- The SensoPart illumination can be mounted in 30° or 60° angles
- Use 30° angle to illuminate parts flat from the side (avoids reflections)
- Use the 60 ° angle to illuminate parts steeply from above (amplified reflections)

### 9.1.4 White balance tab

White balance is used to compensate for a possible color cast in the image due to lighting conditions or camera chip. The White balance tab is only available for color sensors.

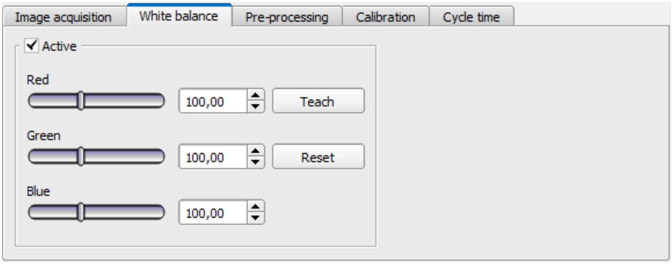



*Fig. 48: White balance tab*

| Parameter | Function |
|---|---|
| Red | Mean of red channel in image |
| Green | Mean of green channel in image |
| Blue | Mean of blue channel in image |
| Teach | Execution of white balance: For white balance, there should be a homogeneous, white or slightly gray area under the camera to position |
| Reset | Reset values |

## 9.1.5 Pre-processing tab

The Pre-processing tab can be used to filter or rearrange the images captured by the sensor before they are analyzed.

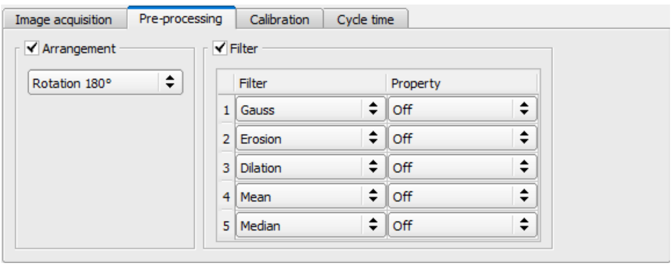



*Fig. 49: Pre-processing tab*

- Up to 5 filters can be activated, which are executed in the specified order.
- All detectors (Alignment and standard detectors) will work on the preprocessed image, not on the original image.

**Arrangement filters**

| Arrangement type | Effect |
|---|---|
| Rotation 180° | Rotation of image by 180° |
| Horizontal mirroring | Horizontal mirroring of the image |
| Vertical mirroring | Vertical mirroring of the image |

**Filter for image improvement**

In particular, with the morphological operators (dilation and erosion), improvements of the image can also be achieved in combination, e.g. by successive erosion and dilatation or vice versa.

Example: Black spots with a bright background can be eliminated when dilatation and erosion follow one another.

| Filter type | Effect |
|---|---|
| Gauss | The image is smoothed with a gaussian filter. This can be used to reduce noise, suppress interfering details and artifacts, and smooth edges. |
| Erosion | Extension of dark areas, elimination of bright pixels in dark areas, elimination of artifacts, separation of bright objects. Effect: Each gray value is replaced by the minimum gray value within the filter mask (e.g. 3x3 filter mask). |
| Dilation | Extension of bright areas, elimination of dark pixels in dark areas, elimination of artifacts, separation of dark objects. Each gray value is replaced by the maximum gray level found within the filter mask (e.g. 3x3). |
| Median | Each gray value is replaced by the median value of the pixels found within the filter mask (e.g. 3x3). Typical application: Smoothing the image, suppressing image noise, especially local light or dark areas / pixels (salt and pepper noise) |
| Mean | Each gray value is replaced by the mean of the pixels found within the filter mask (e.g. 3x3). This can be applied for reduction of disturbances, suppression of disturbing details and artifacts and smoothing the image. |
| Range | Each gray value is replaced by the range value (maximum gray level − minimum gray level) of the pixels found inside the filter mask (e.g. 3x3). Typical applications: Detection and enhancement of edges and improvement of local image contrast. |
| Standard deviation | Each gray value is replaced by the standard deviation of the pixels found within the filter mask (e.g. 3x3).<br>Typical applications: Highlighting surface defects or edges. |

| Filter type | Effect |
|---|---|
| Edge filter (Sobel) | The filtered image contains edges that were found using the Sobel algorithm (see also image-processing literature).<br>Typical applications: Detecting and improving edges, improving local contrast, and detecting surface defects. |
| Multiplication | The gray value of each pixel is multiplied by the chosen multiplier (2x, 4x, 8x, etc.). The value range is limited to 255. |
| Inversion | Inversion of image pixel / gray value |
| Background flattening | If the sensor is not aligned exactly perpendicular to the object or the illumination is inhomogeneous, a brightness gradient may be visible in the background of the image. Local flattening of the brightness values helps to correct this effect. The flattening is carried out via the set number of pixels. |

The effect of an activated filter can be seen immediately in the image. The larger the filter core is selected, the stronger the filtering effect. The filters are executed in the order given from top to bottom (1-5).

**Configuring the filters**

1. Select the filters in the desired order using the drop-down menus in the Filter column.
2. Enter the size of the filter core in the drop-down menu in the "Setting" column. If the setting is "Off", the respective filter is deactivated.

## 9.1.6 Calibration tab

The calibration allows the conversion of image coordinates (pixels) into world coordinates (e.g. millimeters). When using this function, all coordinate outputs (positions and measurement results) are calculated and output in the selected unit.

### 9.1.6.1 Select the calibration method

The calibration methods are distinguished into two fields of application:

- "Measurement": Calibration methods for applications in the field of measurement and testing
- "Robotics": Calibration methods for applications in the field of robotics

**NOTE:**
The calibration methods described below are suitable for standard lenses, integrated or C-mount.
Only the method "Scaling (Measurement)" is suitable for telecentric lenses.

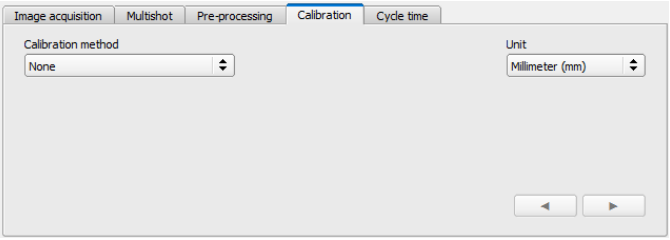



*Fig. 50: Select the calibration method*

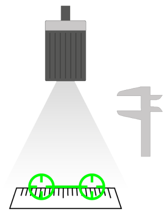
| Parameter | Function |
|---|---|
| Calibration method | Selection of a calibration method:<br>• **None**: Calibration not active, coordinate determination, display and output in pixels [px]<br>• Calibration plate (Robotics)<br>• Point pair list (Robotics)<br>• Hand-Eye calibration (Robotics)<br>• Base-Eye calibration (Robotics)<br>• Calibration plate (Measurement)<br>• Scaling (Measurement) |
| Unit<br>(user unit) | Desired unit for world coordinates. The following units are available:<br>• mm (millimeter)<br>• cm (centimeter)<br>• m (meter)<br>• in (inch)<br>• Arbitrary unit (au)<br>Note: If no calibration has been performed, all values refer to pixels. |

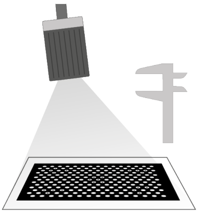
| Parameter | Function |
|---|---|
| Robot: Order of rotation | For 3D references, the order of the specified rotations must be observed. You can choose between the two most common pose types:<br>• Yaw-Pitch-Roll (e.g. Stäubli)<br>• Roll-Pitch-Yaw (e.g. Kuka, Fanuc, Hanwha, ABB*, UR**)<br>Note: Here the rotation is referred to the "old" coordinate axes of the reference coordinate system. If you are using a robot whose rotation order refers to the new axes created by the rotation, the following applies:<br>• Roll-Pitch-Yaw (new axes) = Yaw-Pitch-Roll (old axes)<br>• Yaw-Pitch-Roll (new axes) = Roll-Pitch-Yaw (old axes) |
| ◄ / ► | Go to next / previous step |

*ABB robots use quaternions as order of rotation. To communicate with the VISOR®, you need to convert the coordinates on the robot to Roll-Pitch-Yaw.

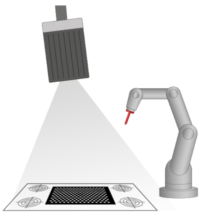
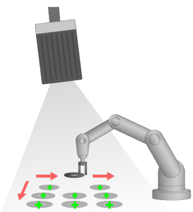
**UR robots use "Axis-Angle" as rotation order. However, they do support a function that converts them into the rotation order "Roll-Pitch-Yaw". Use this function and select Roll-Pitch-Yaw in SensoConfig.

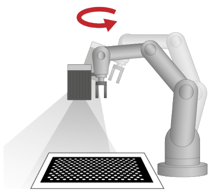
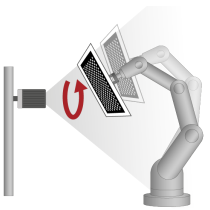
### 9.1.6.1.1 Overview: Calibration methods "Measurement"

| Calibration method | Function |
|---|---|
| Scaling (Measurement) | |
|  | • Relative determination of e.g. distances in world coordinates (e.g. mm) using a simple ratio factor<br>**Note:**<br>• Limited accuracy<br>• No correction of distortions.<br>Additional information: Calibration method "Scaling (Measurement)" |

| Calibration method | Function |
|---|---|
| Calibration plate (Measurement) | |
|  | • Relative determination of e.g. distances in world coordinates (e.g. mm) via the image acquisition of a calibration plate<br>• High accuracy<br>• Correction of tilt and lens distortion<br>• Two variants:<br>  • Single image calibration<br>  • Multi-image calibration<br>Additional information: Calibration method "Calibration plate (Measurement)" |

### 9.1.6.1.2 Overview: Calibration methods "Robotics"

| Calibration method | Function |
|---|---|
| Calibration plate (Robotics) | |
|  | • Determination of absolute positions in robot coordinates (e.g. mm)<br>• Correction of tilt and lens distortion<br>• Variants: Single image calibration, Multi-image calibration<br>Additional information: Calibration method "Calibration plate (Robotics)" |
| Point pair list (Robotics) | |
|  | • Determination of absolute positions in robot coordinates (e.g. mm)<br>• Correction of tilt and lens distortion<br>Additional information: Calibration method "Point pair list (Robotics)" |

| Calibration method | Function |
|---|---|
| Hand-Eye calibration (Robotics) | |
|  | • Determination of absolute positions in robot coordinates (e.g. mm)<br>• Correction of tilt and lens distortion<br>• Determination of the Hand-Eye reference (reference robot TCP to Camera coordinate system)<br>• Enables shifting the image acquisition position<br>Additional information: Calibration method "Hand-Eye calibration (Robotics)" |
| Base-Eye calibration (Robotics) | |
|  | • Determination of absolute positions in robot coordinates (e.g. mm)<br>• Correction of tilt and lens distortion<br>• Determination of the Base-Eye reference (reference robot base to Camera coordinate system)<br>Additional information: Calibration method "Base-Eye calibration (Robotics)" |

**NOTE:**
All result values for positions and measurement results are corrected. However, in order not to burden the cycle time, i.e. to extend it, the image data are not converted or equalized! Thus, even with active calibration, a high execution speed is guaranteed.

### Status LED

As soon as a calibration method is selected, the status LED is shown on the left side next to the tab title "Calibration". If the calibration is active, all affected functions, e.g. detectors, are only carried out correctly if the calibration is valid (=green), i.e. if it was carried out successfully.

| Color of Status LED | Status of the calibration | Meaning / Measures |
|---|---|---|
| ● Green | Valid | No action required |
| ● Yellow | Valid | Deviations. Recommendation: Re-calibrate |
| ● Red | Invalid | Check the calibration object and re-calibrate |

**NOTE:**

- With method "Scaling (Measurement)" only "green" is possible: Default or input values result in the scaling factor. No error calculation possible.
- "Point pair list (Robotics)": "Green" is displayed for a new job. Default values (9 points) result in correct default calibration.
- All methods with calibration plate: When a new job is created, "Red" appears, because no calibration with calibration plate has yet been carried out.

**The calibration affects the following Alignment methods:**

| Alignment | Result value |
|---|---|
| Contour matching | Position Coordinates |
| Pattern matching | Position Coordinates |
| Edge detector | Position coordinates*, distance |

**The calibration affects the following detectors:**

| Detector | Result value |
|---|---|
| Contour | Position Coordinates |
| Contour 3D | Position Coordinates |
| Pattern matching | Position Coordinates |
| Caliper | Position coordinates*, distance |
| BLOB | Position coordinates*, width, height |

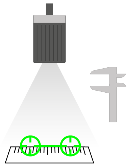
*not supported for: Hand-Eye calibration (Robotics) and Base-Eye calibration (Robotics)

### 9.1.6.2 Calibration methods "Measurement"

By default, distances in the image are given in pixels [px]. These can be converted into metric units such as millimeters by means of a calibration. With calibration methods "Measurement" the origin remains in the upper left corner of the field of view. Besides position coordinates, distances are also converted.

**NOTE:**
World coordinates are not absolute. The coordinate values refer to the principal point in the left, upper corner or the field of view.

#### 9.1.6.2.1 Calibration method "Scaling (Measurement)"



The calibration method "Scaling (Measurement)" is for relative determination of e.g. distances in world coordinates (e.g. mm). This is done using a simple ratio factor for both coordinate axes X and Y. The method is very simple to use, but the accuracy is limited because there is no correction for distortion.

| | |
|---|---|
| **Required objects:** | Working object |
| **Example:** | Determination of the distance between two objects in millimeters (medium accuracy) - if the object typically appears in the same region of the field of view. |
| **Calibration method provides:** | • Conversion from pixel to measurement unit |

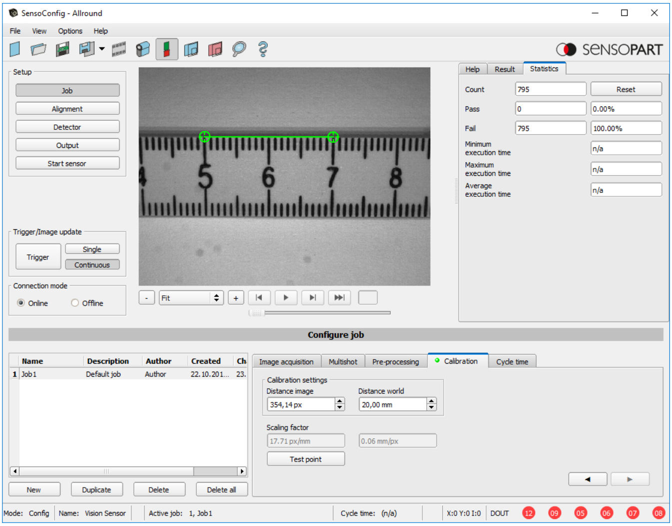


*Fig. 51: Calibration method "Scaling (Measurement)"*

### Parameters "Scaling (Measurement)"

| Parameter | Function |
|---|---|
| Distance image | Distance in the image in pixels [px], by graphical or value input. |
| Distance world | Corresponding distance in world by numerical input (in previously selected unit, e.g. mm) |
| Test point | A test point (graphical or value input) can be set in the image, the coordinates of which are displayed in world coordinates to control the scaling in the test point window. |
| Scaling factor | Scaling factors in px/mm or mm/px resulting from the above settings "Distance image" and "Distance world". |

### Calibration procedure "Scaling (Measurement)"

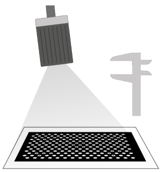
1. To parameterize, place an object of known extent (e.g. gauge) in the image.
2. Position the two graphical green crosshairs in the image on the points with the exact known distance.
   The size of the crosshair can be determined via the scroll wheel of the mouse.
   The distance in the image pixels between the two centers is displayed in the field "Distance image".
3. Now enter the known world distance in the field "Distance world" (e.g. in mm).

The scaling factor is calculated and displayed. From now on, positions and distances are displayed and transferred in world coordinates.

**Optimization of the calibration results**

- Align the sensor as vertically as possible to the field of view plane in order to avoid excessively different distortions in the two axes X and Y.
- Ideally, the calibration object should be located at the point in the field of view where the measurement will be made later.
- After calibration, the working distance (focus) and the position of the sensor to the Measurement plane must not be changed.

### 9.1.6.2.2 Calibration method "Calibration plate (Measurement)"



The calibration method "Calibration plate (Measurement)" is used for the relative determination of e.g. distances in world coordinates (e.g. mm). This is done via the image acquisition of a calibration plate

**Required objects:** Calibration plate

**Example:** Determination of the distance between two objects in millimetres (high accuracy) - the object appears at varying positions in the camera's field of view.

**Calibration method provides:**

- Conversion from pixel to measurement unit
- Distortion correction
- Correction of tilt between VISOR® and Measurement plane

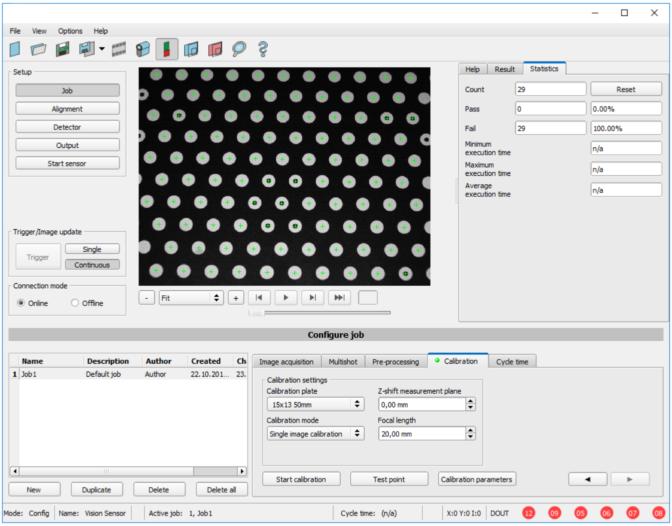
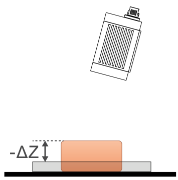
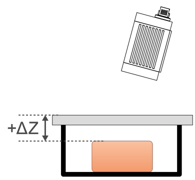



*Fig. 52: Calibration method "Calibration plate (Measurement)"*

### Parameters "Calibration plate (Measurement)"

| Parameter | Function |
|---|---|
| Calibration plate | Selection of the used calibration plate (size/type) (Additional information: Information on calibration plates) |

| Parameter | Function |
|---|---|
| Z-shift Measurement plane | The "Z-shift Measurement plane" parameter can be used to move the measuring plane along the Z axis (perpendicular to the plane) in order to obtain more accurate results, if necessary.<br>For Z=0, the calibration and the measurement plane are identical.<br>For Z ≠ 0, the calibration plane is shifted relative to the measurement plane.<br>The planes are always parallel. The sign of the shift results from the Z direction of the right-handed calibration coordinate system (thumb = X, index finger = Y, middle finger = Z).<br><br>*Fig. 53: "Z-shift Measurement plane" negative*<br><br>*Fig. 54: "Z-shift Measurement plane" positive*<br>**NOTE:**<br>The depth of focus of the sensor must cover the calibration plane and the measurement plane! |
| Calibration mode | Selection Single image calibration or Multi-image calibration (see Calibration procedure) |
| Focal length | Focal length of the lens<br>• With integrated lens: Value is entered automatically according to the internally installed lens.<br>• For C-Mount-variant: Read and enter the value from the lens used. |
| Start calibration | Calibration is started. All visible points of the calibration plate are determined, all detected are marked, and calibration is calculated. |
| Test point | A test point (graphical or value input) can be set in the image, the coordinates of which are displayed in world coordinates in the test point window to check the calibration or as a plausibility test of the scaling. |
| Calibration parameters | In the "Calibration parameters" window, parameters determined from the calibration are displayed. These can be used to find errors and optimize the calibration. Additional information: Calibration parameters |

#### Calibration procedure "Calibration plate (Measurement)"

The sensor can be mounted in any orientation (pose) in relation to the Measurement plane (whereby an orientation that is as perpendicular as possible to the Measurement plane requires less correction and is therefore advantageous). The scaling, in X and Y, the tilt of the sensor with respect to the field of view, and the lens distortion (depending on the selected calibration method) are corrected.

The image sharpness and brightness must first be set and the desired unit (to the right of the calibration method selection) must be selected. Two calibration modes are available: Single image calibration and Multi-image calibration.

#### Calibration mode "Single image calibration"

1. Set "Z-shift Measurement plane" between the calibration plate and Measurement plane.
2. Place the calibration plate in the field of view (Additional information: "Information on calibration plates").
3. Select the appropriate calibration plate (size and type) in the "Calibration plate" selection box.
4. Start calibration via "Start calibration".
   All visible points of the calibration plate are determined, all detected are marked, and calibration is calculated.
5. Check calibration in the "Calibration parameters" window if necessary.

#### Calibration mode "Multi-image calibration" (increased accuracy)

1. Set "Z-shift Measurement plane" between the calibration plate and Measurement plane.
2. Place the calibration plate in the field of view (Additional information: "Information on calibration plates").
3. Select the appropriate calibration plate (size and type) in the "Calibration plate" selection box.
4. Set calibration mode to "Multi-image calibration".
5. Start calibration via "Start calibration".
6. Verschiedene Bilder von der Kalibrierplatte aufnehmen (empfohlen: > 5 Bilder).
   **Note:** Das **erste** Bild bestimmt die Measurement plane (diese kann später ggf. angepasst werden). For the further images, the calibration plate should be tilted to the Measurement plane and shifted in Z-direction for best results.
7. Check calibration in the "Calibration parameters" window if necessary.

**NOTE:**
World coordinates are not absolute. The coordinate values refer to the principal point in the left, upper corner or the field of view.

For tips showing how to best use the calibration plate / boundary conditions, please refer to: Information on calibration plates

### 9.1.6.3 Calibration methods "Robotics"

The robot calibrations first convert pixels into metric units (e.g. mm) and correct distortion and tilted viewing angles. In addition, the camera coordinate system is projected onto that of the robot, so that the robot can now move directly in its coordinate system with the position data supplied by the sensor and can grip a part, for example.

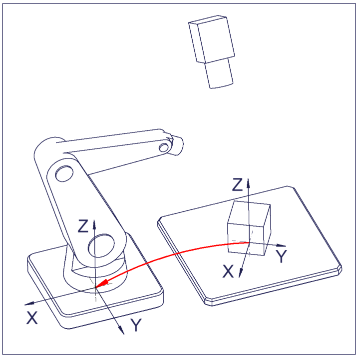
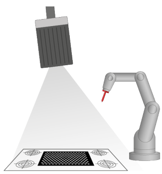



*Fig. 55: Position of the part to be gripped directly in the robot coordinate system*

#### 9.1.6.3.1 Calibration method "Calibration plate (Robotics)"



The calibration method "Calibration plate (Robotics)" is used to determine absolute positions in the robot coordinate system. This is done by acquiring one or more images of the calibration plate and teaching four fiducials.

**Required objects:** Calibration plate "Crosshair" (calibration plate with fiducials)

**Example:** Picking parts from a feeder with a stationary mounted VISOR®.

**Calibration method provides:**

- Conversion from pixel to measurement unit
- Distortion correction
- Correction of tilt between VISOR® and Measurement plane
- Output of world coordinates in robot coordinate system

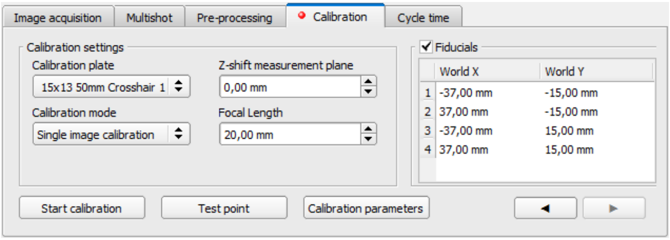
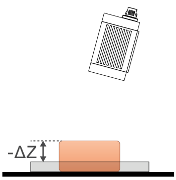
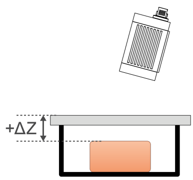



*Fig. 56: Calibration method "Calibration plate (Robotics)"*

**Parameters "Calibration plate (Robotics)"**

| Parameter | Function |
|---|---|
| Calibration plate | Selection of the used calibration plate (size/ type) (Additional information: "Information on calibration plates") |
| Z-shift Measurement plane | The "Z-shift Measurement plane" parameter can be used to move the measuring plane along the Z axis (perpendicular to the plane) in order to obtain more accurate results, if necessary.<br>For Z=0, the calibration and the measurement plane are identical.<br>For Z ≠ 0, the calibration plane is shifted relative to the measurement plane.<br>The planes are always parallel. The sign of the shift results from the Z direction of the right-handed calibration coordinate system (thumb = X, index finger = Y, middle finger = Z).<br><br>*Fig. 57: "Z-shift Measurement plane" negative*<br><br>*Fig. 58: "Z-shift Measurement plane" positive*<br>**NOTE:**<br>The depth of focus of the sensor must cover the calibration plane and the measurement plane! |
| Calibration mode | Selection Single image calibration or Multi-image calibration (see Calibration procedure) |

| Parameter | Function |
| --- | --- |
| Focal length | Focal length of the lens<br>• With integrated lens: Value is entered automatically according to the internally installed lens.<br>• For C-Mount-variant: Read and enter the value from the lens used. |
| Fiducials<br>- World X<br>- World Y | Coordinate values in world in selected unit (e.g. mm), by directly entering values in the list of fiducials.<br>In the case of Pick & Place, for example, these values are the X/Y coordinate values that can be read off from the robot controller when the calibration part is placed or the corresponding point is approached and transferred to the list of fiducials. |
| Start calibration | Calibration is started. All visible points of the calibration plate are determined, all detected are marked, and calibration is calculated. |
| Test point | A test point (graphical or value input) can be set in the image, the coordinates of which are displayed in world coordinates in the test point window to check the calibration or as a plausibility test of the scaling. |
| Calibration parameters | In the "Calibration parameters" window, parameters determined from the calibration are displayed. These can be used to find errors and optimize the calibration. Additional information: Calibration parameters |

#### Calibration procedure "Calibration plate (Robotics)"

The image sharpness and brightness must first be set and the desired unit (to the right of the calibration method selection) must be selected. Two calibration modes are available: Single image calibration and Multi-image calibration.

#### Calibration mode "Single image calibration"

1. Set "Z-shift Measurement plane" between the calibration plate and Measurement plane.
2. Place the calibration plate in the field of view so that it covers as much as possible (Additional information: "Information on calibration plates").
3. Select the appropriate calibration plate (size and type) in the "Calibration plate" selection box.
4. Start calibration via "Start calibration".
   All visible points of the calibration plate are determined, all detected are marked, and calibration is calculated.
5. Check "Fiducials" if not yet active.
6. For fiducial 1, select the first line in list box "Fiducials".
7. Approach the first fiducial with the robot.

8. In the field "World X" and "World Y", enter the corresponding known world coordinate values (for robots: the values from the robot controller).
9. For fiducials 2, 3 and 4: Select the next line in the list box "Fiducials". Move to the next fiducial and enter the corresponding values (see steps 6-8).
10. Check calibration in the "Calibration parameters" window if necessary.

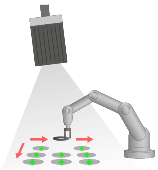
**Calibration mode "Multi-image calibration" (increased accuracy)**

1. Set "Z-shift Measurement plane" between the calibration plate and Measurement plane.
2. Place the calibration plate in the field of view (Additional information: "Information on calibration plates").
3. Select the appropriate calibration plate (size and type) in the "Calibration plate" selection box.
4. Set calibration mode to "Multi-image calibration".
5. Verschiedene Bilder von der Kalibrierplatte aufnehmen (empfohlen: > 5 Bilder).
   **Note:** Das **erste** Bild bestimmt die Measurement plane (diese kann später ggf. angepasst werden). For the further images, the calibration plate should be tilted to the Measurement plane and shifted in Z-direction for best results.
6. Start calibration via "Start calibration".
7. Check "Fiducials" if not yet active.
8. For fiducial 1, select the first line in list box "Fiducials".
9. Approach the first fiducial with the robot.
10. In the field "World X" and "World Y", enter the corresponding known world coordinate values (for robots: the values from the robot controller).
11. For fiducials 2, 3 and 4: Select the next line in the list box "Fiducials". Move to the next fiducial and enter the corresponding values (see steps 8-10).
12. Check calibration in the "Calibration parameters" window if necessary.

See also: Automated procedure via interface commands (Calibration plate (Robotics) - Special case: Separate robot working area and field of view and Automated calibration: Calibration plate (Robotics))

For tips showing how to best use the calibration plate / boundary conditions, please refer to: Information on calibration plates

### 9.1.6.3.2 Calibration method "Point pair list (Robotics)"



The calibration method "Point pair list (Robotics)" is a calibration at the working object - therefore no calibration plate is needed. After calibration of the sensor, the position of the part to be gripped by the robot is directly available in the absolute coordinate system of the robot.

**Required objects:** Working object

**Example:** Determine absolute position (and orientation) of objects in world coordinates (e.g. robot coordinate system).

**Calibration method provides:**

- Conversion from pixel to measurement unit
- Distortion correction
- Correction of tilt between VISOR® and Measurement plane
- Output of world coordinates in robot coordinate system

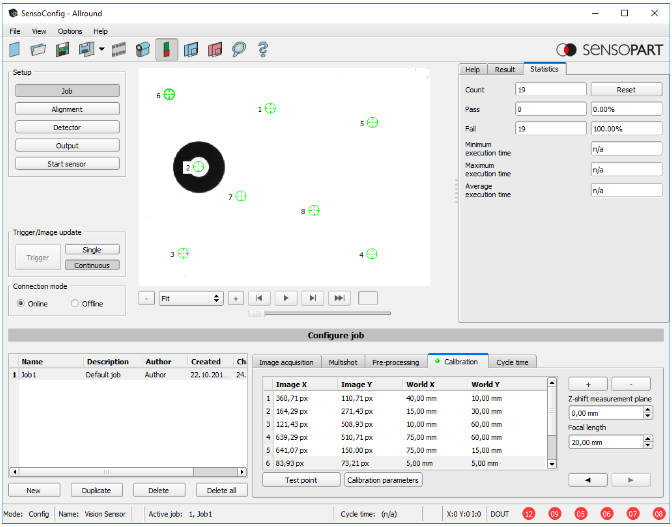
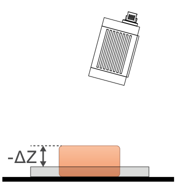
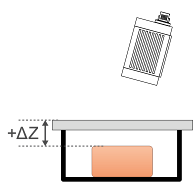



*Fig. 59: Calibration method "Point pair list (Robotics)"*

**Parameters "Point pair list (Robotics)"**

| **Parameter** | **Function** |
|---|---|
| • **Image X**<br>• **Image Y**<br>Values in point list | Coordinate values in pixels [px] in the image, via the exact graphical positioning of the crosshair on the center point of the calibration part, which is precisely placed in world coordinates.<br>Or, via "Snap function": Right mouse click anywhere within the symmetrical calibration object. This way, the exact position of the center can be determined automatically.<br>**NOTE:**<br>The snap function is not available for Color variants. |
| • **World X**<br>• **World Y**<br>Values in point list | Coordinate values in selected unit (e.g. mm), by directly entering values in point pair list.<br>In the case of Pick & Place, for example, these values are the X/Y coordinate values that can be read by the robot controller when the calibration part is placed or the corresponding point is approached and transferred to the reference mark list. |
| + / - | Add or delete one line / list point. The selected line is deleted. |
| Z-shift Measurement plane | The "Z-shift Measurement plane" parameter can be used to move the measuring plane along the Z axis (perpendicular to the plane) in order to obtain more accurate results, if necessary.<br>For Z=0, the calibration and the measurement plane are identical.<br>For Z ≠ 0, the calibration plane is shifted relative to the measurement plane.<br>The planes are always parallel. The sign of the shift results from the Z direction of the right-handed calibration coordinate system (thumb = X, index finger = Y, middle finger = Z).<br><br><br>*Fig. 60: "Z-shift Measurement plane" negative*<br><br><br>*Fig. 61: "Z-shift Measurement plane" positive*<br>**NOTE:**<br>The depth of focus of the sensor must cover the calibration plane and the measurement plane! |

| Parameter | Function |
|---|---|
| Focal length | Focal length of the lens<br>• With integrated lens: Value is entered automatically according to the internally installed lens.<br>• For C-Mount-variant: Read and enter the value from the lens used. |
| Test point | A test point (graphical or value input) can be set in the image, the coordinates of which are displayed in world coordinates in the test point window to check the calibration or as a plausibility test of the scaling. |
| Calibration parameters | In the "Calibration parameters" window, parameters determined from the calibration are displayed. These can be used to find errors and optimize the calibration. Additional information: Calibration parameters |

**Calibration procedure "Point pair list (Robotics)"**

**NOTE:**

- The accuracy depends mainly on the high quality as well as on the sufficient number of calibration points (at least 6 points, recommended: ≥ 9 points).
- The accuracy can be optimized by a high precision in position determination and input of the individual points, e.g. if some points are displayed in yellow color.
- Preferably use flat, point-symmetrical calibration objects (e.g. washer), as this way the center of gravity is independent of the orientation. For calibration objects that are not point-symmetrical, ensure that the orientation is always the same when positioning.

The sensor can be mounted in any orientation (pose) with respect to the Measurement plane. An alignment as perpendicular as possible with the Measurement plane, however, requires less correction and is therefore more advantageous.

The image sharpness and brightness must first be set and the desired unit (to the right of the calibration method selection) must be selected.

1. Set "Z-shift Measurement plane".
2. Select Line 1 in list box "Point pair list".
3. Place the calibration object at an exactly known world coordinate in the field of view (e.g. with a robot).
4. Fadenkreuz (Nr. "n" zu entsprechender Zeile "n" in Punktpaarliste) exakt im Zentrum des Kalibrierobjektes grafisch positionieren. Zoom the image if necessary.
   Alternatively: Use "Snap Function", i.e. right click somewhere inside the calibration object. The center of gravity of the calibration object is determined automatically (not available for Color

variants).
The size of the crosshair can be determined via the scroll wheel of the mouse.
Result: Pixel values for image coordinates "Image X" and "Image Y" are automatically entered in Line "n".

5. Now enter the corresponding known world coordinate values in the "World X" and "World Y" fields (for example, for robots: the values from the robot controller).
6. Schritte 3-5 so lange wiederholen, bis die gewünschte Anzahl an Punktpaaren eingegeben wurde (min. 6 Punkte, empfohlen >10 Punkte). If necessary, create further lines with "+".
7. Check calibration in the "Calibration parameters" window if necessary.

See also: Automated process with interface commands (Automated calibration: Point pair list (Robotics))

### Meaning of the colors of the points in the image and in the point pair list

The entered points are displayed in the following colors to indicate the position quality, i.e. how well they correspond to the position determined by the adjustment calculation (can only be used effecitvely from a minimum number of 6 points).

| Color of the crosshairs | Status of the calibration | Meaning / Measures |
|---|---|---|
| Green | Calibration valid, points accurately positioned | No action required |
| Yellow | Calibration valid, point is not exactly positioned | Check pair of points of the point |
| Red | No valid calibration, assignment of world points / pixels deviates strongly from the model. | Check assignment |

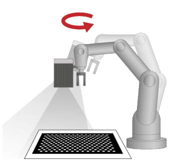
### Error

In the case of yellow points, a line is visible from the point center. It is a measure of the direction and magnitude of the error with respect to the achieved position accuracy of the point input in the real world.

If the errors are large here, the X and Y values or entire pairs of points in the image and world may be interchanged at one or more points.

Im Dialog "Calibration parameters" werden die **Abweichungswerte** / Fehler angezeigt: "Mittelwert", minimaler Fehler "Min." und maximaler Fehler "Max.". The exact position input of the existing points may be optimized with these values.

### 9.1.6.3.3 Calibration method "Hand-Eye calibration (Robotics)"



The "Hand-Eye calibration (Robotics)" calibration method is used to determine the reference between Tool Coordinate System (TCP) and Camera coordinate system (position and orientation) when the VISOR® is attached to the gripper.

**Required objects:** Calibration plate

**Example:** Screwing on components with multiple positions on VISOR® attached to robot arm.

**Calibration method provides:**

- Conversion from pixel to measurement unit
- Distortion correction
- Correction of tilt between VISOR® and Measurement plane
- Output of world coordinates in robot coordinate system, independent of image acquisition position
- References (see figure References: "Robotics" Calibration methods (Page 129))
  - TCP_CF (Tool Coordinate System (TCP) - Camera coordinate system, corresponds to Hand-Eye)
  - CF_CPF (Camera coordinate system - Calibration Plate Coordinate System)
  - CPF_MF (Calibration Plate Coordinate System - Measuring coordinate system)

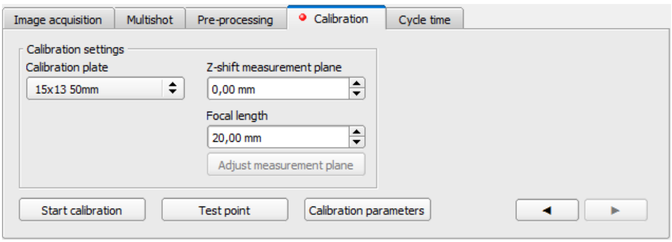
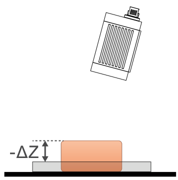
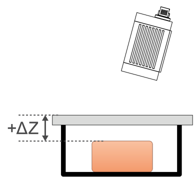



*Fig. 62: Calibration method "Hand-Eye calibration (Robotics)"*

#### Parameter "Hand-Eye calibration (Robotics)"

| Parameter | Function |
|---|---|
| Calibration plate | Selection of the used calibration plate (size/ type) (Additional information: "Information on calibration plates") |

| Parameter | Function |
|---|---|
| Z-shift Measurement plane | The "Z-shift Measurement plane" parameter can be used to move the measuring plane along the Z axis (perpendicular to the plane) in order to obtain more accurate results, if necessary.<br>For Z=0, the calibration and the measurement plane are identical.<br>For Z ≠ 0, the calibration plane is shifted relative to the measurement plane.<br>The planes are always parallel. The sign of the shift results from the Z direction of the right-handed calibration coordinate system (thumb = X, index finger = Y, middle finger = Z).<br><br><br>*Fig. 63: "Z-shift Measurement plane" negative*<br><br><br>*Fig. 64: "Z-shift Measurement plane" positive*<br>**NOTE:**<br>The depth of focus of the sensor must cover the calibration plane and the measurement plane! |
| Focal length | Focal length of the lens<br>• With integrated lens: Value is entered automatically according to the internally installed lens.<br>• For C-Mount-variant: Read and enter the value from the lens used. |
| Adjust Measurement plane | This allows the Measurement plane to be changed subsequently. This is necessary, for example, if the orientation of the vision sensor or the distance to the Measurement plane have changed, typically if the calibration position and working position are different.<br>The tool position (TCP) is used for the calculation. If the current position differs from the stored position, it can be entered in the dialog. |
| Start calibration | Calibration process is initiated: Dialog "Base-Eye calibration (Robotics)" opens (see Dialog Hand-Eye calibration (Robotics) (Page 117)). |

| Parameter | Function |
|---|---|
| Test point | A test point (graphical or value input) can be set in the image, the coordinates of which are displayed in world coordinates in the test point window to check the calibration or as a plausibility test of the scaling. |
| Calibration para-meters | In the "Calibration parameters" window, parameters determined from the calibration are displayed. These can be used to find errors and optimize the calibration. Additional information: Calibration parameters |

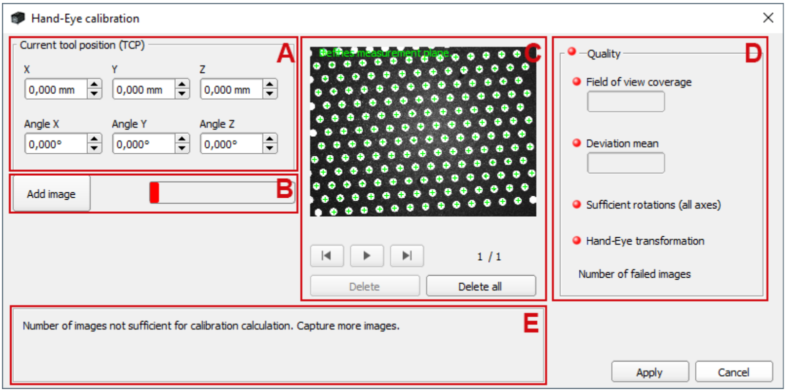



*Fig. 65: Dialog Hand-Eye calibration (Robotics)*

**A**: Input area for current tool position (TCP)

**B**: Adding images for calibration; status display

**C**: Display area of the recorded images

**D**: Evaluation of the currently calculated calibration

**E**: Information area for notes

### General notes on Hand-Eye calibration (Robotics)

- The position of the calibration plate must not change during calibration.
- Make sure that there is enough space around your robot tool (TCP) to perform 10 poses as described in the position selection.
- After calibration, the working distance (focus) must not be changed.
- The calibration is only valid for the tool (TCP) and the coordinate system that are active during the calibration.
- The accuracy of the calibration can often be further increased by adding more images.

- In order to have sufficient freedom of movement for the robot, the distance of flange to VISOR® should be significantly smaller (~< 25%) than the length of the robot arm.

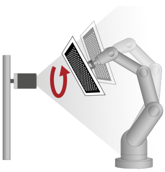
**Notes on position selection for a Hand-Eye calibration (Robotics)**

- Use your Tool Coordinate System (TCP) to move the robot
- Tilt your tool (TCP) strongly around 2 axes between each pose you approach, ideally ~ 60° (min. 20°).
- Then perform the necessary translation to bring the calibration plate back into the field of view of the camera.
- In this procedure, try to achieve the greatest possible variation of the tilts between all poses used for calibration.

**Procedure of the calibration method "Hand-Eye calibration (Robotics)"**

1. Select the correct size and type of calibration plate.
2. Set "Z-shift Measurement plane".
3. Position the calibration plate in the field of view or move the camera (attached to the robot arm) over the calibration plate.
4. Set image acquisition parameters (Shutter speed, Working distance of the VISOR®). The "Parameters" Working distance must not be changed anymore from now on.
5. Click "Start calibration".
6. Read the current position of the robot from the controller and transfer values to the SensoConfig dialog.
7. Click "Add image".
8. Change robot position and perform steps 6 and 7 for at least 6 (recommended: 10) robot positions.
   Please observe the notes on position selection!
9. Click "Apply".
10. Check calibration in the "Calibration parameters" window if necessary.

### 9.1.6.3.4 Calibration method "Base-Eye calibration (Robotics)"



The "Base-Eye calibration (Robotics)" calibration method is used to determine the reference from Camera coordinate system to the robot base (position and orientation) when the vision VISOR® sensor is mounted stationary.

**Required objects:** Calibration plate

**Example:** Position correction of gripped component in front of a stationary mounted VISOR®.

**Calibration method provides:**

- Conversion from pixel to measurement unit
- Distortion correction
- Correction of tilt between VISOR® and Measurement plane
- Output of world coordinates in robot coordinate system
- References
  - RF_CF (Robot Coordinate System - Camera coordinate system, corresponds to Base-Eye )
  - CF_CPF (Camera coordinate system - Calibration Plate Coordinate System)
  - CPF_MF (Calibration Plate Coordinate System zu Measuring coordinate system)

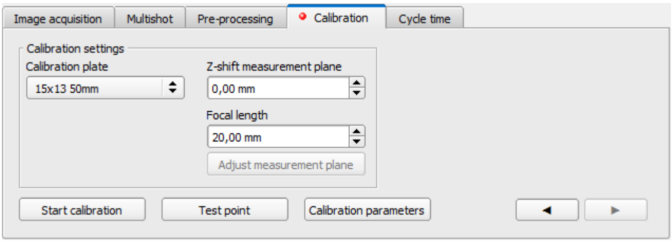
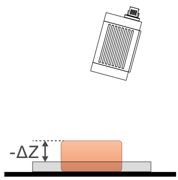
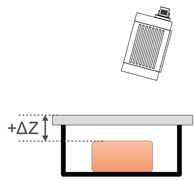



*Fig. 66: Calibration method "Base-Eye calibration (Robotics)"*

### Parameter "Base-Eye calibration (Robotics)"

| Parameter | Function |
|---|---|
| Calibration plate | Selection of the used calibration plate (size/ type) (Additional information: "Information on calibration plates (Page 122)") |

| Parameter | Function |
|---|---|
| Z-shift Measurement plane | The "Z-shift Measurement plane" parameter can be used to move the measuring plane along the Z axis (perpendicular to the plane) in order to obtain more accurate results, if necessary.<br>For Z=0, the calibration and the measurement plane are identical.<br>For Z ≠ 0, the calibration plane is shifted relative to the measurement plane.<br>The planes are always parallel. The sign of the shift results from the Z direction of the right-handed calibration coordinate system (thumb = X, index finger = Y, middle finger = Z).<br> <br>*Fig. 67: "Z-shift Measurement plane" negative*<br> <br>*Fig. 68: "Z-shift Measurement plane" positive*<br>**NOTE:**<br>The depth of focus of the sensor must cover the calibration plane and the measurement plane! |
| Focal length | Focal length of the lens<br>• With integrated lens: Value is entered automatically according to the internally installed lens.<br>• For C-Mount-variant: Read and enter the value from the lens used. |
| Adjust Measurement plane | This allows the Measurement plane to be changed subsequently. This is necessary, for example, if the orientation of the vision sensor or the distance to the Measurement plane have changed, typically if the calibration position and working position are different.<br>The tool position (TCP) is used for the calculation. If the current position differs from the stored position, it can be entered in the dialog. |
| Start calibration | Calibration process is initiated: Dialog "Base-Eye calibration (Robotics)" opens (see Dialog Base-Eye calibration (Robotics) (Page 121)). |

| Parameter | Function |
|---|---|
| Test point | A test point (graphical or value input) can be set in the image, the coordinates of which are displayed in world coordinates in the test point window to check the calibration or as a plausibility test of the scaling. |
| Calibration para-meters | In the "Calibration parameters" window, parameters determined from the calibration are displayed. These can be used to find errors and optimize the calibration. Additional information: Calibration parameters |

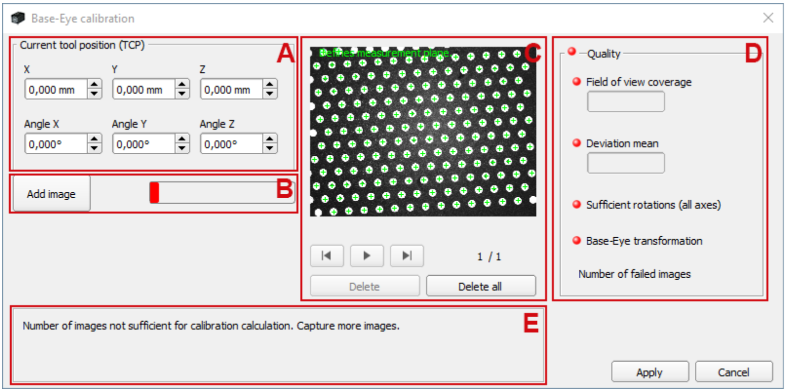



*Fig. 69: Dialog Base-Eye calibration (Robotics)*

**A**: Input area for current tool position (TCP)

**B**: Adding images for calibration; status display

**C**: Display area of the recorded images

**D**: Evaluation of the currently calculated calibration

**E**: Information area for notes

### General notes on Base-Eye calibration (Robotics)

- The position of the calibration plate to the gripper must not change during calibration (calibration plate must not slip).
- Make sure that there is enough space around your robot tool (TCP) to perform 10 poses as described in the position selection.
- After calibration, the working distance (focus) must not be changed.
- The calibration is only valid for the coordinate system active during calibration.
- The accuracy of the calibration can often be further increased by adding more images.

**Notes on position selection for a Base-Eye calibration (Robotics)**

- Use your Tool Coordinate System (TCP), to move the robot
- Tilt your tool (TCP) strongly around 2 axes between each pose you approach, ideally ~ 60° (min. 20°).
- Then perform the necessary translation to bring the calibration plate back into the field of view of the camera.
- In this procedure, try to achieve the greatest possible variation of the tilts between all poses used for calibration.

**Procedure of the calibration method "Base-Eye calibration (Robotics)"**

1. Select the correct size and type of calibration plate.
2. Set "Z-shift Measurement plane".
3. Attach the calibration plate to the tool (TCP) and move it into the field of view with the robot arm.
4. Set image acquisition parameters (Shutter speed, Working distance of the VISOR®). The "Parameters" Working distance must not be changed anymore from now on.
5. Click "Start calibration".
6. Read the current position of the robot from the controller and transfer values to the SensoConfig dialog.
7. Click "Add image".
8. Change robot position and perform steps 6 and 7 for at least 6 (recommended: 10) robot positions.
   Please observe the notes on position selection!
9. Click "Apply".
10. Check calibration in the "Calibration parameters" window if necessary.

### 9.1.6.4 Information on calibration plates

When using calibration plates, the scaling in X and Y, the tilt of the sensor relative to the plane of view and the lens distortion (depending on the calibration method selected) are corrected.

Calibration plates can be ordered via the website or they can be printed or applied to paper or any other flat medium. In the installation directory \SensoPart\VISOR Vision Sensor\Documentation\Calibrationplates, the calibration plates available for this purpose can be found as PDF files. When printing, use the "Actual Size" setting, and ensure not to scale the print. The edge length / label of the plate must match the name of the plate when selected in the software.

### Calibration plates without fiducials

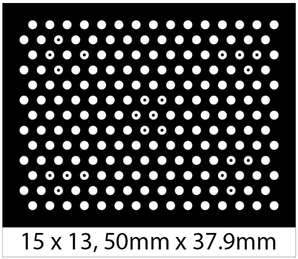



*Fig. 70: Calibration plate without fiducials*

Typically used for calibration methods: Calibration plate (Measurement), Hand-Eye calibration (Robotics), Base-Eye calibration (Robotics)

### Calibration plates with fiducials

Typically used for calibration method: Calibration plate (Robotics)

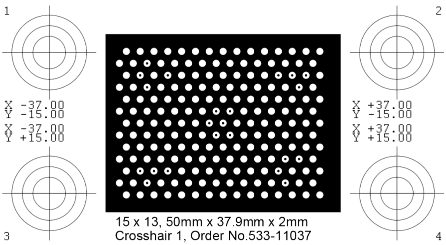



*Fig. 71: Calibration plate with fiducials*

### Advice on optimized use of the calibration plate / boundary conditions

- The calibration plate must be clean and level.
- The illumination of the panel should be homogeneous throughout the field of view and not overexposed. The light areas should have a gray value of at least 100 and all below the value 255. The contrast between light and dark areas should be at least 100 gray levels. This means that the image may not be underexposed or overexposed.
- The calibration pattern should ideally cover the entire field of vision of the VISOR® vision sensor. This can be ensured either by a large calibration plate or by a Multi-image calibration.
- To perform a calibration, at least one search pattern must be found.
- For small calibration patterns, it may be necessary to use two search patterns.

- After calibration is complete, the working distance (focus) must not be changed anymore. If the position of the camera to the Measurement plane changes, the Measurement plane must be taught-in again.

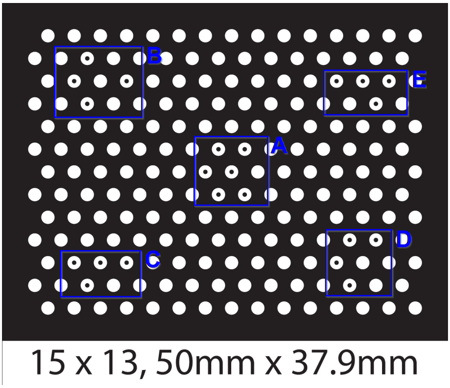



*Fig. 72: Calibration plate, blue = search pattern*

### 9.1.6.5 Calibration parameters

In the "Calibration parameters" window, parameters determined from the calibration are displayed. These can be used to find errors and optimize the calibration. The displayed parameters are read-only parameters.

All parameters are described below - however, not all parameters are available for every calibration method.

### "Overview" tab

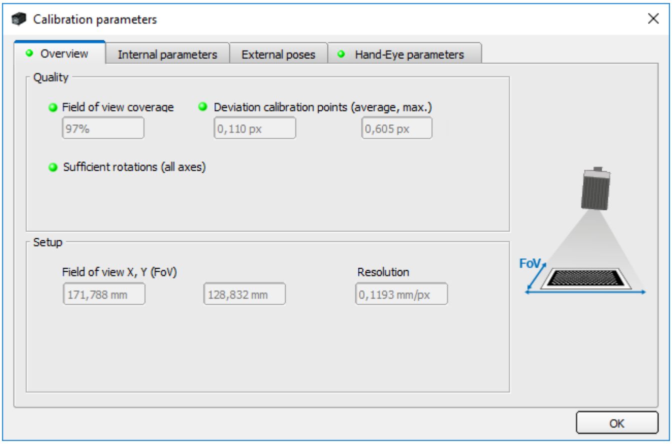



*Fig. 73: Calibration parameters, "Overview" tab*

| Parameter | Function |
|---|---|
| **Quality** | |
| Field of view coverage | Indicates in which part of the field of view calibration objects (e.g. calibration plate) were detected. In order to obtain accurate results, the highest possible value should be achieved here (100%). |
| Deviation Calibration points (Mean, Max.) | Deviation of the calibration points between detected and expected position in pixels. |
| Deviation fiducials (Average, Max.) | Deviation of the reference marks between specified and expected position in pixels. |
| Sufficient rotations (all axes) | Indicator for a good calibration.<br>Green: exact calibration<br>Yellow: Image width cannot be determined exactly, position information becomes less accurate.<br>LED turns green when tilt difference between some of the recorded calibration plate images is at least 20 degrees. |
| **Setup** | |
| Field of view X, Y (FoV) | Size of the field of view detected by the VISOR® |
| Resolution | Resolution from customer unit to pixels (customer unit / px) in the determined field of view |

### "Internal Parameters" tab

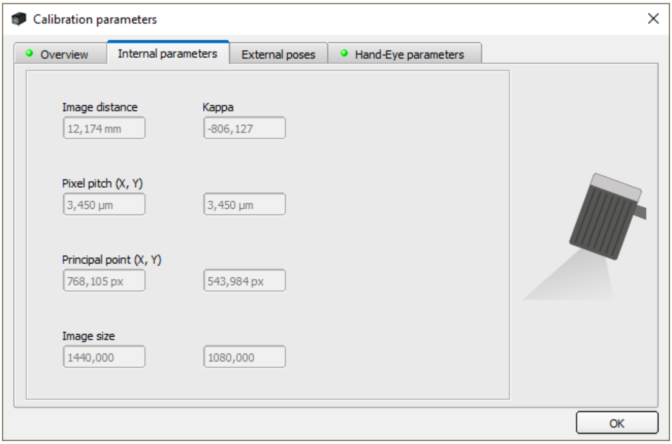



*Fig. 74: Calibration parameters, "Internal parameters" tab*

| Parameter | Function |
|---|---|
| Image distance | Image distance determined from the calibration |
| Kappa | Calculated kappa value of the lens, display in micro kappa ( x 10E-6) |
| Pixel pitch (X, Y) | Calculated grid / axis distance from pixel to pixel on the sensor. Reducing the resolution in the "Image acquisition" tab affects this parameter. |
| Coordinate origin (X, Y) | Puncture point of the optical axis through Measurement plane in the center of the sensor chip, opposite ideal center, in relation to upper left corner in pixels. |
| Image size | Image size in pixels |

### "External poses" tab

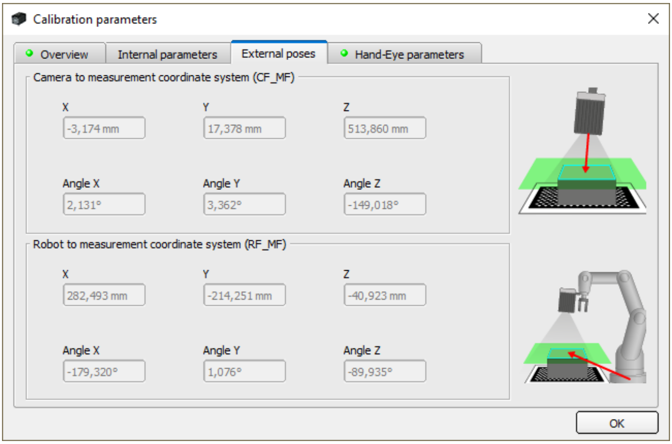



*Fig. 75: Calibration parameters, "External poses" tab*

Note: The references in the following tabs are given in the rotation order that was selected when the calibration method was selected (Yaw-Pitch-Roll / Roll-Pitch-Yaw). Regardless of this, the order in which the values are specified is always (X, Y, Z, Angle X, Angle Y, Angle Z).

| Parameter | Function |
|---|---|
| Camera- to Measuring coordinate system (CF_MF) | Describes the 3D reference from Camera coordinate system (CF) to Measuring coordinate system (MF, determined by the calibration). |
| Robot- to Measuring coordinate system (RF_MF) | Describes the 3D reference from the Robot Coordinate System (RF) that is active during calibration to the Measuring coordinate system (MF, determined by the calibration). |
| X<br>Y<br>Z | Translation values of the considered reference |
| Angle X<br>Angle Y<br>Angle Z | Rotation values (angles) of the considered reference |

### "Hand Eye Parameters" and "Base Eye Parameters" tab

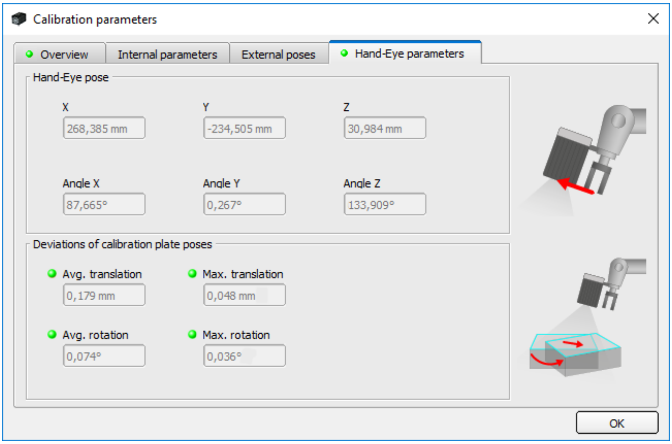



*Fig. 76: Calibration parameters, "Hand-Eye parameters" tab*

| Parameter | Function |
|---|---|
| Hand eye reference (TCP_CF) or base eye reference (RF_CF) | |
| X<br>Y<br>Z | Translation values of the Hand Eye or Base Eye reference |
| Angle X<br>Angle Y<br>Angle Z | Rotation values of the Hand Eye or Base Eye reference |
| **Deviation Calibration plate poses** | |
| Average translation | Average translation deviation, Root Mean Square Error (RMSE), calculated from the expected to the measured translations of the calibration plate poses<br>(in user unit * 1000) |
| Maximum translation | Maximum translation deviation, calculated from the expected to the measured translations of the calibration plate poses<br>(in user unit * 1000) |
| Average rotation | Mean rotation deviation, Root Mean Square Error (RMSE), calculated from the expected to the measured rotations of the calibration plate poses<br>(in degrees * 1000) |

| Parameter | Function |
|---|---|
| Maximum rotation | Maximum rotation deviation, calculated from the expected to the measured rotations of the calibration plate poses (in degrees * 1000) |

Interpretation of the deviation values: These deviations can be used as an indication of the approach accuracy of the robot, ...

- ...if the image acquisition positions during object acquisition are within the robot's movement range that was covered during calibration. If the image acquisition positions vary less, the deviations are smaller.
- ...if no result offset is used (if the approach point lies on the localization feature). The further the approach point is from the result point, the greater the deviations.
- ...if the Measurement plane corresponds to the calibration plane. The further away the Measurement plane is from the calibration plane, the greater the deviations.

### 9.1.6.6 Coordinate systems and transformations

The following figure shows the notations of the references used in the context of calibration.

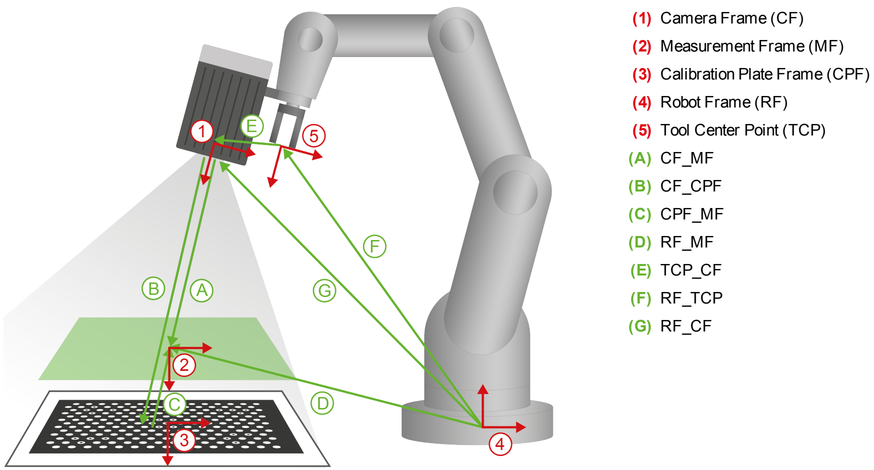



*Fig. 77: References: "Robotics" Calibration methods*

| Coordinate system | Description |
|---|---|
| (1) Camera Frame (CF) | Camera coordinate system<br>Origin of Camera coordinate system lies inside the camera housing. |
| (2) Measurement Frame (MF) | Measuring coordinate system<br>Measuring coordinate system is shifted parallel to Calibration Plate Coordinate System by the "Z-shift Measurement plane" parameter. |
| (3) Calibration Plate Frame (CPF) | Calibration Plate Coordinate System<br>Origin of Calibration Plate Coordinate System lies in the center of the calibration plate. |
| (4) Robot Frame (RF) | Robot Coordinate System<br>The Robot Coordinate System is the coordinate system that is active during calibration or the base (depending on the manufacturer). |
| (5) Tool Center Point (TCP) | Tool Coordinate System (TCP) |

| Reference | Description |
|---|---|
| (A) CF_MF | Reference: Camera coordinate system - Measuring coordinate system |
| (B) CF_CPF | Reference: Camera coordinate system - Calibration Plate Coordinate System |
| (C) CPF_MF | Reference: Calibration Plate Coordinate System - Measuring coordinate system |
| (D) RF_MF | Reference: Robot Coordinate System - Measuring coordinate system |
| (E) TCP_CF | Reference: Tool Coordinate System (TCP) - Camera coordinate system (corresponds to Hand-Eye reference). |
| (F) RF_TCP | Reference: Robot Coordinate System - Tool Coordinate System (TCP) This reference is visualized by most robots as "current position". |
| (G) RF_CF | Reference: Robot Coordinate System - Camera coordinate system |

### 9.1.6.7 Calibration via telegrams

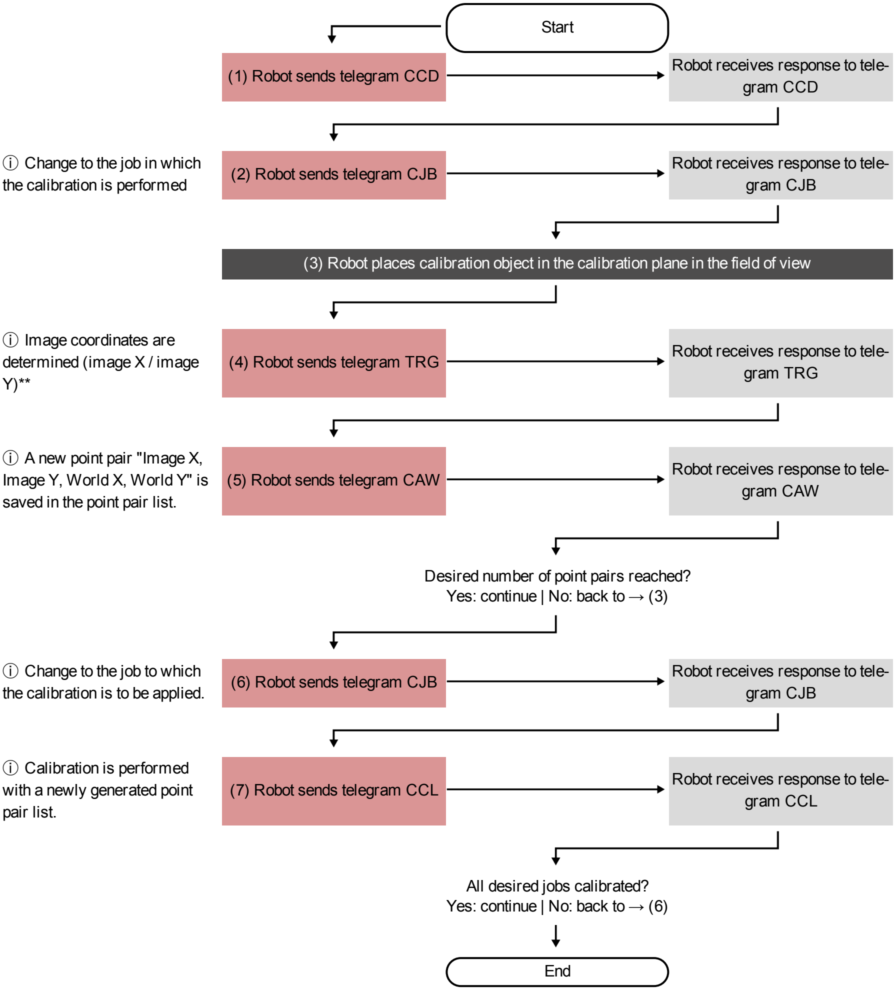
Various interface telegrams are available for the "Robotics" calibration methods (see also Communications manual, Chapter "Overview of telegrams" )

The telegrams can be used for recalibration during process drift or with changed mounting situation. For example, they can be executed automatically directly from the robot controller.

### Meaning of the colors

| Robot sends | Robot receives | Robot action | User action |
| --- | --- | --- | --- |

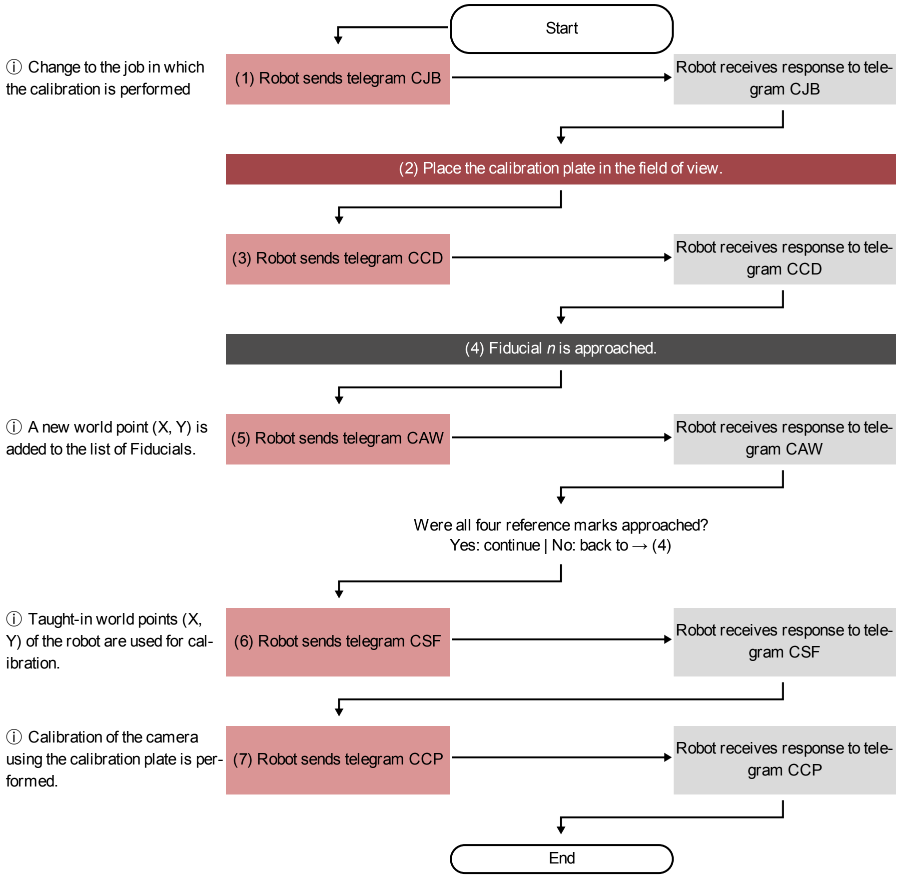
### 9.1.6.7.1 Automated calibration: Point pair list (Robotics)




** To add image coordinates, the X value must be entered in line 1 and the Y value must be entered in line 2 under Output / Telegram / Payload. Furthermore, the overall job result must be positive.

### 9.1.6.7.2 Automated calibration: Calibration plate (Robotics)

#### Calibration plate (Robotics) - Standard process Single image calibration



## Calibration plate (Robotics) - Standard process Multi-image calibration

Start

ⓘ Change to the job in which the calibration is performed

(1) Robot sends telegram CJB → Robot receives response to telegram CJB

(2) Place the calibration plate in the field of view.

(3) Robot sends telegram CCD → Robot receives response to telegram CCD

(4) Fiducial *n* is approached.

ⓘ A new world point (X, Y) is added to the list of Fiducials.

(5) Robot sends telegram CAW → Robot receives response to telegram CAW

Were all four reference marks approached?
Yes: continue | No: back to → (4)

ⓘ Taught-in world points (X, Y) of the robot are used for calibration.

(6) Robot sends telegram CSF → Robot receives response to telegram CSF

ⓘ An image of the calibration plate is added

(7) Robot sends telegram CAI → Robot receives response to telegram CAI

Desired number of calibration images taken? (Recommendation: >6)
Yes: continue | No: back to → (7)

ⓘ Calibration of the camera using the calibration plate is performed.

(8) Robot sends telegram CMP → Robot receives response to telegram CMP

End

## Calibration plate (Robotics) - Special case: Separate robot working area and field of view

Start

ⓘ Change to the job in which the calibration is performed

(1) Robot sends telegram CJB → Robot receives response to telegram CJB

(2) Place the calibration plate in the field of view.

ⓘ Image coordinates are determined (image X / image Y)**

(3) Robot sends telegram CCP → Robot receives response to telegram CCP

(4) Plate is moved into the working area of the robot.

(5) Robot sends telegram CCD → Robot receives response to telegram CCD

(6) Fiducial *n* is approached.

ⓘ A new world point (X, Y) is added to the list of Fiducials.

(7) Robot sends telegram CAW → Robot receives response to telegram CAW

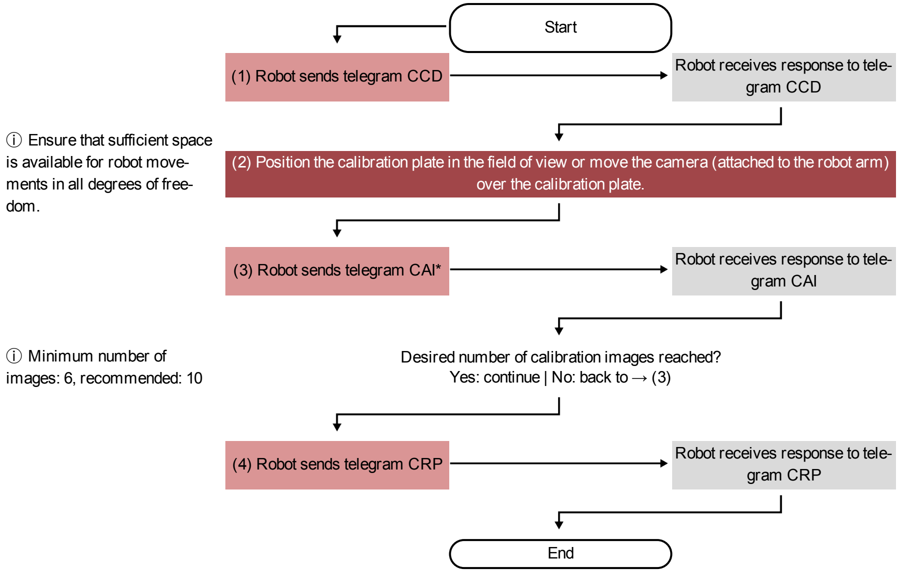
Were all four reference marks approached?
Yes: continue | No: back to → (6)

ⓘ Taught-in world points (X, Y) of the robot are used for calibration.

(8) Robot sends telegram CSF → Robot receives response to telegram CSF

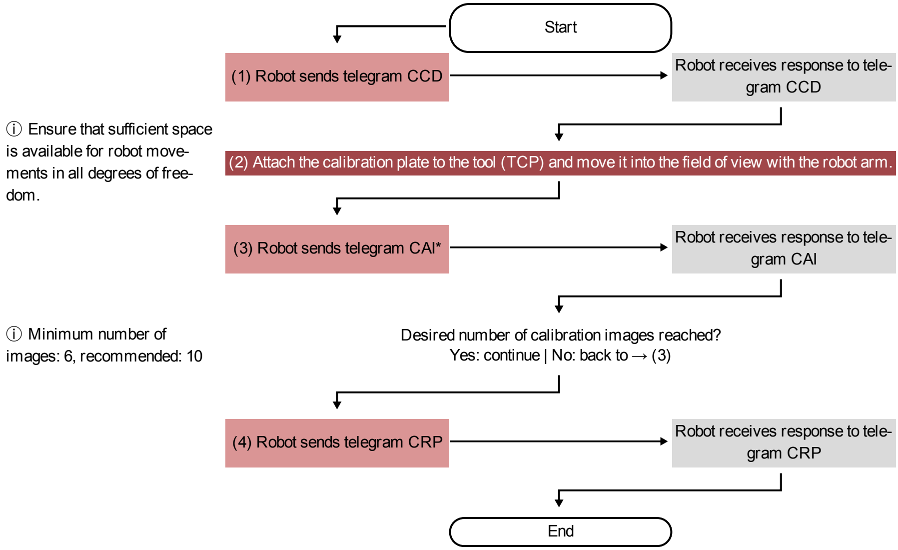
End

### 9.1.6.7.3 Automated calibration: Hand-Eye calibration (Robotics)




*Additional information: Notes on position selection for a Hand-Eye calibration (Robotics)

### 9.1.6.7.4 Automated calibration: Base-Eye calibration (Robotics)




*Additional information: Notes on position selection for a Base-Eye calibration (Robotics)

## 9.1.6.8 Validation of a robotics calibration

A validation can be performed after a successful robot calibration to check whether the robot camera system still delivers the desired accuracy. For this purpose the calibration plate must not be moved between calibration and validation. Typically, it is screwed tight. Validation is only possible on the basis of requests. Depending on the calibration method, use the following requests:

| | |
|---|---|
| Calibration plate (Robotics), Point pair list (Robotics) | CCD, CCP |
| Multi-image calibration | CCD, CMP |
| Hand-Eye calibration (Robotics), Base-Eye calibration (Robotics) | CCD, CAI, CRP |

Additional information: see Communications manual

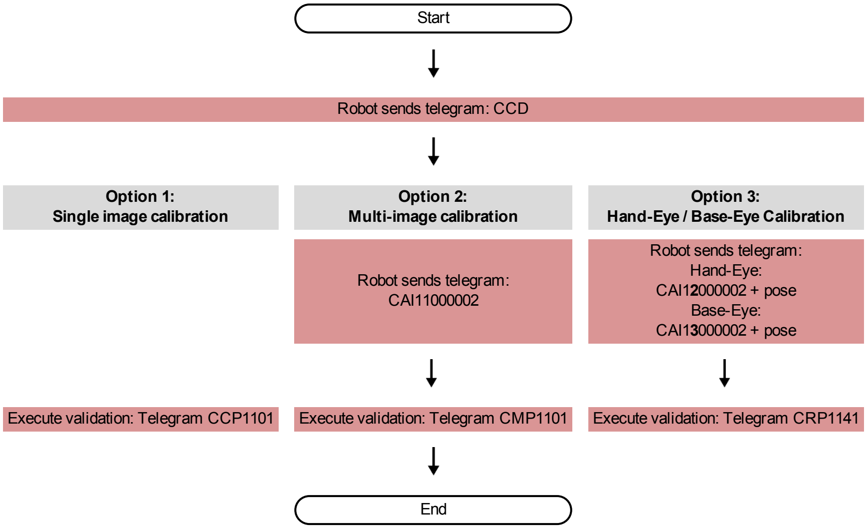
Select the option "Validation of the calibration" in the request CCP or CRP.

**Procedure**

1. Move the camera to the position above the calibration plate where the Measurement plane was set.
2. Call up the request (sequence) according to your selected calibration method. In the request-response (CCP or CRP), consider the mean deviation (RMSE).

When mounted on the gripper, this deviation is typically higher than that of the calibration, since it includes the accuracy of the robot positioning.

**Request sequences for validation**



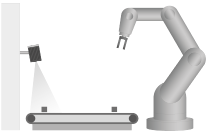
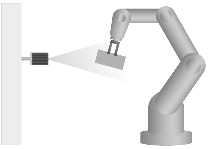
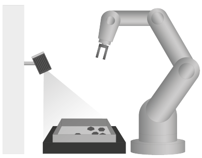

The deviations can be interpreted as follows:

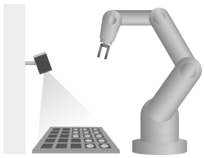
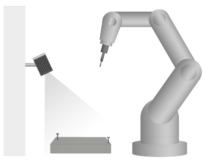
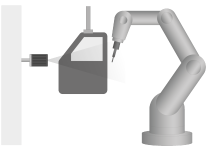
- The deviation is specified in pixels*1000, as this is a relative value to the field of view and therefore independent of the actual working distance of your application. If you divide the deviation value by the pixel resolution of your camera, you get the relative deviation of the field of view.
- With the request CGP it is possible to retrieve the average customer unit/pixel and thus convert the pixel value into the user unit, e.g. millimeter (parameter number 041).

As a guide value for mounting on the gripper, 0.4% deviation of the field of view is a realistic limit value above which a new calibration is recommended. Please check for yourself whether this applies to your application or whether you have to adapt it. For a stationary installation, the recommendation would be to set the limit value below 0.4% deviation of the field of view.

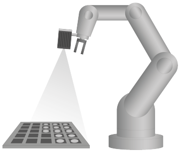
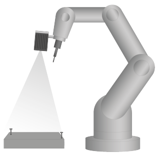
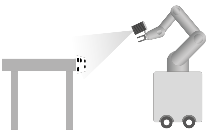
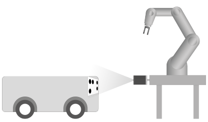
### 9.1.6.9 Application-specific calibration recommendations

**Applications: VISOR® Vision Sensor stationary**

<table>
<tr><th>Application example</th><th>Recommended calibration procedure</th></tr>
<tr><td colspan="2">Part is moved by robot (pick and place)</td></tr>
<tr><td><br>Picking from a conveyor belt</td><td>Calibration plate (Robotics) Multi-image calibration:<br>Performing the calibration<br>• in SensoConfig<br>• via telegram<br>Base-Eye calibration (Robotics):<br>Performing the calibration<br>• in SensoConfig<br>• via telegram</td></tr>
<tr><td><br>Fine positioning in gripper</td><td>Base-Eye calibration (Robotics):<br>Performing the calibration<br>• in SensoConfig<br>• via telegram</td></tr>
<tr><td><br>Picking from a vibration feeder</td><td>Calibration plate (Robotics) Multi-image calibration:<br>Performing the calibration<br>• in SensoConfig<br>• via telegram</td></tr>
</table>

| Application example | Recommended calibration procedure |
|---|---|
| <br>Picking from a load carrier | Calibration plate (Robotics) Multi-image calibration: Performing the calibration<br>• in SensoConfig<br>• via telegram<br>Base-Eye calibration (Robotics): Performing the calibration<br>• in SensoConfig<br>• via telegram |
| Part is processed by robot (screwing, gluing) | |
| <br>Automated screw insertion | Calibration plate (Robotics) Multi-image calibration: Performing the calibration<br>• in SensoConfig<br>• via telegram<br>Base-Eye calibration (Robotics): Performing the calibration<br>• in SensoConfig<br>• via telegram |
| <br>Glue bead application | Base-Eye calibration (Robotics): Performing the calibration<br>• in SensoConfig<br>• via telegram |

### Applications: VISOR® Vision Sensor in motion

| Application example | Recommended calibration procedure |
|---|---|
| Part is moved by robot (pick and place) | |
|  Picking from a load carrier | Hand-Eye calibration (Robotics):<br>Performing the calibration<br>• in SensoConfig<br>• via telegram |
| Part is processed by robot (screwing, gluing) | |
|  Automated screw insertion | Hand-Eye calibration (Robotics):<br>Performing the calibration<br>• in SensoConfig<br>• via telegram |
| Mobile workstation is calibrated | |
|  Calibration of mobile robots | Hand-Eye calibration (Robotics):<br>Performing the calibration<br>• in SensoConfig<br>• via telegram |
|  Calibration of driver-less transport systems | Base-Eye calibration (Robotics):<br>Performing the calibration<br>• in SensoConfig<br>• via telegram |

## 9.1.7 Cycle time tab

The Cycle time tab is used to configure the time response parameters for the VISOR® vision sensor.

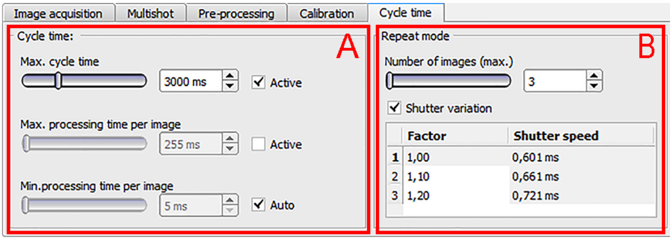



*Fig. 78: Setup Job, tab Cycle time*

### A: Cycle time

The cycle time measures the time from the trigger to the setting of the digital switching outputs. If the cycle time should be limited, e.g. because the machine cycle must not be exceeded, the value for the maximum cycle time must be limited accordingly. The result of all unfinished detectors up to this time is set to faulty. When choosing the maximum cycle time, it must be taken into account that this is not adhered to strictly. However, depending on the detector that has just been executed, it can take several milliseconds before it can break off. It is recommended that this maximum cycle time over the actual execution time is checked and the set maximum cycle time is reduced accordingly.

| Parameter | Function |
|---|---|
| Max. cycle time | Parameter for controlling the execution time of a cycle. Inside a cycle, some images can be evaluated (in case of "Number of images" >1). Maximum execution time is used to abort a cycle after a defined time. The result of the cycle after a timeout is always "not OK". The maximum cycle time should always be greater than the time required for a complete evaluation. |
| Max. processing time per image | Maximum duration of one evaluation inside a cycle including image acquisition |
| Min. processing time per image | Minimum duration of one evaluation inside cycle including image acquisition. The minimum processing time can be used to suppress multiple triggers. In case of "Number of images" = 1 (default) the "Min. processing time per image" corresponds to the minimum cycle time. |
| Auto | The "Auto" switch sets the "Min. processing time per image" so that the LED strength is always 100% and the processing time is minimal. |

### B: Repeat mode

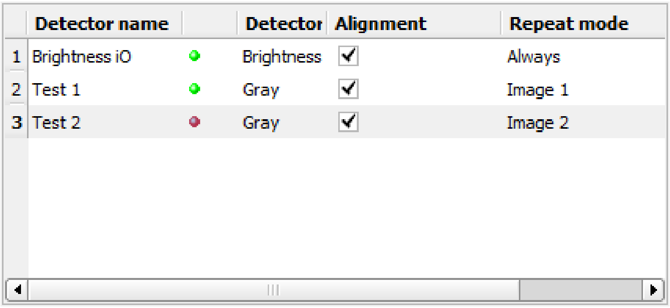
| Parameter | Function |
|---|---|
| Number of images (max.) | Maximum number of frames taken after a trigger, if none of the following abort criteria is met:<br>• "Overall job result"= positive (access via Output/Output Signals)<br>• "Max. cycle time" is not elapsed (if active).<br>Optional:<br>• Assign detector to an image, see also: Multiple image capture: Allocate the detector to an image |
| Shutter variation | If the shutter speed variation is "Active", a variation of several different shutter speeds can be created across a table. One image is then captured per set shutter speed, i.e. the first image is captured at shutter speed 1, the second image at shutter speed 2, etc. Default of the "Shutter variation" is "Off". In this case, the listbox is not displayed. |
| Factor and Shutter speed | The default value for the factor is: First value = 1.00 (the first value is always identical to 1.00 and is read-only). Subsequent default values each increase by 0.1, e.g. 1.10, 1.20, etc. The user can change the factor in the table, automatically adjusting the shutter speed (second column, read-only) and taking a picture. A mouse click in a row of the table will take a picture with the settings of the clicked table row.<br>**NOTE:**<br>If the parameter "Shutter speed" in the "Image acquisition" tab is changed, the shutter speed in the "Shutter variation" list box is recalculated. |

### Multiple image capture: Allocate the detector to an image

In the "Detector" setup, all set detectors are listed. If the parameter "Number of images (max.)" of the multiple image acquisition is greater than 1, you get the option to assign a detector to an image acquisition. In the column "Repeat mode", this setting can be made for each detector.

- Always: Run for all image acquisition operations
- Recording n: Executed in the corresponding image acquisition

Open the selection table by double-clicking.



| | Detector name | | Detector | Alignment | Repeat mode |
|---|---|---|---|---|---|
| 1 | Brightness iO | green | Brightness | ✓ | Always |
| 2 | Test 1 | green | Gray | ✓ | Image 1 |
| 3 | Test 2 | red | Gray | ✓ | Image 2 |

*Fig. 79: Detector list, Multiple image capture*

**Min. processing time per image**

Bei Verwendung der internen Beleuchtung hängt die minimale Zykluszeit von der eingestellten Belichtungszeit ab (länger => höhere min. Zykluszeit).

## 9.2 Setup Alignment

For objects or features whose position varies in the image, Alignment may be useful or necessary. The Alignment determines the object / feature position in the image. Three different detection methods (alignment detectors) are available for this purpose: Pattern matching, Edge detector, and Contour matching.

**Alignment functionality:**

An Alignment is an aligned coordinate system anchored to a selected feature. Defined detectors are alignment relative to this coordinate system. The aligned coordinate system is drawn in dark blue (for information on the meaning and adjustment of the different frames: see Search and feature ranges).

**NOTE:**

- A maximum of one alignment detector can be defined for each job.
- For each detector in the job, it can be selected whether the detector should be aligned with the Alignment.
- Since the Alignment is an additional calculation step and thus makes use of cycle time, it should only be used if the application requires it.

### 9.2.1 Selection and configuration of Alignment

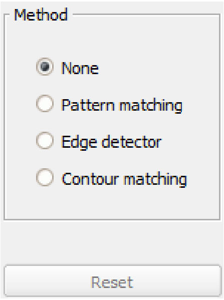



*Fig. 80: Selecting Alignment*

1. Click on the setup button "Alignment".
2. Select a detection method in the configuration window "Method".

| Method | Selection criteria |
|---|---|
| None | Alignment not active |
| Pattern matching | Detection of contours and edges at any angle.<br>Pattern matching can be used preferably if:<br>• There are no / few high-contrast or paraxial edges, but areas with gray value patterns are present in the image<br>• an angular offset (rotation offset against teach-in position) of up to 360° can occur. |
| Edge detector | The detection of edge should always be selected:<br>• if an offset of the position in X and / or Y direction can occur.<br>• at a maximum angle offset (rotational offset compared to the teach-in position) of approx. ± 20° (depending on the object and application).<br>• if there are edges with strong contrast<br>If the above-mentioned criteria are fulfilled, Edge detector is a very quick method of Alignment. |

| Method | Selection criteria |
|---|---|
| Contour matching | Detection of contours and edges at any angle<br>Contour detection must always be used if:<br>• An angular offset (rotation offset against teach-in position) of up to 360° can occur.<br>It can be used preferably when high-contrast edges are present in the image. The relatively complex contour detection function usually results in a comparatively longer cycle time. |

### Configuration of alignment detector

1. Adapt the position and size of the search and function ranges displayed on the screen if necessary.
2. Configure the alignment detector in the Parameters tab.

### Set Alignment active for detectors

In the "Detectors" setup, all set detectors are listed. In the "Alignment" column, it is possible to select for each detector whether it should be aligned by the adjusted Alignment. Default value is "Active".

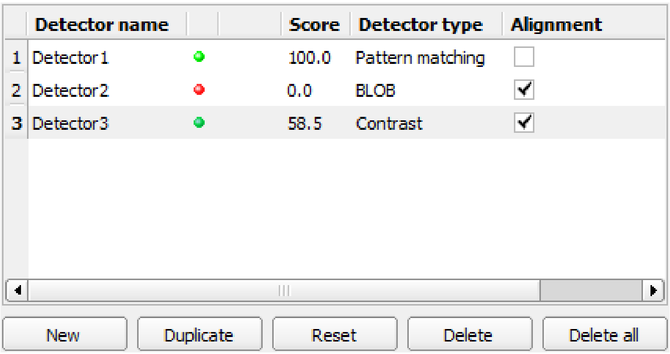



*Fig. 81: Detector list, Alignment active / inactive*

### Reset

The "Reset" button can be used to restore the factory settings for the selected alignment detectors.

## 9.2.2 Alignment Pattern matching

This method is suitable for detecting patterns of any shape, even without clear edges or contours The patterns of the object in the search field are stored on the sensor during teach-in. In run mode, the sensor searches for the position of the greatest match with the taught pattern in the current image. If the match is greater than the set threshold value, the pattern is considered recognized and its position is used as the origin of the coordinate system of the Alignment. The pattern recognition is completely rotational tolerant, i.e. the searched object may appear in any position in the image (choose angle setting accordingly).

### 9.2.2.1 Color Channel tab

In the Color Channel tab, a color image (3 channel) can be converted to a gray value image (1 channel). In contrast to the gray scale value image on a monochrome VISOR® vision sensor, contrasts can be increased significantly. The highlighting of a color can be set individually for each detector. Thus, the flexibility compared to the use of optical color filters is significantly higher.

The image displayed is dependent on the selected detector.

- Color detectors: Display always colored.
- Object recognition detectors: Monochrome image, display dependent on the selected color space and the color channels

**Parameter description:**

| Parameter | Function |
|---|---|
| Color space | Color spaces:<br>RGB, Color model RGB (Page 355)<br>HSV, Color model HSV (Page 355)<br>LAB, Color model LAB (Page 356) |
| Selection color filter | Depending on the color space, all or part of the following color filters are available:<br>Color channel (default)<br>Color distance<br>Binarization |
|  | Switching the image between color and monochrome. |

#### 9.2.2.1.1 Selecting a color filter

The following color filters are available:

**Color channel (default)**

The selected color channel is used as a gray value image.

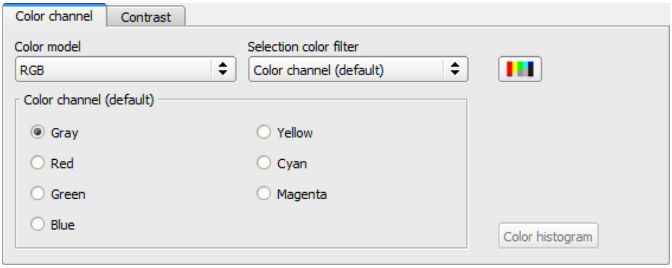



*Fig. 82: Color filter, Color channel (default)*

### Color distance

A color is selected as reference color by specifying the color model values or by pipette. The gray value image indicates the distance of each pixel to this reference color. Typical application: Segmentation of letters for OCR.

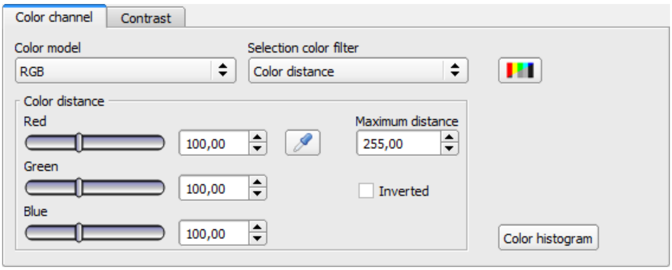



*Fig. 83: Color filter, Color distance*

| **Parameter** | | **Function** |
|---|---|---|
| Red<br>Green<br>Blue | Luminance<br>A<br>B | Color channels:<br>The color channel can be set via the slider or by entering a value (default 0). |
| Pipette symbol | | With the selection of the pipette button and then clicking on the image, the selected color channel is determined automatically. |
| Maximum color distance | | Distance of the current color versus the taught-in color. Colors that will exceed the maximum color distance will be black or white depending on the setting of "Inverted". |
| Inverted | | Inversion of the color distance image. |

### Binarization

A color range is selected. All pixels within this color range become white. Pixels with deviating color values become black.

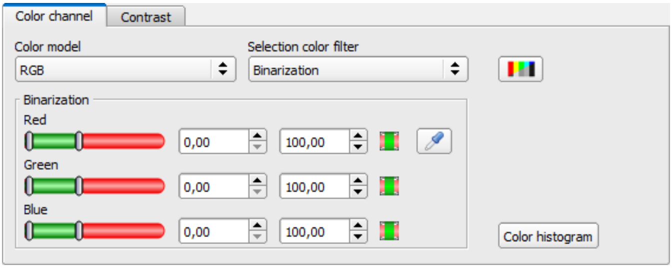



*Fig. 84: Color filter, Binarization*

| Parameter | | | Function |
|---|---|---|---|
| Red<br>Green<br>Blue | Shade (hue)<br>Saturation<br>Brightness (value) | Luminance<br>A<br>B | Determination of the color range. The color ranges can be user-defined via the slider or set by entering a value. |
| Invert button / | | | The current setting is inverted when selecting the button. |
| Pipette symbol | | | With the selection of the pipette button and then clicking on the image, the selected color channel is determined automatically. |

### 9.2.2.2 Parameters tab

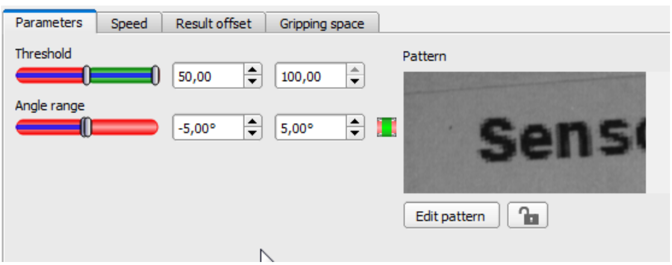



*Fig. 85: Alignment Pattern matching, Parameters tab*

The following parameters can set in the "Parameters" tab:

| Parameter | Function |
|---|---|
| Threshold | Range for the required concordance of the found sample with the taught sample |
| Angle range | Angle range in which to search (larger range leads to longer processing times). Depending on the size and complexity of the image, the Angle range may be limited. |
| Pattern | Shows the taught pattern (red frame in the field of view) |
| Edit pattern | By masking the pattern, areas of the taught-in pattern can be deactivated (see also Function: Edit pattern / contour (Page 181)) |
| Lock | Lock / unlock pattern. When locked, the taught-in pattern is protected against (unintentional / accidental) changes, e.g. accidental adaptation of the teaching range. Unlock to modify taught pattern. |

### 9.2.2.3 Speed tab

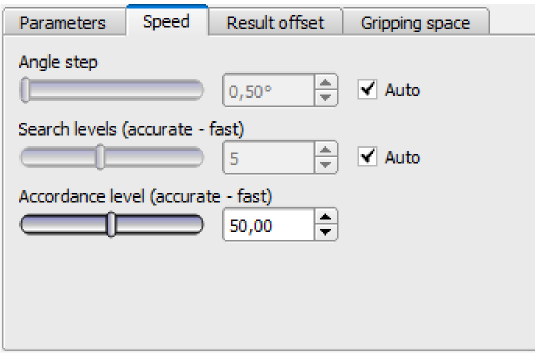



*Fig. 86: Alignment, Pattern matching, Tab Speed*

The execution speed is influenced by the adjustable speed parameters. The search is performed either less finely, i.e. earlier canceled and thus faster, or even finer details are taken into account in the search, i.e. search longer and the search is slower.

**Parameter description:**

| Parameter | Function |
|---|---|
| Angle step | Sensitivity of the search throughout the selected angle range in degrees [°]<br>(smaller value leads to longer machining times, but higher accuracy) |

| Parameter | Function |
|---|---|
| Search levels (accurate - fast) | Number of search levels (one search level corresponds to an image with half resolution)<br>• Small value (accurate): Slow search = lower risk (less likely to overlook candidates)<br>• High value (fast): Fast search = higher risk (candidates can be overlooked) |
| Accordance level (accurate - fast) | Candidates with a degree of compliance below the specified value are already discarded during the search.<br>• Small value (accurate): Late rejection = slower = less risky<br>• High value (fast): Early rejection = quicker = riskier<br>In case of false results, this value can be decreased (more accurate). |
| Auto | Automatic setting |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

### 9.2.2.4 Result offset tab

With the Result offset, the final position of a found object can be modified. This can be helpful e.g. to define a suitable gripping point on the found part in pick and place applications. Depending on the detector, different options are available.

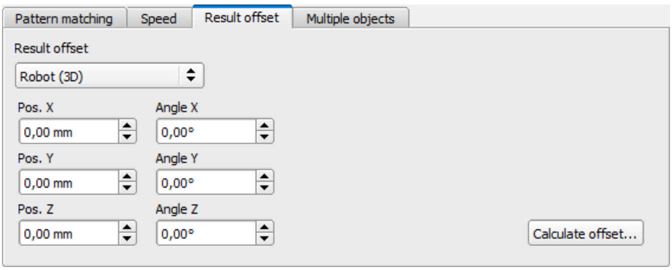
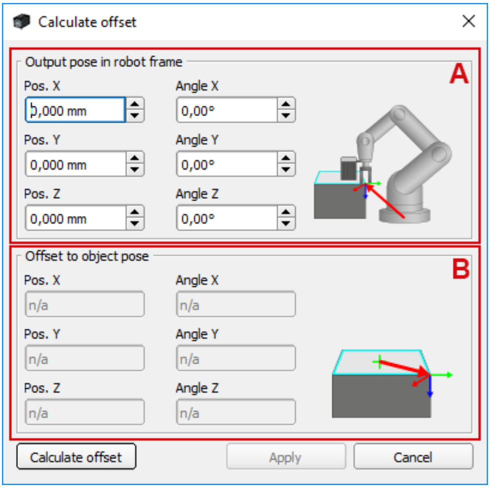



*Fig. 87: Result offset tab, Option "Robot (3D)"*

**Parameter description:**

| Parameter | Function |
|---|---|
| Off | No offset, i.e. automatically determined center of the found object / finder region. |

| Parameter | Function |
|---|---|
| Image plane (in pixels) | Freely selectable position (graphically or by value input) in the Image Coordinate System and in pixels<br>• "X": Offset in X direction<br>• "Y": Offset in Y direction<br>• "Angle": Angle offset |
| Align (2D) | Freely selectable position (by value input or calculation using e.g. gripping point) in the World coordinate system and in user unit<br>• "X": Offset in X direction<br>• "Y": Offset in Y direction<br>• Angle: Angle offset |
| Robot (3D) | Freely selectable position (by value input or calculation using e.g. gripping point) in the World coordinate system and in user unit<br>• "X": Offset in X direction<br>• "Y": Offset in Y direction<br>• "Z": Offset in Z direction<br>• "Angle": Angle offset<br>• "Angle X": Rotation around X-axis<br>• "Angle Y": Rotation around Y-axis<br>• "Angle Z": Rotation around Z-axis |

| Parameter | Function |
|---|---|
| Button "Calculate offset" (only for Align (2D) and Robot (3D)) | Opens dialog window "Calculate offset 2D / 3D".<br><br><br>*Fig. 88: Calculate offset*<br>**A**: Data input, e.g. gripping point<br>**B**: Calculated value (read-only) |

### 9.2.2.5 Gripping space tab

Robots grip objects, e.g. with a jaw gripper, on the outer contour of the objects. If the objects touch or overlap, gripping with the robot may not be possible. The VISOR® gripping space check can be used to check whether the gripping positions on the object are actually clear in the required size.

The gripping space check is available for Contour matching and Pattern matching Alignment.

The VISOR® gripping space check is an extension of the Alignment. For an Alignment **without** gripping space check, the position of the object that has the highest score value is output. **With an** active gripping space check, the position of the first object found is output, in which its tracking detectors (gripping ranges) are also "OK" (according to the logical links in the overall result).

**Procedure:**

1. The Alignment Contour matching / Pattern matching identifies objects as candidates whose contour / pattern matches the taught contour / pattern.
2. These candidates are sorted. The sorting takes place according to the set values for "Sorting criteria" and "Sorting order" in the "Gripping space" tab.

3. In this order, the candidates are checked to see if the detectors aligned by the Alignment (e.g. checking for gripping space) are all met. This happens under consideration of the logical links in the overall result.
   In setup step "Output" / "Digital output" tab, logical links can be made to evaluate the objects. Here, for example, free spaces for different gripping positions can be defined (see figure below).
4. The position data of the first object that meets all these criteria are output, and the search is terminated at this point.

The gripping positions X-X and Y-Y are possible for the object shown in the following figure. Of these gripping possibilities, only those that are actually necessary for one grip can then be checked for "free".

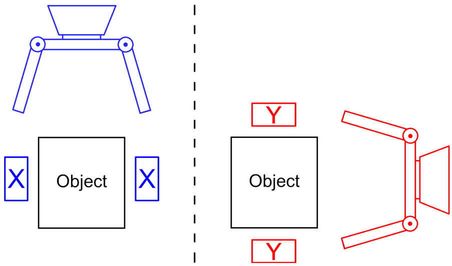



*Fig. 89: Possible gripping position X-X (left) and possible gripping position Y-Y (right).*

**NOTE:**
In the detector list, Alignment per detector can be activated or deactivated (default: active). Only detectors activated here are effective for the gripping space check.

Prerequisite for successfully finding an object is at least one object per image / evaluation, where the overall result is "OK", i.e. also tracked detectors!

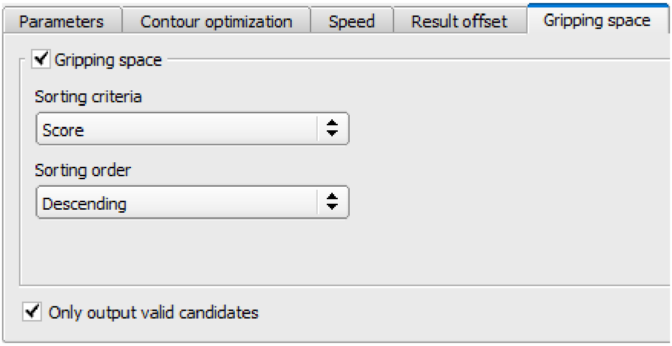



*Fig. 90: Alignment Contour matching, Gripping space tab*

The following parameters can set in the "Gripping space" tab:

| Parameter | Function |
|---|---|
| Sorting criterion<br>• Score<br>• Position X<br>• Position Y<br>• Angle Z<br>• Scaling | Sorting criteria according to which objects should be "pre-sorted".<br>• Overall result<br>• Position X<br>• Position Y<br>• Angle Z<br>• Scaling (only for Alignment Contour matching) |
| Sorting order<br>• Ascending<br>• Descending | Sorting order for the selected sorting criterion.<br>The values of the sorting criteria are sorted in ascending order.<br>The values of the sorting criteria are sorted in descending order. |
| Only output valid candidates | If this checkbox is activated, only objects whose score value is above the set threshold ("Parameters" tab) are displayed and output. It can be used e.g. for parameter optimization. |

## 9.2.3 Alignment Edge detector

This Alignment determines the object position and thus the aligned coordinate system based on the intersection point of edges in the image. Angle positions up to approx. ±20% deviation (depending on the object) can be compensated.

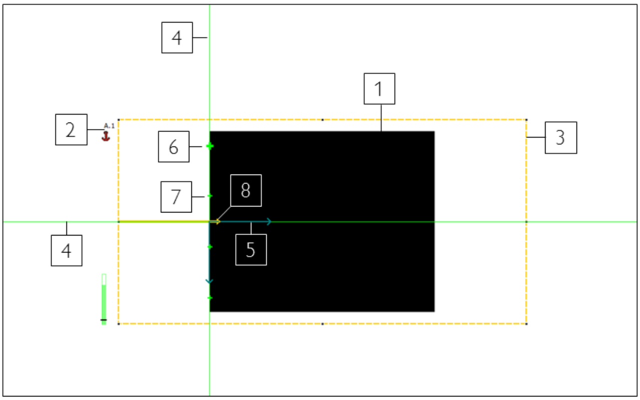
### 9.2.3.1 Structure of the Edge detector

The "Edge detector" is carried out using "Probes". Depending on the probe type, there are between one and three probe(s). The search range of a probe is indicated by the yellow frame

(ROI). Within this ROI, the object is searched for and the edge of the object is scanned. The scanning is performed in the direction of the yellow arrow, the "Search direction". This yellow arrow can also be used to turn the search range of the detector.

From the starting point of the search range, search stripes (number can be set as desired) are sent out into the search direction. If the search stripe touches the edge of the object, the "Touching point" of the search stripe is marked with a cross at this point. Depending on the number and the setting, there may be a "Winner search stripe", the touching point of which is shown in bold.

Which edge of an object is probed can be seen on the "Scanning line" in the search direction. If an object is not scanned from both the X and Y direction but only from one direction, the second scanning line is at the center of the search range. The arrows with the origin at the intersection of the scanning lines form the aligned coordinate system. In the following figure, the structure of the Alignment "Edge detector" is visualized.




[1] Object
[2] Probe 1
[3] Search range of probe 1
[4] Scanning line
[5] Tracked coordinate system
[6] Touching point of the winning search stripe
[7] Touching point of a search stripe
[8] Search direction of the probe

*Fig. 91: Structure of the Edge detector*

### 9.2.3.2 Color Channel tab

See Chapter: Color Channel tab (Page 147)

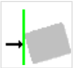
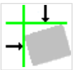
### 9.2.3.3 Parameters tab

The probe type must be selected to perform the Edge detector. The probe type determines which change in position of the object can be aligned: Shift in one or two directions, rotation. The following probe types are available and recommended for use with varying object positions ...

① ...in one direction

② ...in two directions

③ ...with rotation

| Probe type | Function | ① | ② | ③ |
|---|---|---|---|---|
| 1<br> | **One probe: Alignment for shift in one direction**<br>Object position is aligned when moving in one direction.<br>The position of the scanning line is determined by the probe's scanning direction. The other scanning line is in the middle of the search range (ROI).<br>**NOTE:**<br>A rotation of the object position is not aligned. | ✓ | | |
| 2<br> | **One probe: Alignment for shift in one direction and rotation**<br>Alignment of object position by shift in one direction and rotation.<br>The position of the scanning line is determined by the probe's scanning direction. The other scanning line is in the middle of the search range (ROI). | ✓ | | ✓ |
| 3<br> | **Two probes: Alignment for shift in two directions**<br>Object position is aligned when moving in two directions.<br>The position of the scanning line in X direction of the coordinate system is determined by Probe 1.<br>The position of the scanning line in Y direction of the coordinate system is determined by Probe 2.<br>The origin of the coordinate system lies at the intersection of both scanning lines.<br>**NOTE:**<br>A rotation of the object position is not aligned. | | ✓ | |
| 4<br> | **Two probes: Alignment for shift in two directions and rotation**<br>Object position is aligned by shift in two directions and rotation.<br>The position of the scanning line in X direction of the coordinate system is determined by Probe 1.<br>The position of the scanning line in Y direction of the coordinate system is determined by Probe 2.<br>The origin of the coordinate system lies at the intersection of both scanning lines.<br>In addition, the orientation of the object is determined. Probe 2 is rotated and moved according to the object movement.<br>The position of Probe 2 is aligned relative to the position and orientation of the scanning line of Probe 1. | | ✓ | ✓ |

| Probe type | Function | ① | ② | ③ |
|---|---|---|---|---|
| 5  | **Three probes: Alignment for shift in two directions and rotation**<br>Object position is aligned by shift in two directions and rotation.<br>A straight line is drawn through the touching points of the winner search stripes of Probes 1 and 2. This scanning line (12) determines the position and orientation of the coordinate system. The origin of the coordinate system lies at the intersection of Scanning line 12 and Scanning line 3. Probe 3 is rotated and moved according to the object movement.<br>The position of Probe 3 is aligned relative to the position and orientation of Scanning line 12. | | ✓ | ✓ |

After selecting the probing mode, the corresponding parameters must be determined. The following parameters can set in the "Parameters" tab:

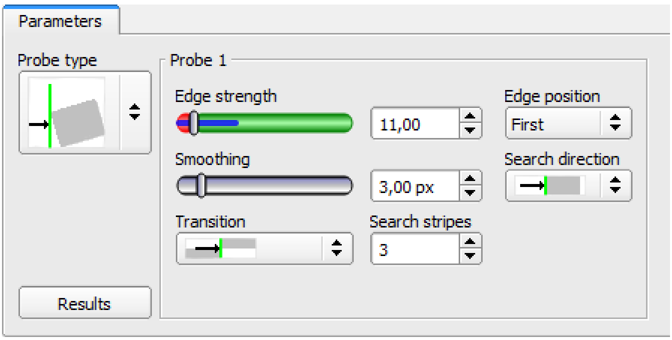



*Fig. 92: Alignment Edge detector, Parameters tab*

| Parameter | Function |
|---|---|
| Edge thickness | Edge thickness / contrast at which an edge should be detected as an edge. |
| Smoothing | The edge contour is smoothed in the search direction. With larger values, noisy edges, blurred edges, or edges that are not perpendicular to the search direction, are detected more reliably. In addition, light-dark-light or dark-light-dark transitions which are close together with larger values can be ignored. Thus, interfering edges, e.g. scratches, can be hidden. The effect of smoothing can be displayed graphically using the button "Results". |

| Parameter | Function |
|---|---|
| Transition | With the "Transition" parameter, the edge transition can be determined. |
| Both directions  | Edge transition from light to dark and vice versa. |
| Light → dark  | Edge transition from light to dark. |
| Dark → light  | Edge transition from dark to light. |
| Search stripes | Number of parallel search stripes into which the width of the search range is divided. Edge detector is carried out in each search stripe, and the first edge is decisive. |
| Edge position | The parameter "Edge position" determines which edge should be detected from the search direction. It is determined how the winner search stripes and thus the edge position are determined. |
| • First | The first edge in the search direction is detected.<br>The distances from the beginning of the search range to the touching points of all search stripes in the search direction are determined. The winner search stripe is the one with the shortest distance to the beginning of the search range. |
| • Last | The last edge in the search direction is detected.<br>The distances from the beginning of the search range to the touching points of all search stripes in the search direction are determined. The winner search stripe is the one with the longest distance to the beginning of the search range. |
| • Median | The distances from the beginning of the search range to the touching points of all search stripes in the search direction are determined. The median value of these distances is then formed. |
| • Mean | The distances from the beginning of the search range to the touching points of all search stripes in the search direction are determined. The mean of these distances is then formed. |
| Orientation | The "Orientation" parameter defines the type of the scanning line determination. |
| • Best-fit line | In this setting, the scanning line is determined by placing a best-fit line through all search stripes. |

| Parameter | Function |
|---|---|
| • Edge guide | In this setting, a scanning line is determined, which acts like a mechanical edge stop. This makes it possible to achieve more robust results for convex-shaped edges than with a simple best-fit line. |
| Search direction | This parameter determines the search direction of the probes. From this direction, the object edge is aligned. All probes can be rotated with the small black arrow. |
|  | The search direction takes place in only one direction: the direction of the yellow arrow (ROI). The touching points and thus the origin of the coordinates lie at an edge of the object. |
|  | For each search stripe, a touching point is determined from both directions of the probe. The center between these touching points is then determined. The origin of the coordinate system is at the center of the winner search stripe, i.e. in the object. |
| Results | Opens the results and histogram window.<br>For more information, please refer to: Results / Histogram window (Page 236) |

### Improvement of execution speed

- Search range for position (yellow frame) only as large as necessary.
- Reduce search stripes
- Reduce smoothing value
- Reduce resolution from SVGA to QQVGA, QVGA or VGA

**ATTENTION:**
This parameter affects all detectors!

### Robust detection

- If edges are blurred: Increase smoothing value
- If interfering edges such as scratches are detected: Increase switching threshold or / and smoothing value
- If edge is not perpendicular to search direction: Increase search stripes

### 9.2.3.4 More information on Edge detector (Alignment)

#### 9.2.3.4.1 Effect of number of search stripes

"Search stripes" represent how many parallel search stripes the search range is divided into. Edge detector is performed separately in each search stripe. The first edge in the search direction of all search stripes is considered to be the overall result. Increasing the value of "search stripes" ensures that the first edge is found in the search range.

When increasing "search stripes", the edge thickness found can fluctuate greatly, e.g. if only half the search range is occupied by the edge. The reason for this is that the edge thickness is displayed for the first (not the thickest) edge, which lies in the search direction above the switching threshold.

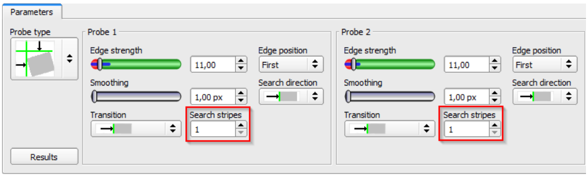
<table>
<tr><th colspan="2">Edge detector with search stripes = 1<br>The dominating edge perpendicular to the search direction is found.</th></tr>
<tr><td>Settings in the Parameters tab:<br>Search stripes = 1</td><td>


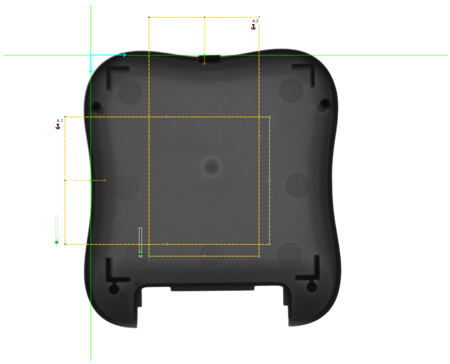
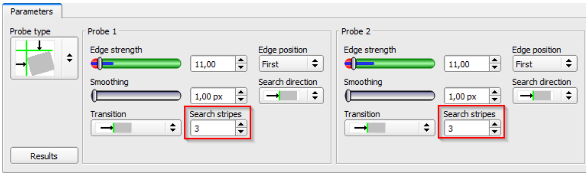
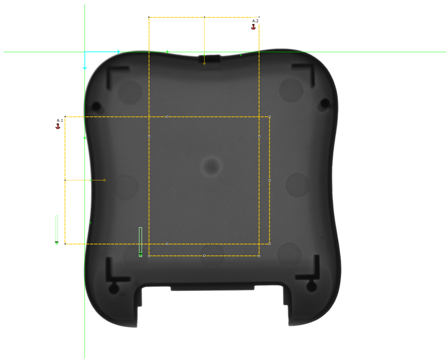
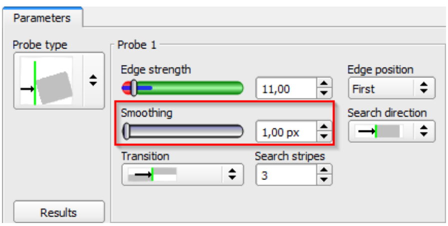

Fig. 93: Parameters "search stripes" = 1</td></tr>
<tr><td>Image display:<br>The dominating edge perpendicular to the search direction is found.</td><td>


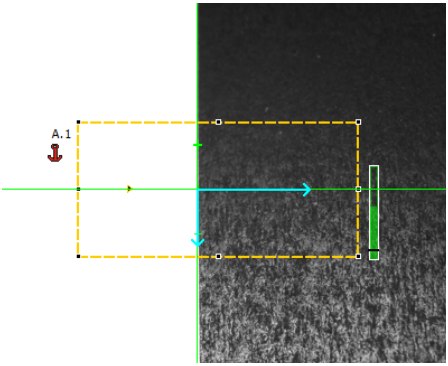
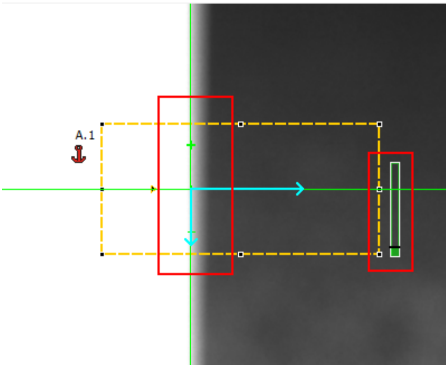
Fig. 94: Dominating edge perpendicular to the search direction</td></tr>
</table>

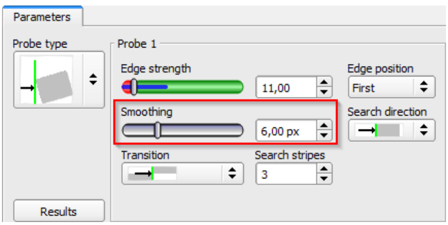
| Edge detector with search stripes > 1.<br>The first edge perpendicular to the search direction is found | |
|---|---|
| Settings in the Parameters tab:<br>Search stripes = 3 | <br><br>*Fig. 95: Parameters "search stripes" = 3* |
| Image display:<br>The first edge perpendicular to the search direction is found | <br>*Fig. 96: First edge perpendicular to the search direction is found* |

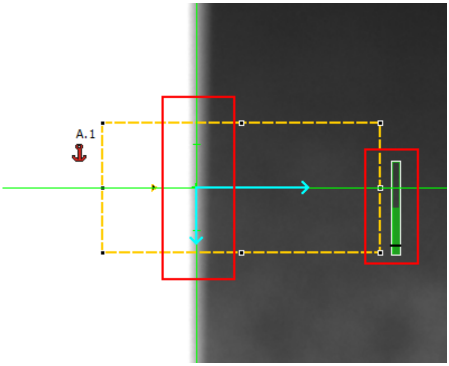
## 9.2.3.4.2 Function of smoothing on sharp or blurred edges

The edge thickness results from the addition of edge steps over an area in the search direction, the size of which is given by the "smoothing" parameter.
For sharp edges, the edge thickness is not increased by increasing smoothing.
However, for blurred edges, the edge thickness is increased by increasing smoothing.

| Detection of sharp and blurred edges with a low level of smoothing | |
|---|---|
| Settings in the Parameters tab:<br>Smoothing = 1 | <br><br>*Fig. 97: Parameters smoothing = 1* |

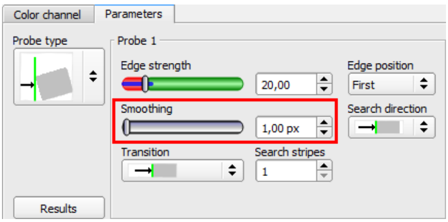
| | |
|---|---|
| Image display: Detection of sharp edge. High edge thickness with a low level of smoothing. |  *Fig. 98: High edge thickness with a low level of smoothing* |
| Image display: Detection of blurred edge. Low edge strength with a low level of smoothing. |  *Fig. 99: Low edge strength with a low level of smoothing* |

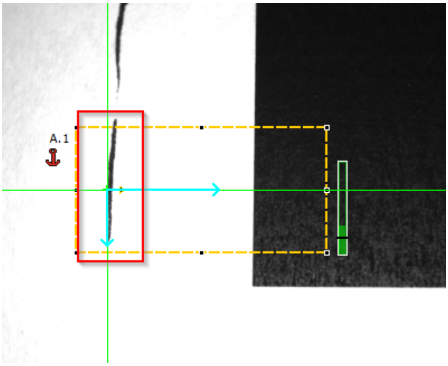
| Detection of blurred edges with a high level of smoothing | |
|---|---|
| Settings in the Parameters tab: Smoothing = 6 |  *Fig. 100: Parameters smoothing > 1* |

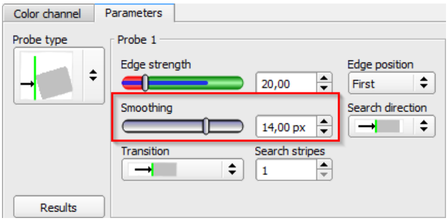
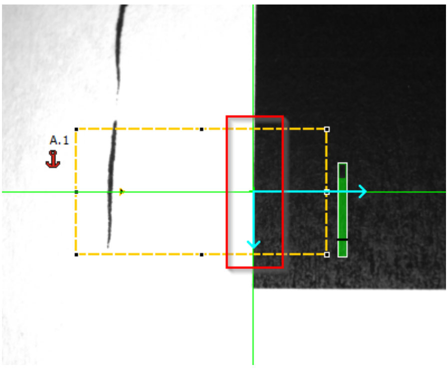
| | |
|---|---|
| Image display: Detection of blurred edges. High edge strength with a high level of smoothing. |   *Fig. 101: High edge strength with a high level of smoothing* |

### 9.2.3.4.3 Effect of smoothing on interfering edges

As mentioned above, the edge thickness results from the addition of edge steps over an area in the search direction, whose size is given by the "smoothing" parameter. If edges of different polarity lie in this area (dark-bright: positive polarity, bright-dark: negative polarity), their edge steps can neutralize each other. This can be used to eliminate interfering edges, by choosing "smoothing" which is sufficiently large.

| Detection with smoothing = 1. Interfering edge is not skipped. | |
|---|---|
| Settings in the Parameters tab: Smoothing = 1 |   *Fig. 102: Parameters smoothing = 1* |

| | |
|---|---|
| Image display:<br>Interfering edge is not skipped. | <br><br>*Fig. 103: Interfering edge is not skipped* |

| Detection with smoothing > 1. Interfering edge is skipped. | |
|---|---|
| Settings in the Parameters tab:<br>Smoothing > 1 | <br><br>*Fig. 104: Parameters smoothing > 1.* |
| Image display:<br>Interfering edge is skipped. | <br><br>*Fig. 105: Interfering edge is skipped* |

### 9.2.4 Alignment Contour matching

This detector is suitable for detecting contours based on edges. The contours of the object in the search range are taught and stored in the sensor. In Run mode, the sensor searches for the position of the best fit with the taught contour in the current image. If the fit is higher than the selected threshold, the result is considered positive. Contour detection is completely tolerant to rotation positions, i.e. the searched object may appear in any position in the image (choose angle setting accordingly).

#### 9.2.4.1 Color Channel tab

See Chapter: Color Channel tab (Page 147)

#### 9.2.4.2 Parameters tab

The most important parameters for contour detection can be set in the "Parameters" tab.

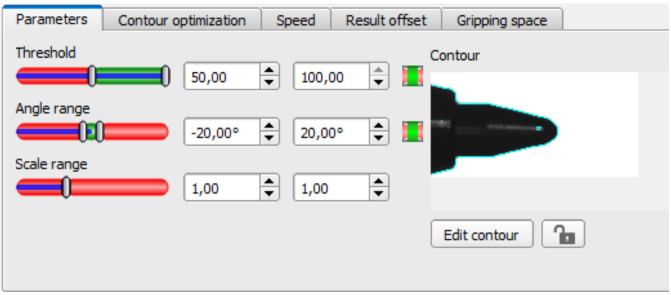



*Fig. 106: Alignment Contour matching, Parameters tab*

The edges marked in light blue at the bottom right (high-contrast transitions in the image) were identified and drawn in on the basis of the parameter settings made in the teach-in area (red frame). These can be further altered by changing the parameters or with the function "Edit contour". The VISOR® vision sensor now searches for this contour in the image within the search range (yellow frame).

The following parameters can be set in the "Parameters" tab:

| Parameter | Function |
|---|---|
| Threshold | Value for the required match of the found contour with the taught-in contour |
| Angle range | Angle range in which to search (larger range leads to longer processing times). |
| Scaling | Detection also of enlarged or reduced objects in a given scale range. |

| Parameter | Function |
|---|---|
| Contour | Shows the taught-in contour |
| Edit contour | Via the parameter "Edit contour", ranges of the taught-in contour can be hidden. As with an eraser, the areas that are not needed for the evaluation can be removed in the search range. The setting can also be inverted via the option "Invert all".<br>Additional information: Function: Edit pattern / contour (Page 181) |
| Lock | Locking / unlocking the contour: In the locked state, the taught-in contour is protected against (unintentional) change, e.g. accidentally adjusting the learning area. Unlock (click the lock icon again) for changing the contour. |

Additional information:

### Improvement of execution speed

- Search range for position (yellow frame) only as large as necessary.
  The search range indicates the area in which the center of gravity of the contour is searched.
- Search range for angles only as large as necessary.
- The search range for scaling is only as large as necessary
- Reduce resolution (e.g. to WGA)

  **ATTENTION:**
  This parameter affects all detectors!

- Set slider (accurate - fast) to "fast"
- Wert für „Min. Kontrast Modell“ erhöhen, da kleine Werte zu einer größeren Anzahl an extrahierten Konturen führen können. In the display of the pattern, check whether the relevant contours are still present.
- Wert für „Min. Kontrast Bild“ erhöhen.
- Especially in the case of Alignment: Use an alternate search pattern. For example, with higher contrast, so that "Min. Kontrast Modell" and "Min. Kontrast Bild" can be increased.

### Robust detection

- Search range (yellow frame) sufficiently large?
- Search range for angles sufficiently large?
- Search range for scaling sufficiently large?
- Min. Kontrast Modell and Min. Kontrast Bild sufficient? Is the contrast sufficient when acquiring the model and the images that are to be checked? Is the model detectable in the images that are to be checked?
- Set slider (accurate - fast) to "accurate"

- Objects should not overlap.
- Are there distinctive edges in the model? If necessary, teach in the model again so that prominent edges lie in the taught-in model.
- Is "Min. Kontrast Modell" suitably selected? If the relevant contours are not displayed in the taught-in model, reduce "Min. Kontrast Modell". If too many contour lines are shown, increase "Min. Kontrast Modell".
- Is "Min. Kontrast Bild" suitably chosen for the current image? If the current image has a smaller / larger contrast than the taught-in pattern, "Min. Kontrast Bild" should be smaller / larger than "Min. Kontrast Modell".
- Model found in wrong position? If the taught-in model is not unique, teach-in a new model.
- Does the result value fluctuate from image to image? If necessary, ensure that no "wrong" contours are taught in the image (edges due to shadows or contour fragments that are not desired in the contour model). This can be achieved by increasing "Min. Kontrast Modell". With the help of "Edit contour", search ranges can be hidden.

**Parameter angle range: Rotational direction of angle**

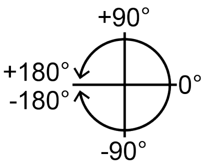



*Fig. 107: Rotational direction of angle*

### 9.2.4.3 Contour optimization tab

In the "Contour optimization" tab, further settings for the edge transition and the contrast can be made.

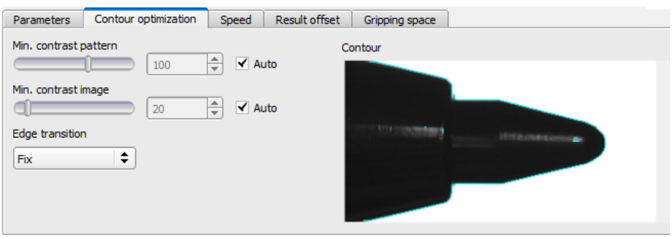



*Fig. 108: Alignment Contour matching, tab Contour optimization*

The following parameters can be set in the "Contour optimization" tab:

| Parameter | Function |
|---|---|
| Min. Kontrast Modell | Minimum required contrast for the taught-in model, in which an edge is accepted as such. |
| Min. Kontrast Bild | Minimum contrast required in current image for an edge to be accepted as one. |
| Edge transition<br>• fix<br>• fix + inverted<br>• flexible | The parameter "Edge transition" can be used to determine the transition between object or contour and background. The way in which the contour is to be recognized is selected:<br>• "fix" = only on the taught-in background<br>• "fix + inverted" = only on the taught-in and inverted background<br>• "flexible" = on any background<br>For more information, please refer to "Additional information:". |
| Auto | Automatic setting |

**Additional information: see description** Edge transition Contour detector

### 9.2.4.4 Speed tab

Using the adjustable parameters in the Speed tab, execution speed of the sensor can be altered. Adjusting the search levels sets the level of detail of the search and thus the time for a particular search. The search is either performed less finely, i.e. the search is canceled sooner and is therefore faster, or even finer details are taken into account in the search, i.e. a longer search is performed and the search is slower.

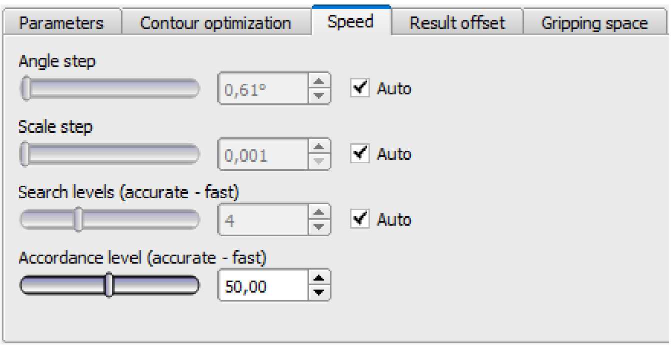



*Fig. 109: Alignment Contour matching, Speed tab*

The following parameters can set in the "Speed" tab:

| Parameter | Function |
|---|---|
| Angle step | Sensitivity of the search throughout the selected angle range in degrees [°] |
| Scale step | Sensitivity of the search throughout the selected scale range |
| Search levels (accurate - fast) | Number of search levels (one search level corresponds to an image with half resolution)<br>• Small value (accurate): Slow search = lower risk (less likely to overlook candidates)<br>• High value (fast): Fast search = higher risk (candidates can be overlooked) |
| Accordance level (accurate - fast) | Candidates with a degree of compliance below the specified value are already discarded during the search.<br>• Small value (accurate): Late rejection = slower = less risky<br>• High value (fast): Early rejection = quicker = riskier<br>In case of false results, this value can be decreased (more accurate). |
| Auto | Automatic setting |

### 9.2.4.5 Result offset tab

See Chapter: Result offset tab (Page 151)

### 9.2.4.6 Gripping space tab

See Chapter: Gripping space tab (Page 153)

## 9.3 Setup Detectors

Each job contains one or several inspection steps (detectors), which you can define here. By clicking on the "Detectors" button, or the "New" button under the Detector list, a window with a list of all available detectors opens. The corresponding setting ranges are displayed graphically in the image as a frame in preset position and size. Now the frames and the parameters can be adjusted according to the inspection task.

**For information on what the various frames in the image mean and how to adjust them, please refer to section:** Search and feature ranges.

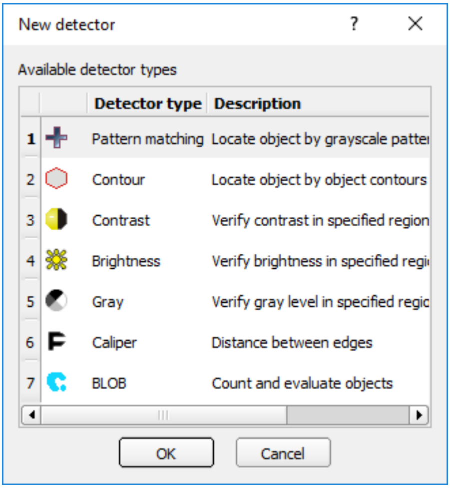



*Fig. 110: Detector selection list, example: Object sensor*

## 9.3.1 Creating and adjusting detectors

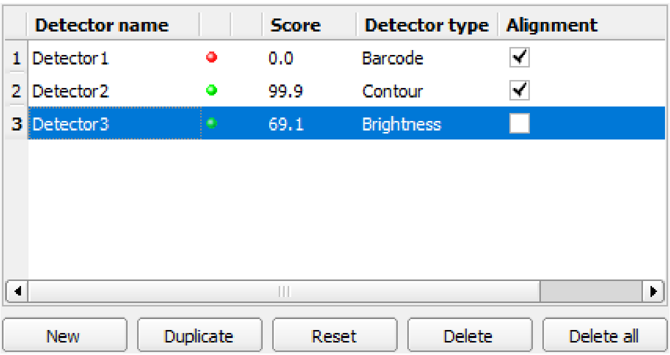



*Fig. 111: Detector list*

### Create new detector

1. Click on the button "New" under the detector list in the configuration window and select the type of detector required. A new detector entry appears in the detector list.
2. Edit the name of the detector by double clicking on the field "Name".

### Configuring the detector

1. Select a detector in the drop-down list and assign a name for each detector.
2. Define the corresponding Search and feature ranges (Page 352)graphically in the image.
3. Configure the detector by entering / setting parameters in the tabs in the configuration window to the right of the detector list. Which tabs are shown depends on the type of detector selected.

### Configure overlay

In the menu "View"/"Overlay settings..." or in the toolbar, the overlays in the image (frame in yellow, red, etc.) can be switched on or off for each detector or category.

In "View" / "Overlay current detector only" or with the frame symbol button, all overlays in the image can be switched off except for those of the currently processed detector.

 Overlay of current detector only

 Overlay of failed detectors only

### Functions for administration of detectors

| Button | Function |
| --- | --- |
| New | Adds new detector > Dialogue with detector drop-down list appears |
| Duplicate | Create a new detector by copying the existing detector with all settings (incl. search range etc.). |
| Copy (right click on Detector > "Detector Parameter copy...") | Copies all parameters from one detector to one or several others. All detectors must be from the same type. To copy the characteristic areas (different colored frames such as: Teach-in area, Search area, etc.), select the corresponding checkboxes.<br>Copy process:<br>Create all desired target detectors of the same type as the source detector. Mark source detector in the detector list.<br>Press to button "copy".<br>In the appearing list, mark all desired target detectors (multiple marking with a held "Ctrl" key). Click "Copy" to confirm. |
| Reset | Reset the parameters and the search and feature range of the selected detector to the default values |
| Delete | Deletes the selected detector |
| Delete all | Deletes all of the detectors in the list |

### Error display in the detector list

The following icons can be displayed in the third column of the detector list in the event of an error:

 = the computing time of a detector is longer than specified in the Cycle time tab, detector is NOK

 = no part is found during Alignment, all detectors dependent on it are NOK

 = Calibration has not been performed correctly, all following detectors are NOK

 = result buffer is exceeded (>10 MB)

### 9.3.2 Selecting a suitable detector

**The following detectors are available:**

| Detector Type | | Description |
|---|---|---|
| | Detector Pattern matching | Locate and count objects by patterns |
| | Detector Contour | Locate and count objects by contours |
| | Detector Contour 3D | Locating objects in space (3D) |
| | Detector Contrast | Check contrast |
| | Detector Brightness | Check brightness |
| | Detector Gray | Check grey value distribution |
| | Detector BLOB | Count and evaluate objects |
| | Detector Caliper | Measure distance between edges |
| | Detector Color Area | Check color value distribution |
| | Detector Color List | Assign objects to a color |
| | Detector Color Value | Check color values |
| | Detector Barcode | Read barcodes and evaluate quality |
| | Detector Datacode | Read datacodes and evaluate quality |
| | Detector OCR | Read plain text (OCR) |
| | Detector Result processing: Text, numbers | Process and evaluate detector results |
| | Detector Wafer | Locating and verifying of wafers |
| | Detector Busbar | Locating and verifying of busbars on wafers |

## 9.3.3 Detector Pattern matching

This detector is suitable for detecting patterns of any shape, even without clear edges or contours

### 9.3.3.1 Color Channel tab

See Chapter: Color Channel tab (Page 147)

### 9.3.3.2 Pattern matching tab

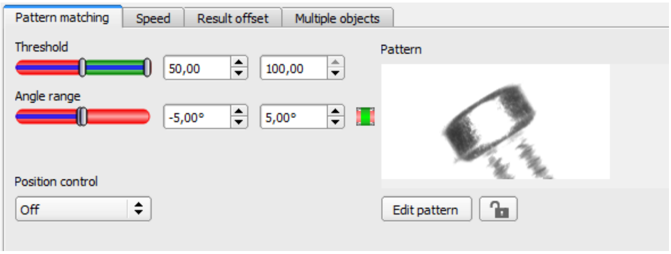



*Fig. 112: Pattern matching detector, Pattern matching tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Threshold | Range for the required match of the found pattern with the learned pattern in %. |
| Angle range | Angle range in which to search (larger range leads to longer processing times). Depending on the size and complexity of the image, the Angle range may be limited. |
| Position control | Checks whether the pattern found is in the right position.<br>If position check is activated, the position frame is shown in blue (either rectangular or elliptic). The center of the pattern must be within the blue frame. |
| Pattern | Shows the taught pattern (red frame in the field of view) |
| Edit pattern | With the parameter "Edit pattern", areas of the taught-in pattern can be hidden. As with an eraser, the areas that are not needed for the evaluation can be removed in the search range. The marked areas can also be inverted (see also Function: Edit pattern / contour (Page 181)) |

| Parameter | Function |
|---|---|
| Lock  | Lock / unlock pattern. When locked, the taught-in pattern is protected against (unintentional / accidental) changes, e.g. accidental adaptation of the teaching range. Unlock to modify taught pattern. |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

#### Improvement of execution speed

- Select the taught-in pattern (red frame) as small as possible.
- Search range for position (yellow frame) only as large as necessary.
  The search range indicates the area in which the center of gravity of the pattern is searched.
- Reduce resolution (e.g. to WGA)

**ATTENTION:**
This parameter affects all detectors!

- Set slider (accurate - fast) to "fast"

#### Robust detection

- Search range (yellow frame) sufficiently large?
- Set slider (accurate - fast) to "accurate"
- Select a distinctive gray value pattern; re-teach if necessary.
- If found in the wrong position: Use a unique pattern, and re-teach and adjust the threshold if necessary.

### 9.3.3.3 Speed tab

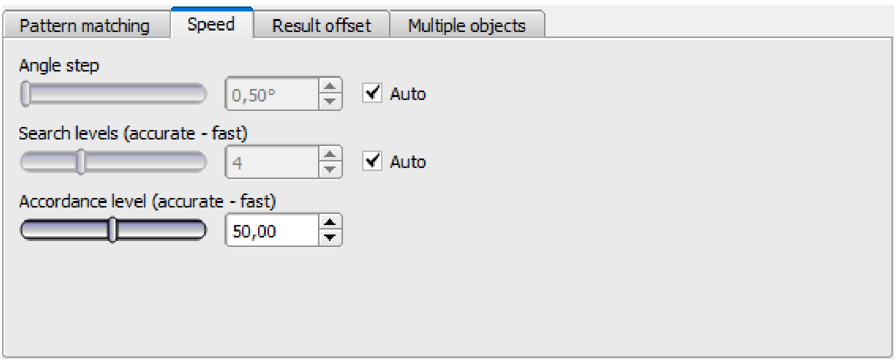



*Fig. 113: Detector Pattern matching, Speed tab*

The execution speed is influenced by the adjustable speed parameters. The search is performed either less finely, i.e. earlier canceled and thus faster, or even finer details are taken into account in the search, i.e. search longer and the search is slower.

If, immediately after teach-in, the found position (green ROI) does not coincide with the teach-in range (red ROI), the "Search levels (accurate - fast)" slider should be set to "fast".

**Parameter description:**

| Parameter | Function |
|---|---|
| Angle step | Sensitivity of the search throughout the selected angle range in degrees [°] |
| Search levels (accurate - fast) | Number of search levels (one search level corresponds to an image with half resolution)<br>• Small value (accurate): Slow search = lower risk (less likely to overlook candidates)<br>• High value (fast): Fast search = higher risk (candidates can be overlooked) |
| Accordance level (accurate - fast) | Candidates with a degree of compliance below the specified value are already discarded during the search.<br>• Small value (accurate): Late rejection = slower = less risky<br>• High value (fast): Early rejection = quicker = riskier<br>In case of false results, this value can be decreased (more accurate). |
| Auto | Automatic setting |

### 9.3.3.4 Result offset tab

See Chapter: Result offset tab (Page 151)

### 9.3.3.5 Multiple objects tab

By default, a maximum of one instance of the taught object is found in the image. The "Multiple objects" tab makes it possible to find several instances of a taught pattern. Multiple recognition identifies objects whose pattern matches the taught-in pattern. The output of the object results is sorted according to the set criteria in ascending or descending order.

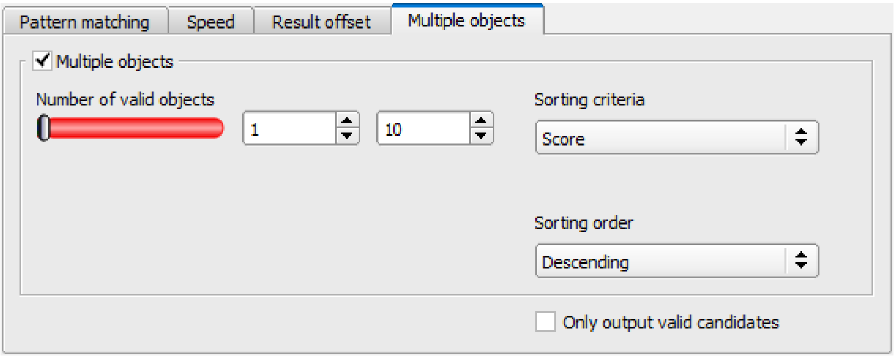



*Fig. 114: Detector Pattern matching, Multiple objects tab*

It is also possible to use this function to count objects in the image. The number of objects found can be output as a telegram. The minimum and maximum number of tolerated objects can be specified with the "Number of valid objects" parameter. If the number of objects found is outside this range, the detector result is NOK.

**Parameter description:**

| Parameter | Function |
|---|---|
| Number of valid objects | This parameter allows you to check whether the number of objects found is within a specified range. If it is within the range, the detector result is OK, otherwise it is NOK. |
| Sorting criterion<br>• Score<br>• Position X<br>• Position Y<br>• Angle Z | Sorting criteria according to which objects should be "pre-sorted".<br>• Overall result<br>• Position X<br>• Position Y<br>• Angle around Z |
| Sorting order<br>• Ascending<br>• Descending | Sorting order for the selected sorting criterion.<br>The values of the sorting criteria are sorted in ascending order.<br>The values of the sorting criteria are sorted in descending order. |
| Only output valid candidates | If this checkbox is activated, only objects whose score value is above the set threshold (tab "Pattern matching") are displayed and output. It can be used e.g. for parameter optimization. |

## 9.3.3.6 Pattern matching application

In this example, a contact (far left) of the test piece was taught in as a pattern and is recognized at this point even with a high degree of compliance (switching threshold near 100%).

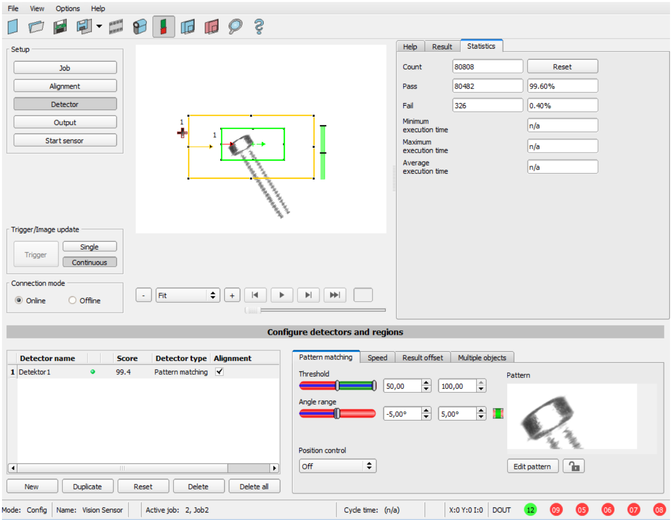



*Fig. 115: Pattern matching, application example, result OK*

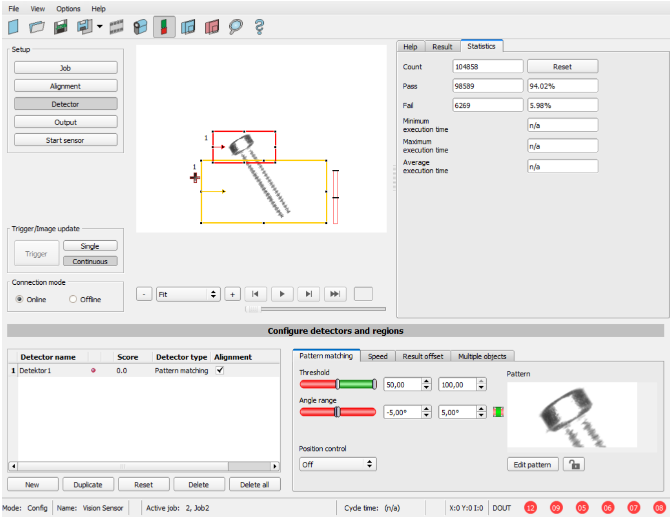



*Fig. 116: Pattern matching, application example, result NOK*

If the same Pattern matching is performed at a location of the test part where the contact sought is missing, the degree of conformity does not reach the required threshold value and the result becomes NOK. The contact is searched for here because of the gray values at the respective places in the image. Since the interior, very highly reflective and therefore bright area does not exist, and instead the image pixels correspondingly have darker values, the degree of conformity here is not as high as with an existing contact. However, because large parts of the pattern are identical to the one taught in (the entire outer, black area), the degree of conformity is still quite high, at around 70%!

The settings made here are chosen only to illustrate the operation of the Pattern matching detector. They should be further optimized in real operation (e.g. by reducing the search and feature range >> relevant pattern becomes more significant, etc.).

By teaching, the pattern inside the red frame is stored in the sensor as a reference. Size and position of the reference is defined by the red frame. In Run mode, the VISOR® will search the current image for the best reference image / pattern match inside the search region. Depending on the setting of the threshold value (= degree of conformity), the object is recognized as good or not.

**Example:**

The following pattern was taught:

*Fig. 117: Pattern, reference*

For the following three example images, the 100% match object is recognized because the taught pattern is exactly the same even though it is in a different location in the image. However, it is only moved in the X or Y direction and not rotated.

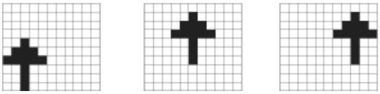


*Fig. 118: Pattern, result OK*

In the case of the three following example images, the object is also recognized, but with less than 100% match (about 70 - 80%), because in some pixels, it differs from the taught-in pattern. Depending on the setting of the threshold value (degree of match), good or bad results are returned.

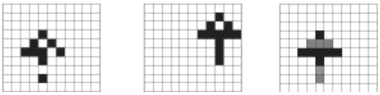


*Fig. 119: Pattern, borderline cases*

### 9.3.3.7 Function: Edit pattern / contour

With the function "Edit search range" or "Edit pattern" / "Edit contour", areas for the evaluation can be allowed or excluded within the search fields / feature fields of the various detectors.

#### Application example

Outer and inner contour lines, as well as holes should be irrelevant for the evaluation, but all surface defects should be detected.

After masking, only the unmarked areas within the ROI of the detector are used for the evaluation. The yellow marked areas are masked and thus no longer relevant for the evaluation.

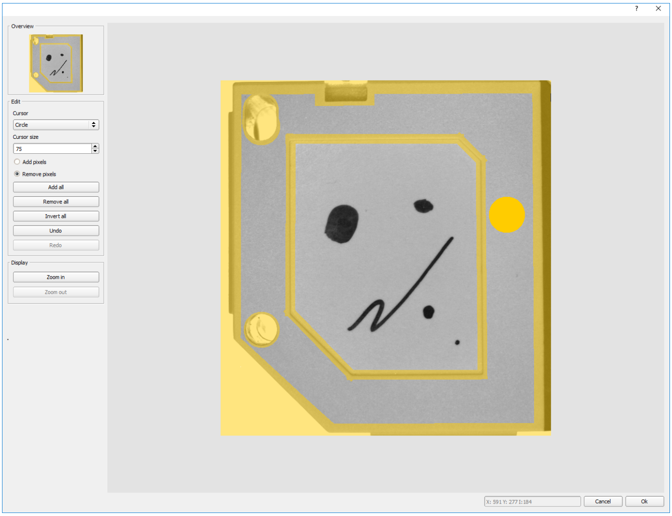


*Fig. 120: Edit region*

### Handling

| Parameter | Function |
|---|---|
| Cursor (shape) | Changing shape of the cursor (square, circle, or line) With the setting: "Cursor = Line", the angular position of the line jumps while the Shift key is pressed in 15 ° increments. |
| Cursor size | Changing the size of the cursor (possible setting 1-500, also e.g. with the mouse wheel) |
| Add / remove pixels | Select whether the cursor adds or excludes pixels for image pro-cessing |
| Add all | Adds all pixels to image processing |
| Removes all | Removes all pixels from image processing |
| Invert all | Inverts all pixels |
| Undo | Undoes the last action |
| Restore | Restores the last undo action |
| Display | Selects the display mode (zoom in / out) |

The flexible selection of cursor shape and size, as well as whether an action adds or removes pixels, can easily and quickly define complex geometrical or free-form areas that are relevant or not taken into account (yellow).

#### Settings for using the "Edit region" function for the different detector types

| Detector Type | Necessary setting for editing |
|---|---|
| Pattern matching, Contour | Generally possible with "Edit pattern" / "Edit contour" |
| Contrast, Brightness, Gray, BLOB, Color Value, Color Area, Color List | Select search range "Free shape" |

#### Masking of search and learn ranges, Examples

Adjustment of the search range to the object for detectors Contrast, Brightness, and Gray level

For the detectors mentioned above, there are three forms for the search area: Circle, Rectangle, and Free form If the work area cannot be adjusted well enough to the object with a circle or rectangle that can be rotated over the control point on the arrow, use the freeform search range. With this feature, any geometry can be designed for the search range. The cursor for editing the search range can be selected as a square or a circle of any size. Below are some examples of freeform search ranges with a brief description of how they were created in the freeform editor.

#### Example 1: Circles with relevant areas

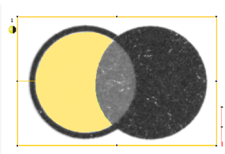


*Fig. 121: Edit region 1*

Generated by adding and a withdrawing a circle.

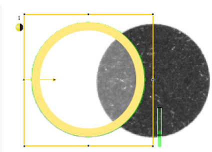


*Fig. 122: Edit region 2*

Generated by adding and a withdrawing a circle.

### Example 2: Only surface defects are relevant, object contour lines are masked

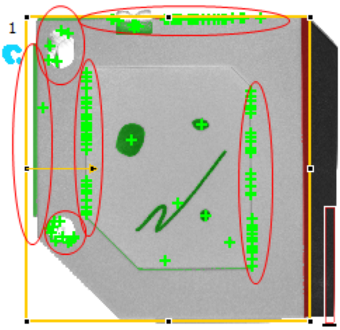


*Fig. 123: BLOB detector without the use of masking*

With the used BLOB detector, all surface defects as well as the outer and inner contour lines are detected.

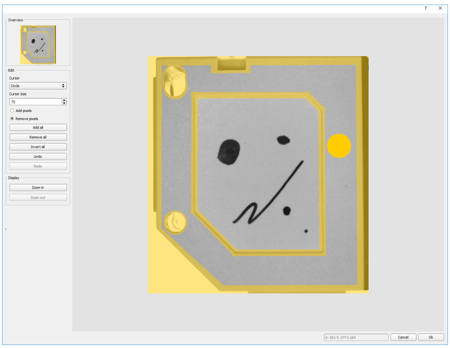


*Fig. 124: Masking the contour lines that are not to be recognized = yellow areas.*

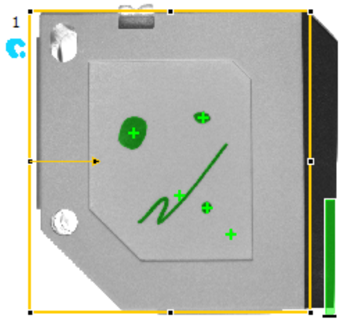



*Fig. 125: BLOB detector with the use of masking*

Only surface defects are detected; all contour lines / objects in the masking area are no longer recognized.

## 9.3.4 Detector Contour

This detector is suitable for the rotational position tolerant detection of contours on the basis of edges.

The contours of the object in the search range are taught and stored in the sensor. In Run mode, the sensor searches for the position of the best fit with the taught contour in the current image. If the fit is higher than the selected threshold, the result is considered positive. The contour recognition is completely rotational tolerant, i.e. the searched object may appear in any position in the image (choose angle setting accordingly!).

### 9.3.4.1 Color Channel tab

See Chapter: Color Channel tab (Page 147)

### 9.3.4.2 Contour tab

The most important parameters for contour detection can be set in the "Contour" tab.

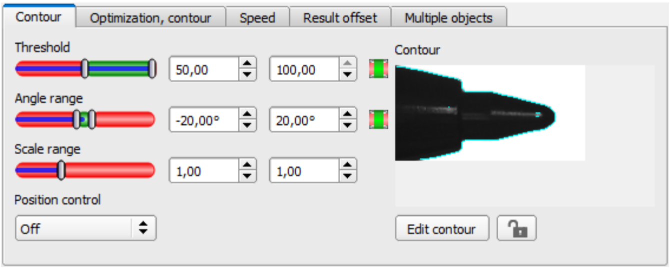



*Fig. 126: Detector Contour, Contour tab*

The edges marked in light blue at the bottom right (high-contrast transitions in the image) were identified and drawn in on the basis of the parameter settings made in the teach-in area (red frame). These can be further altered by changing the parameters or with the function "Edit contour". The VISOR® vision sensor will search for this contour only in the image within the search region (yellow frame).

**Parameter description:**

| Parameter | Function |
|---|---|
| Threshold | Value for the required match of the found contour with the taught-in contour |
| Angle range | Angle range in which to search (larger range leads to longer processing times). |
| Scaling | Detection also of enlarged or reduced objects in a given scale range. |
| Contour | Shows the taught-in contour |
| Edit contour | Via the parameter "Edit contour", ranges of the taught-in contour can be hidden. As with an eraser, the areas that are not needed for the evaluation can be removed in the search range. The setting can also be inverted via the option "Invert all".<br>Additional information: "Function: Edit pattern / contour (Page 181)" |
| Lock  | Locking / unlocking the contour: In the locked state, the taught-in contour is protected against (unintentional) change, e.g. accidentally adjusting the learning area. Unlock (click the lock icon again) for changing the contour. |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

**Improvement of execution speed**

- Search range for position (yellow frame) only as large as necessary.
  The search range indicates the area in which the center of gravity of the contour is searched.

- Search range for angles only as large as necessary.
- The search range for scaling is only as large as necessary
- Reduce resolution (e.g. to WGA)

**ATTENTION:**
This parameter affects all detectors!

- Set slider (accurate - fast) to "fast"
- Wert für „Min. Kontrast Modell“ erhöhen, da kleine Werte zu einer größeren Anzahl an extrahierten Konturen führen können. In the display of the pattern, check whether the relevant contours are still present.
- Wert für „Min. Kontrast Bild“ erhöhen.
- Especially in the case of Alignment: Use an alternate search pattern. For example, with higher contrast, so that "Min. Kontrast Modell" and "Min. Kontrast Bild" can be increased.

**Robust detection**

- Search range (yellow frame) sufficiently large?
- Search range for angles sufficiently large?
- Search range for scaling sufficiently large?
- Min. Kontrast Modell and Min. Kontrast Bild sufficient? Is the contrast sufficient when acquiring the model and the images that are to be checked? Is the model detectable in the images that are to be checked?
- Set slider (accurate - fast) to "accurate"
- Objects should not overlap.
- Are there distinctive edges in the model? If necessary, teach in the model again so that prominent edges lie in the taught-in model.
- Is "Min. Kontrast Modell" suitably selected? If the relevant contours are not displayed in the taught-in model, reduce "Min. Kontrast Modell". If too many contour lines are shown, increase "Min. Kontrast Modell".
- Is "Min. Kontrast Bild" suitably chosen for the current image? If the current image has a smaller / larger contrast than the taught-in pattern, "Min. Kontrast Bild" should be smaller / larger than "Min. Kontrast Modell".
- Model found in wrong position? If the taught-in model is not unique, teach-in a new model.
- Does the result value fluctuate from image to image? If necessary, ensure that no "wrong" contours are taught in the image (edges due to shadows or contour fragments that are not desired in the contour model). This can be achieved by increasing "Min. Kontrast Modell". With the help of "Edit contour", search ranges can be hidden.

**Parameter angle range: Rotational direction of angle**

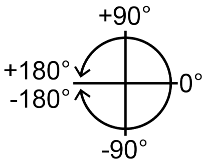



*Fig. 127: Rotational direction of angle*

### 9.3.4.3 Contour optimization tab

In the "Contour optimization" tab, further settings for the edge transition and the contrast can be made.

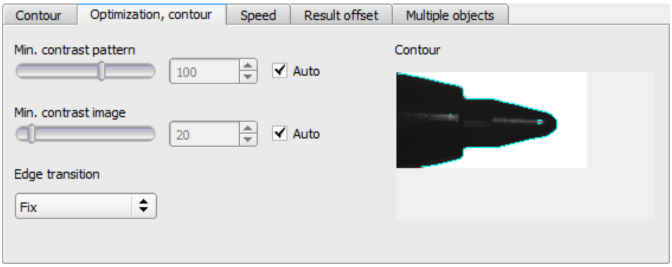



*Fig. 128: Detector Contour, Contour optimization tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Min. Kontrast Modell | Minimum required contrast for the taught-in model, in which an edge is accepted as such. |
| Min. Kontrast Bild | Minimum contrast required in current image for an edge to be accep-ted as one. |
| Edge transition<br>• fix<br>• fix + inverted<br>• flexible | The parameter "Edge transition" can be used to determine the trans-ition between object or contour and background. The way in which the contour is to be recognized is selected:<br>• "fix" = only on the taught-in background<br>• "fix + inverted" = only on the taught-in and inverted background<br>• "flexible" = on any background<br>Additional information: see below |
| Auto | Automatic setting |

### Edge transition

Example:

A gray object is taught-in in front of a brighter background, as shown in the following figure.

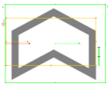
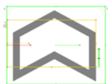
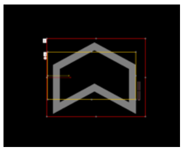
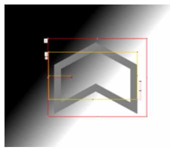
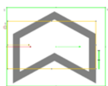
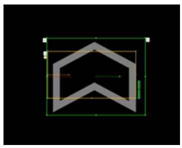
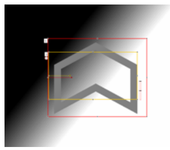
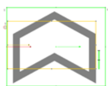
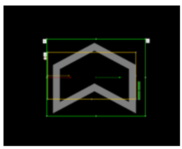
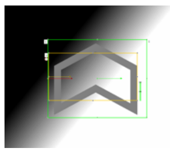


*Fig. 129: Taught-in contour in front of a bright background*

The following table shows how the contour detector behaves with the respective edge transition setting.

| Settings for parameter "Edge transition" | Bright background | Dark background | Inconsistent background |
| --- | --- | --- | --- |
| fix |  |  |  |
| | Contour detector: OK | Contour detector: NOK | Contour detector: NOK |
| fix + inverted |  |  |  |
| | Contour detector: OK | Contour detector: OK | Contour detector: NOK |
| flexible |  |  |  |
| | Contour detector: OK | Contour detector: OK | Contour detector: OK |

### 9.3.4.4 Speed tab

The execution speed is influenced by the adjustable speed parameters. The search is performed either less finely, i.e. earlier canceled and thus faster, or even finer details are taken into account in the search, i.e. search longer and the search is slower.

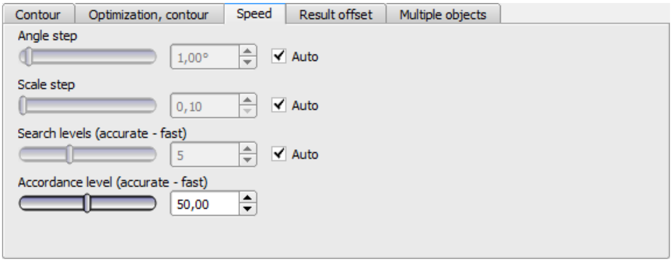



*Fig. 130: Detector Contour, Speed tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Angle step | Sensitivity of the search throughout the selected angle range in degrees [°] |
| Scale step | Sensitivity of the search throughout the selected scale range |
| Search levels (accurate - fast) | Number of search levels (one search level corresponds to an image with half resolution)<br>• Small value (accurate): Slow search = lower risk (less likely to overlook candidates)<br>• High value (fast): Fast search = higher risk (candidates can be overlooked) |
| Accordance level (accurate - fast) | Candidates with a degree of compliance below the specified value are already discarded during the search.<br>• Small value (accurate): Late rejection = slower = less risky<br>• High value (fast): Early rejection = quicker = riskier<br>In case of false results, this value can be decreased (more accurate). |
| Auto | Automatic setting |

### 9.3.4.5 Result offset tab

See Chapter: Result offset tab (Page 151)

### 9.3.4.6 Multiple objects tab

By default, a maximum of one instance of the taught object is found in the image. The "Multiple objects" tab makes it possible to find several instances of a taught contour. Multiple recognition identifies objects whose contours match the taught-in contour. The output of the object results is sorted according to the set criteria in ascending or descending order.

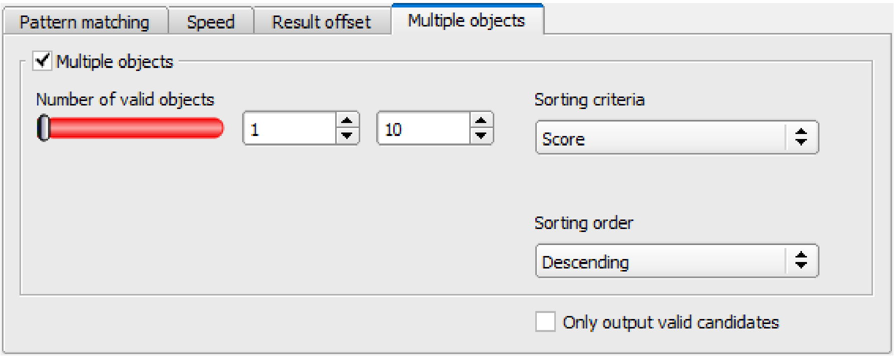



*Fig. 131: Contour Detector, Multiple objects tab*

It is also possible to use this function to count objects in the image. The number of objects found can be output as a telegram. The minimum and maximum number of tolerated objects can be specified with the "Number of valid objects" parameter. If the number of objects found is outside this range, the detector result is NOK.

**Parameter description:**

| Parameter | Function |
|---|---|
| Number of valid objects | This parameter allows you to check whether the number of objects found is within a specified range. If it is within the range, the detector result is OK, otherwise it is NOK. |
| Sorting criterion<br>• Score<br>• Position X<br>• Position Y<br>• Angle Z<br>• Scaling | Sorting criteria according to which objects should be "pre-sorted".<br>• Overall result<br>• Position X<br>• Position Y<br>• Angle Z<br>• Scaling |
| Sorting order<br>• Ascending | Sorting order for the selected sorting criterion.<br>The values of the sorting criteria are sorted in ascending order. |

| Parameter | Function |
|---|---|
| • Descending | The values of the sorting criteria are sorted in descending order. |
| Only output valid candidates | If this checkbox is activated, only objects whose score value is above the set threshold ("Contour" tab) are displayed and output. It can be used e.g. for parameter optimization. |

### 9.3.5 Detector Contour 3D

This detector allows the localization of objects in space in all six degrees of freedom (position X, Y, Z and Angle X, Y, Z) with the acquisition of a single image.

To teach-in the Contour 3D detector, the contours used must lie in **one** plane. The default contour plane is the Measurement plane that was defined during calibration (Measurement plane (Page 192)). However, the contour plane can also be adjusted in the "Contour plane" tab using the calibration plate (Contour plane (Page 192)).

**NOTE:**
For reliable detection, the use of the Contour 3D detector is recommended for printed objects.

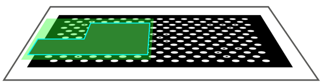


*Fig. 132: Measurement plane*

Contour plane (green) is parallel to the Measurement plane from calibration. The Measurement plane can be used as the contour plane via Z offset.

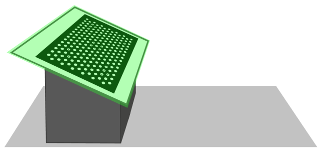


*Fig. 133: Contour plane*

The contour plane (green) is not parallel to the Measurement plane defined in the calibration (grey). The contour plane must be taught-in separately by taking an image of a calibration plate.

**NOTE:**

- To use the Contour 3D detector, a calibration must first be performed (calibration methods: Calibration plate (Robotics), Hand-Eye calibration (Robotics) or Base-Eye calibration (Robotics)).
- If no calibration is defined, no contour can be taught in.

For reliable detection, the object or the desired contours should not be perpendicular to the optical axis, but slightly tilted. This causes the object to appear distorted in perspective when the image is captured. The more distinctive the perspective distortion is, the better a clear pose can be assigned to this distortion.

### 9.3.5.1 Color Channel tab

See Chapter: Color Channel tab (Page 147)

### 9.3.5.2 Contour tab

The most important parameters for contour detection can be set in the "Contour" tab.

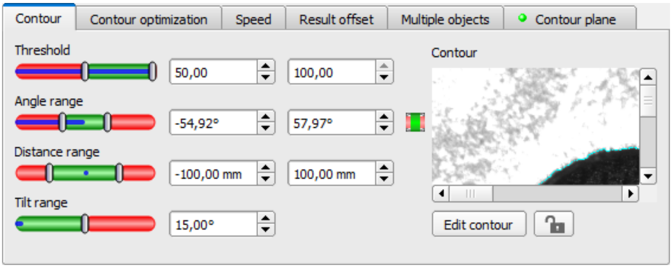



*Fig. 134: Contour 3D detector, Contour tab*

The edges in the lower right corner marked in light blue (high-contrast transitions in the image) were identified and drawn on the basis of the parameter settings made. These can be further altered by changing the parameters or with the function "Edit contour". The contours used here must lie in one plane to obtain correct localization results. The VISOR® vision sensor now searches for this contour in the image within the search range (yellow frame).

**Parameter description:**

| Parameter | Function |
|---|---|
| Threshold | Value for the required match of the found contour with the taught-in contour |
| Angle range | Angle range in which to search (larger range leads to longer processing times). Depending on the size and complexity of the image, the Angle range may be limited. |
| Distance range | Depth range in mm in which the search is carried out, starting from the contour plane. |
| Tilt range | Maximum tilt of the object in relation to the contour plane, i.e. the maximum rotation around the X and Y axes of the contour plane. |
| Contour | Shows the taught-in contour |

| Parameter | Function |
|---|---|
| Edit contour | Via the parameter "Edit contour", ranges of the taught-in contour can be hidden. As with an eraser, the areas that are not needed for the evaluation can be removed in the search range. The setting can also be inverted via the option "Invert all".<br>Additional information: "Function: Edit pattern / contour (Page 181)" |
| Lock  | Locking / unlocking the contour: In the locked state, the taught-in contour is protected against (unintentional) change, e.g. accidentally adjusting the learning area. Unlock (click the lock icon again) for changing the contour. |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

### Improvement of execution speed

- Reduce resolution (e.g. to WGA) (→Setup step Job)

**ATTENTION:**
This parameter affects all detectors!

- Remove all unneeded areas around the taught-in contour with the "Edit contour" function.
- Search range for position (yellow frame) only as large as necessary.
  The search range indicates the area in which the center of gravity of the contour is searched.
- Wert für „Min. Kontrast Modell“ erhöhen, da kleine Werte zu einer größeren Anzahl an extrahierten Konturen führen können. In the display of the pattern, check whether the relevant contours are still present. (→ Contour optimization tab)
- Wert für „Min. Kontrast Bild“ erhöhen.
- Search levels / Accordance level: Set slider (accurate - fast) to "fast" (→ Speed tab)
- Especially in the case of Alignment: Use an alternate search pattern. For example, with higher contrast, so that "Min. Kontrast Modell" and "Min. Kontrast Bild" can be increased.

### Robust detection

- The contours to be taught in must all be in the same plane. If necessary, use "Edit contour" to remove contours that do not lie in the contour plane.
- Set slider (accurate - fast) to "accurate". However, this leads to an increased cycle time.
- The object or the desired contours should not be perpendicular to the optical axis, but slightly tilted so that the object appears distorted in perspective when the image is taken. This increases the accuracy.
- The taught-in model should have as few symmetries as possible, since the results are not clear for symmetrical contours (example: teach-in of a circle: angle of rotation is undefined;

teach-in of an ellipse: angle of rotation is undefined at 180°). If necessary, add further contour features to create an asymmetry.

- The contours used must not be caused by reflections, as these change their shape and position when the objects are tilted. Use illumination that is as homogeneous as possible.
- Objects should not overlap.
- The objects to be found should appear as large as possible in the image.
- The taught-in contour should have as many features / points as possible (which of course must correspond to real contours / points of the object).

**Parameter angle range: Rotational direction of angle**

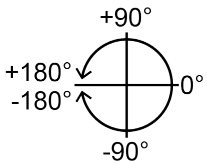



*Fig. 135: Rotational direction of angle*

### 9.3.5.3 Contour optimization tab

In the "Contour optimization" tab, further settings for the edge transition and the contrast can be made.

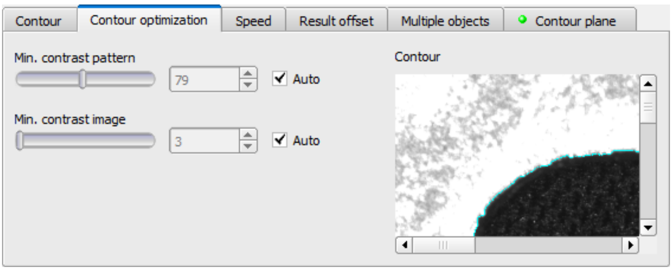



*Fig. 136: Detector Contour 3D, Contour optimization tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Min. Kontrast Modell | Minimum required contrast for the taught-in model, in which an edge is accepted as such. |

| Parameter | Function |
|---|---|
| Min. Kontrast Bild | Minimum contrast required in current image for an edge to be accepted as one. |
| Edge transition<br>• fix<br>• fix + inverted | The parameter "Edge transition" can be used to determine the transition between object or contour and background. The way in which the contour is to be recognized is selected:<br>• "fix" = only on the taught-in background<br>• "fix + inverted" = only on the taught-in and inverted background<br>Additional information: see below |
| Auto | Automatic setting |

**Additional information: see description** Edge transition Detector Contour

### 9.3.5.4 Speed tab

The execution speed is influenced by the adjustable speed parameters. The search is performed either less finely, i.e. earlier canceled and thus faster, or even finer details are taken into account in the search, i.e. search longer and the search is slower. For further information on how to influence the speed, see also: Contour tab.

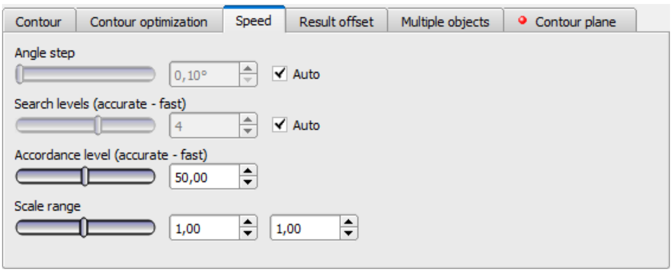



*Fig. 137: Contour 3D detector, Speed tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Angle step | Sensitivity of the search throughout the selected angle range in degrees [°] |

| Parameter | Function |
|---|---|
| Search levels (accurate - fast) | Number of search levels (one search level corresponds to an image with half resolution)<br>• Small value (accurate): Slow search = lower risk (less likely to overlook candidates)<br>• High value (fast): Fast search = higher risk (candidates can be overlooked) |
| Accordance level (accurate - fast) | Candidates with a degree of compliance below the specified value are already discarded during the search.<br>• Small value (accurate): Late rejection = slower = less risky<br>• High value (fast): Early rejection = quicker = riskier<br>In case of false results, this value can be decreased (more accurate). |
| Auto | Automatic setting |

### 9.3.5.5 Result offset tab

See Chapter: Result offset tab (Page 151)

### 9.3.5.6 Multiple objects tab

By default, a maximum of one instance of the taught object is found in the image. The "Multiple objects" tab makes it possible to find several instances of a taught contour.

Multiple recognition identifies objects whose contours match the taught-in contour. The output of the object results is sorted according to the set criteria in ascending or descending order.

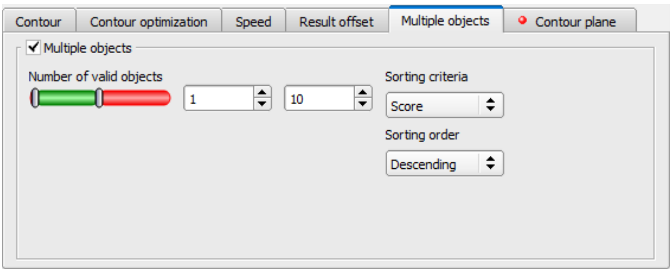



*Fig. 138: Detector Contour 3D, Multiple objects tab*

It is also possible to use this function to count objects in the image. The number of objects found can be output as a telegram. The minimum and maximum number of tolerated objects can be specified with the "Number of valid objects" parameter. If the number of objects found is outside this range, the detector result is NOK.

**Parameter description:**

| Parameter | Function |
|---|---|
| Number of valid objects | This parameter allows you to check whether the number of objects found is within a specified range. If it is within the range, the detector result is OK, otherwise it is NOK.<br>Note: By default, only objects whose score value is above the set threshold are displayed and output. |
| Sorting criterion<br>• Score<br>• Position X<br>• Position Y<br>• Position Z<br>• Angle X<br>• Angle Y<br>• Angle Z | Sorting criteria according to which objects should be "pre-sorted".<br>• Overall result<br>• X-position<br>• Y-position<br>• Z-position<br>• Angle around X<br>• Angle around Y<br>• Angle around Z |
| Sorting order<br>• Ascending<br>• Descending | Sorting order for the selected sorting criterion.<br>The values of the sorting criteria are sorted in ascending order.<br>The values of the sorting criteria are sorted in descending order. |

### 9.3.5.7 Contour plane tab

The Contour 3D detector requires knowledge of the plane in which the taught-in contours lie (contour plane). In the Contour plane tab, a contour plane can be taught in that differs from the Measurement plane. By default, the contour plane corresponds to the Measurement plane defined during calibration. However, by teaching in a calibration plate, the contour plane can be transformed in the X, Y and Z directions and rotated around the X, Y and Z axes.

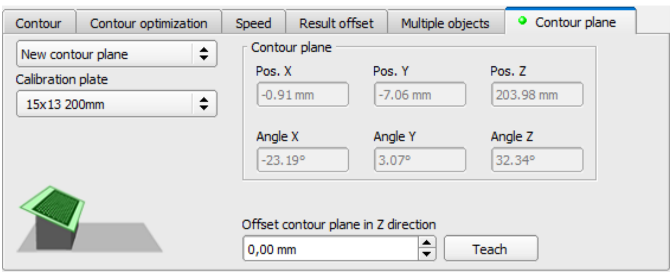



*Fig. 139: Contour 3D detector, Contour plane tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Selection of "Use measurement plane" | By default, the contour plane corresponds to the Measurement plane defined during calibration. |
| Selection of "New contour plane" | • Selection of the calibration plate (from the drop-down list) to be used for the definition of the contour plane<br>• "Offset contour plane in Z direction": relative to the contour plane, perpendicular to it<br>• "Teach" button: Contour plane is taught in. |
| Contour plane | Display (read-only) of X, Y and Z position and X, Y and Z rotation of the contour plane |

## 9.3.6 Detector Contrast

This detector determines the contrast in the selected search field.

For this purpose, all pixels within the search range are evaluated with their gray values, and the contrast is calculated. If the contrast value is inside the limits set under "Threshold", the result is positive (OK). The position of the light or dark pixels is not relevant here. It only depends on the spreading of light and dark pixels and their quantity ratio. Highest contrast value with 50% gray value "0" (= black) and 50% gray value "255" (= white).

### 9.3.6.1 Color Channel tab

See Chapter: Color Channel tab (Page 147)

### 9.3.6.2 Contrast tab

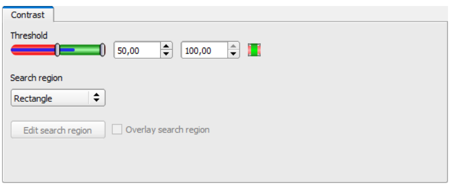



*Fig. 140: Contrast detector, Contrast tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Threshold | Specification of the contrast range that is accepted |

| Parameter | Function |
|---|---|
| Search range (shape) | The shape of the search range can be set as Rectangle, Circle, or Free shape. If Freeform is selected, "Edit search range" is active. |
| Edit search range | With the parameter "Edit search range", areas of the search range can be hidden. As with an eraser, the areas that are not needed for the evaluation can be removed in the search range. The marked areas can also be inverted. This marks the areas that are important for the execution. See Chapter: Function: Edit pattern / contour |
| Display search range | Enable / disable the display of search range edits |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

### 9.3.6.3 Contrast application

In the example, the presence of a contact is checked by means of a contrast detector.

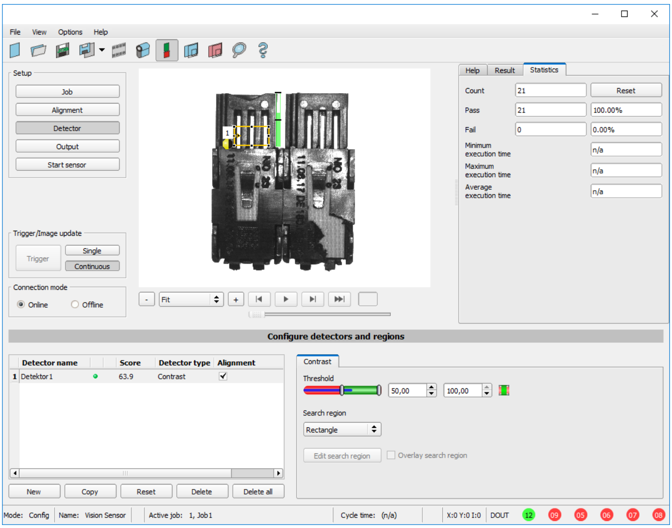



*Fig. 141: Contrast, application example, result OK*

The highly reflective, i.e. bright metallic contact, in the midst of the surrounding black plastic housing, is checked for presence with a contrast detector. Since the contrast is very high in this range, the detector provides a high value and thus in conjunction with Alignment a reliable result.

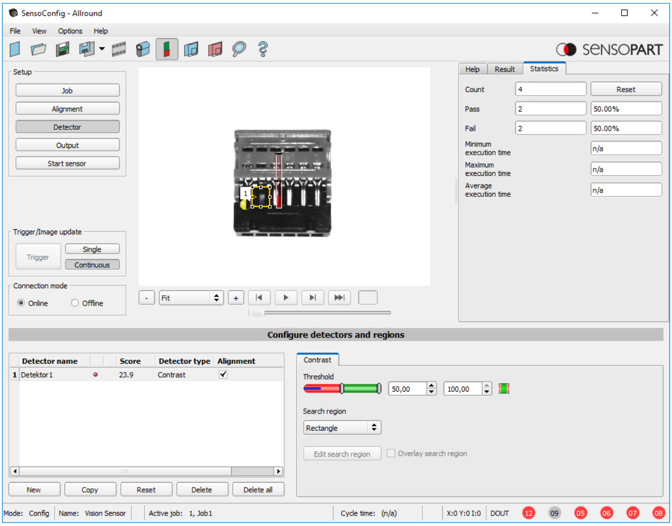



*Fig. 142: Contrast, application example, result NOK*

If the same detector is used at a location where contact is missing, the detector will give a negative result. This is because the contrast is too low between the black environment and the now visible black background of the contact.

### Function detector Contrast

The dark and light pixels are scored for number and light / dark intensity.

The position of the bright or dark pixels in the search range is irrelevant.

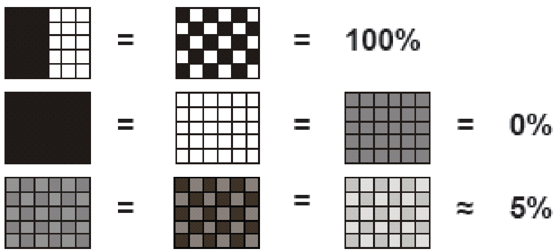



*Fig. 143: Contrast examples*

| | Evaluation method |
|---|---|
| Pattern | **Contrast** Bar graph |
| | <10% |
| | >90% |
| | <10% |

*Fig. 144: Contrast explanation*

## 9.3.7 Detector Gray

This detector is suitable for determining the gray values in the selected search field.

In the first step of this detector, the parameter "Graylevel" is used to determine the value range of the gray values that may occur in the search range. In the second step, under "Threshold", the area percentage (in %) of the search range is defined which must have the gray values defined in Step 1 in order to provide a positive result.

### 9.3.7.1 Color Channel tab

See Chapter: Color Channel tab (Page 147)

### 9.3.7.2 Gray tab

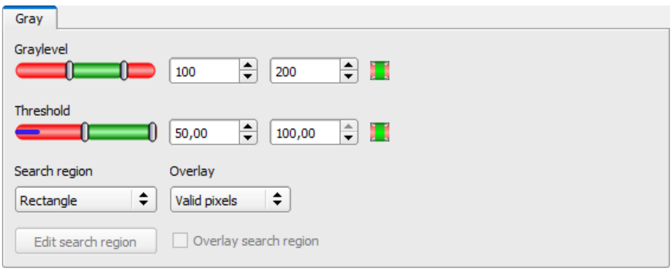



*Fig. 145: Gray detector, Gray tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Graylevel | Value range of the gray values that may occur in the search range |
| Threshold | Percentage of the area that must have the gray values defined under "Graylevel" |
| Invert button / | With the respective inversion of "Graylevel" or "Threshold", all conceivable combinations can be set, even those in which e.g. only gray values are allowed at the top and bottom of the value range. The position of the light or dark pixels is not relevant here. |
| Search range (shape) | The shape of the search range can be set as Rectangle, Circle, or Free shape. If Freeform is selected, "Edit search range" is active. |
| Overlay | Selection of the pixels that have a gray value within (valid pixels) or outside (invalid pixels) of the gray level definition under "Graylevel". These are then marked in color as a selection aid. Thus, e.g. noise pixels / areas that are not covered by the gray value range can very easily be detected. |
| Edit search range | With the parameter "Edit search range", areas of the search range can be hidden. As with an eraser, the areas that are not needed for the evaluation can be removed in the search range. The marked areas can also be inverted. This marks the areas that are important for the execution. See Chapter: Function: Edit pattern / contour |
| Display search range | Enable / disable the display of search range edits |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

### 9.3.7.3 Gray application

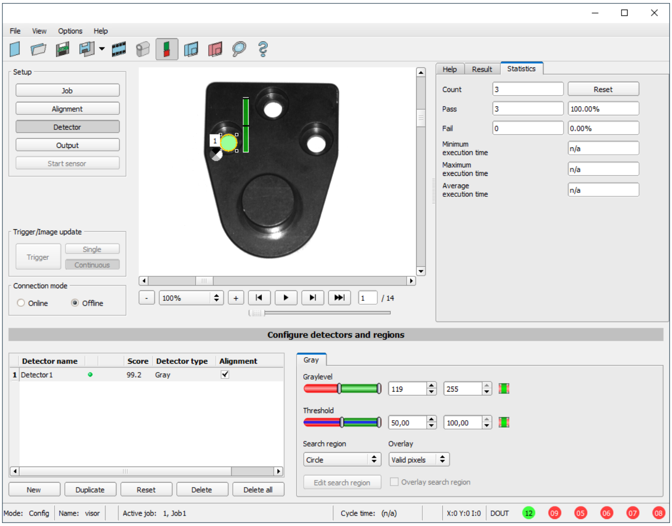



*Fig. 146: Gray, application example, positive result*

Requested characteristic is present in the search range, gray values within the required threshold values = positive result

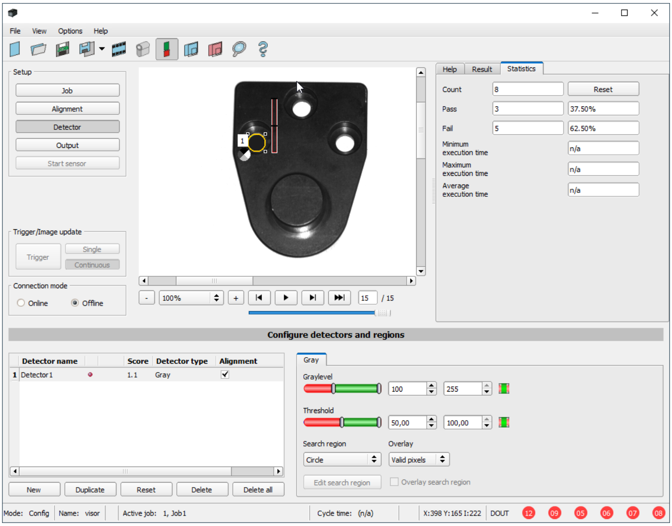



*Fig. 147: Gray, application example, negative result*

Searched characteristic is not present in the search range. I.e. mean value of the gray values in the search range not within the threshold values = negative result.

### Note for determining the gray values

If the cursor is placed anywhere in the image area, the corresponding X and Y coordinates as well as the gray value ("I" = intensity) are displayed in the second field from the right in the status line at the lower edge of the image.

### Detector Gray functionality:

The permissible gray value range is defined with the two limits of the "Graylevel" slider.

All pixels within this gray value range and within the defined working zone (yellow frame) are added together. The proportion of the number of all the pixels in the working zone (yellow frame) and of the number of pixels in the authorized gray value range represents the result of this detector.

If this result is within the limits set on the "Threshold" slider, the result is positive.

The position of the gray value pixels on the screen is of no importance.

**Example:** (when the gray level slider is set to very dark values):

The two images deliver exactly the same result with detector Gray, because 9 of 25 pixels are recognized as dark.

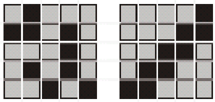


*Fig. 148: Graylevel, Example 1*

If the threshold value were set to 10 in this example, the following images would produce a positive result:

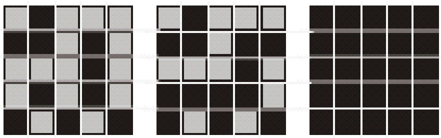


*Fig. 149: Graylevel, Example 2*

## 9.3.8 Detector Brightness

 The Brightness detector calculates the mean of the gray values of all pixels within the search range.

The two threshold value sliders of the "Threshold" parameter are used to set the permissible range for this brightness average. As soon as the calculated mean is within these two limits, the result is positive. The result is standardized to %. The position of the bright or dark pixels in the search range is irrelevant.

The detector can be used effectively if the position of the searched object in the image is absolutely unchanged from inspection to inspection. If deviations in the position may occur, Alignment must be used.

### 9.3.8.1 Color Channel tab

See Chapter: Color Channel tab (Page 147)

### 9.3.8.2 Brightness tab

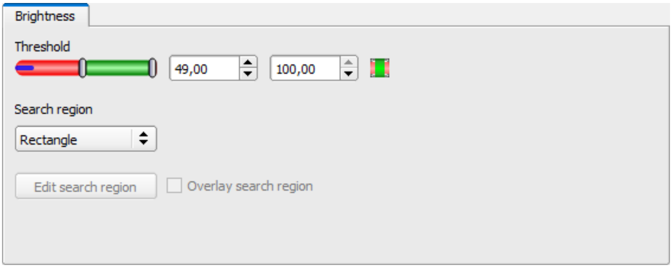



*Fig. 150: Brightness detector, Brightness tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Threshold | Specification of the brightness range that is accepted |
| Search range (shape) | The shape of the search range can be set as Rectangle, Circle, or Free shape. If Freeform is selected, "Edit search range" is active. |
| Edit search range | With the parameter "Edit search range", areas of the search range can be hidden. As with an eraser, the areas that are not needed for the evaluation can be removed in the search range. The marked areas can also be inverted. This marks the areas that are important for the execution. See Chapter: Function: Edit pattern / contour |
| Display search range | Enable / disable the display of search range edits |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

### 9.3.8.3 Brightness application

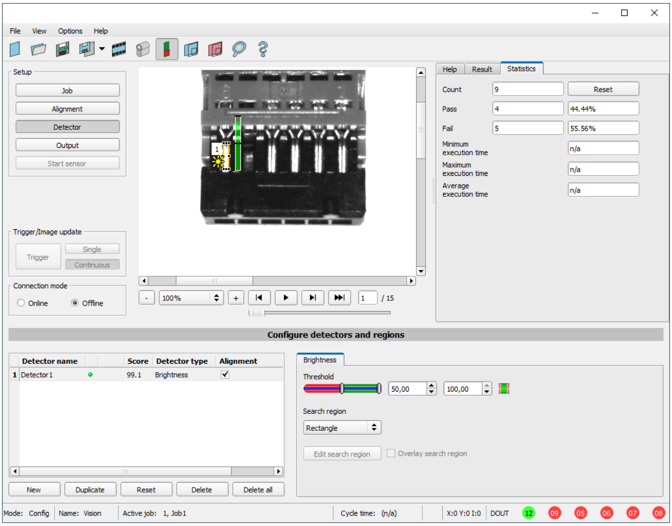


*Fig. 151: Brightness, application example, result OK*

The contact is present at the searched location and the mean value of the gray values in the search range therefore provides a very high value (close to 100%). This means the current value is within the requested threshold limits and the result is positive = contact present.

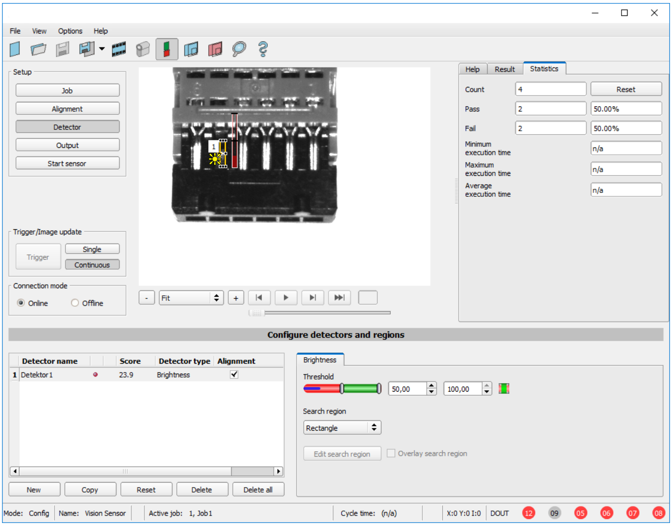



*Fig. 152: Brightness, application example, result NOK*

Contact does not exist at the requested location, and the mean of the gray values in the search area therefore provides a very low value (close to 0%). This means the current value is not within the requested threshold limits and the result is negative = contact not available.

**Examples: Brightness value as mean of the gray values.**

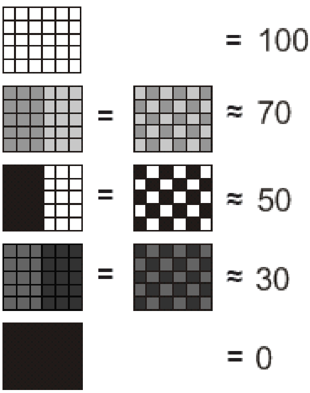



*Fig. 153: Brightness, examples*

## 9.3.9 Detector BLOB

The BLOB detector is used to identify and count one or more objects with some common features such as the same gray value range, same area, and same circumference.

The BLOB detector is used to identify and count one or more objects with some common features such as the same gray value range, same area, and same circumference.

- "BLOB", abbreviation for "Binary Large Object" or "Binary Labeled Object".
- Basic image processing function for evaluating **contiguous** surfaces and objects in an image.
- Differentiation of the individual objects on the basis of simple features, such as area, width, height, etc.

Binarized | Detected as BLOB / object

*Fig. 154: Screws*

**Typical applications**

- Counting of objects
- Differentiation / classification of objects in the image by:
  - Size, area, contour
  - Shape, Geometry
  - Position, Orientation
- Position, side
- Surface inspection

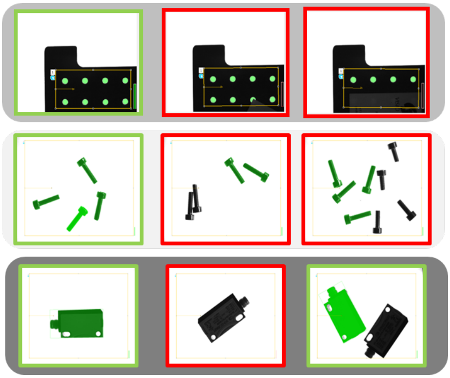
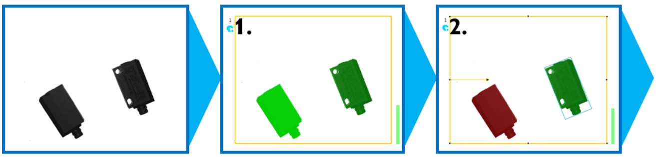


*Fig. 155: Typical applications: Count, classify / sort, location / side*

## BLOB, simple configuration in 3 steps




**3.**
Configuration and data output, Ethernet, M12, 4-pin
- Ethernet
- EtherNet/IP
- PROFINET
- FTP/SMB Archiving

### 1. Binarization

Separation between background and relevant object

Absolute threshold

Dynamic threshold

### 2. Filtering of detected BLOBs

Based on various characteristics such as: area, perimeter, orientation, location, etc.

Features tab

### 3. Data output

Definition of data output telegram and sorting of results.

Sorting tab

Telegram tab

### 9.3.9.1 Color Channel tab

See Chapter: Color Channel tab (Page 147)

### 9.3.9.2 Binarization tab

In this tab, all parameters for binarization of a BLOB can be set.

Binarization is the first step in BLOB evaluation and is used to separate relevant objects from the background of the image. This is done by converting the gray level image to a pure black and white image, i.e. binary image.

Two methods of binarization are available: "Absolute threshold" and "Dynamic threshold".


*Fig. 156: Detector BLOB, tab Binarization*

**Selection of binarization method**

| Parameter | Function |
|---|---|
| Absolute threshold | The binarization switching threshold is set to an absolute gray value in the value range of 0 ... 255. Additional information: Absolute threshold |
| Dynamic threshold | The dynamic switching threshold is automatically adjusted for each image to a position optimized according to statistical methods in order to be able to differentiate between foreground and background as well as possible. Additional information: Dynamic threshold |

**Boundary objects, overlays, and histogram**

| Parameter | Function |
|---|---|
| Boundary objects | If the "Boundary BLOBs" checkbox is activated, the selected BLOBs / objects are taken into account, even if they are not completely within the yellow search range.<br>Please note: BLOBs are also considered to be boundary BLOBs when they touch or are partially covered by an area that has been masked with "Edit pattern" / "Edit search range" (even if the masked / hidden area is inside the search range).<br>Additional information: Boundary objects |
| Search range | The shape of the search range can be set to: "Rectangle", "Circle" or "Free shape". In the mode "Free shape", the masking function "Edit search range" can be used to hide areas that are not relevant. |
| Edit search range | By clicking on "Edit search range", the window opens for editing the search range. Additional information: Function: Edit pattern / contour |
| Overlay | "Valid BLOBs": All valid BLOBs that meet the feature criteria within the set feature thresholds are marked green. Invalid BLOBs are marked red.<br>"BLOB contour": All valid BLOBs which fulfill the feature criteria within the set feature thresholds are marked with a green contour line. Invalid BLOBs are not marked. |
| Histogram | Clicking on "Histogram" opens the Histogram window.<br>Additional information: Histogram |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

### 9.3.9.2.1 Absolute threshold

**Parameter description:**

| Parameter | Function |
|---|---|
| Absolute threshold | The upper and lower limit defines the range of valid gray values of pixels assigned to the respective BLOB. |
| Gray value range | Setting the upper and lower limit of the gray values for binarization. |
| Invert button  /  | With the "Invert button" (default: red/green/red), the logic for the evaluation can be inverted. This way, the relevant range can be included or excluded. |

| Parameter | Function |
|---|---|
| Pipette symbol | By clicking on the Pipette button, the cursor changes into a pipette symbol. If the cursor is now moved and a pixel within the image is clicked, the thresholds of the "Absolute switching threshold" are set to +/- 10 gray values above or below the gray value of the selected pixel (maximum value range 0 ... 255). |

Additional information:

Histogram (Page 215)

### 9.3.9.2.2 Dynamic threshold

The dynamic switching threshold can be used if the searched BLOBs / objects have significantly different gray values than the background, and the brightness / illumination fluctuates evenly over the entire image.

If the image brightness changes evenly across the entire image, the two switching thresholds are automatically readjusted for each image. (When using "Absolute threshold", the thresholds ought to be readjusted manually!)

**Please note:**

- When using the dynamic switching threshold, the thresholds are recalculated and readjusted with each new image / evaluation.
- Changing lighting conditions or surface conditions / reflectivity can affect the result.

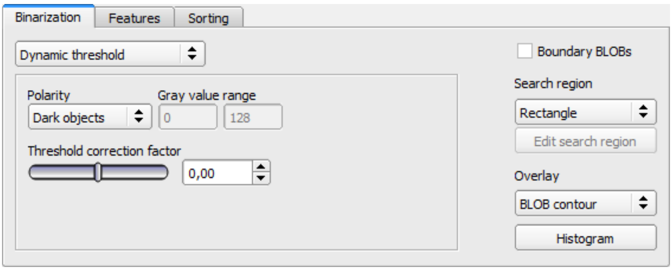



*Fig. 157: BLOB detector, Binarization tab, Dynamic threshold*

**Parameter description:**

| Parameter | Function |
|---|---|
| Dynamic threshold | The switching threshold is automatically adjusted for each image to a position optimized according to statistical methods in order to be able to distinguish as well as possible between foreground and background. |

| Parameter | Function |
|---|---|
| Polarity | Defined whether BLOB / object is brighter or darker than the background. |
| Gray value range | Gray value thresholds for binarization |
| Correction factor for threshold | The binarization threshold value can be shifted in the direction of the foreground or the background brightness via this correction factor. |

Additional information:

Histogram (Page 215)

### 9.3.9.2.3 Histogram

In this window, the histogram of the gray values inside the yellow search range and the chosen thresholds are shown.

In the example shown here, clear maxima for foreground and background can be recognized. The switching threshold for binarization is positioned approximately in the middle in between.

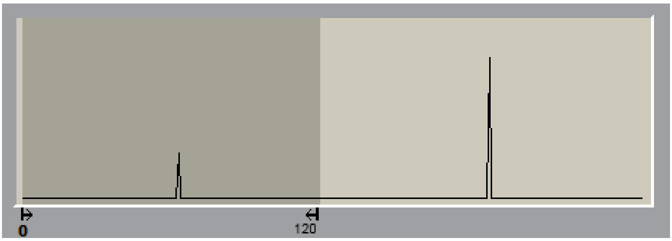



*Fig. 158: Binarization tab, histogram*

### 9.3.9.2.4 Boundary objects

If the "Boundary BLOBs" checkbox is activated, the selected BLOBs / objects are taken into account, even if they are not completely within the yellow search range. (Of course, the objects must always meet the BLOB features within the selected thresholds.)

**NOTE:**
BLOBs are also considered to be boundary BLOBs when they touch or are partially covered by an area that has been masked with "Edit pattern" / "Edit search range" (even if the masked / hidden area is inside the search range).

### Example 1: Boundary objects, touches outer yellow search range

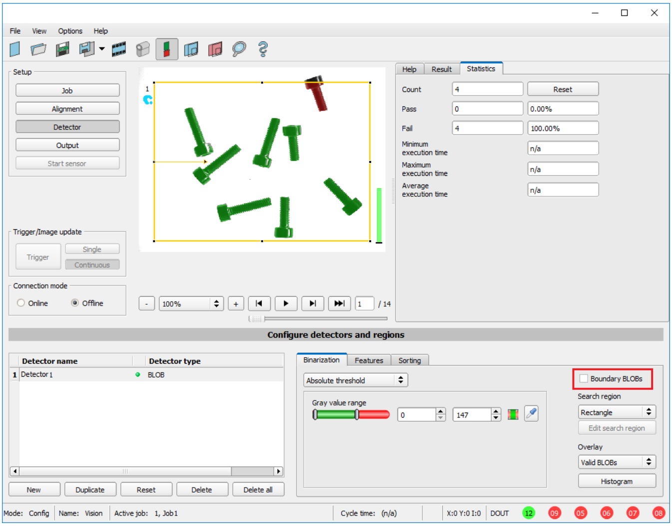



*Fig. 159: Edge objects, example 1.1*

The BLOB touches the outer, yellow search area. It is not detected / marked as a valid BLOB because "Boundary objects" is NOT enabled.

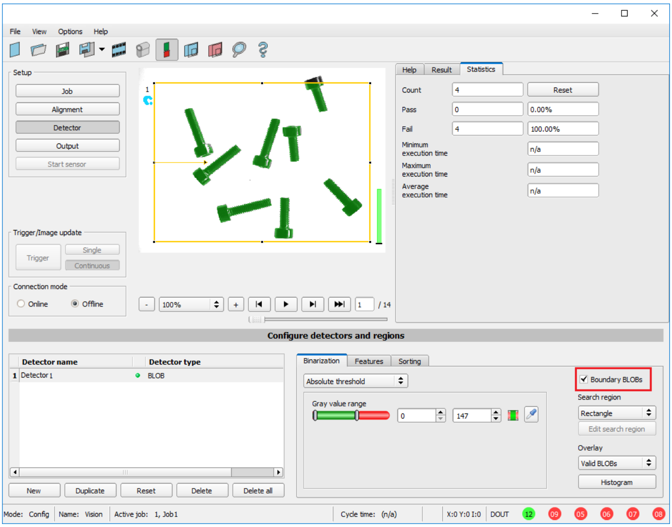


*Fig. 160: Edge objects, example 1.2*

The BLOB touches the outer, yellow search area. It will now be detected / marked as a valid BLOB because "boundary objects" is ACTIVE.

**Example 2, boundary objects touching the inner, masked area.**

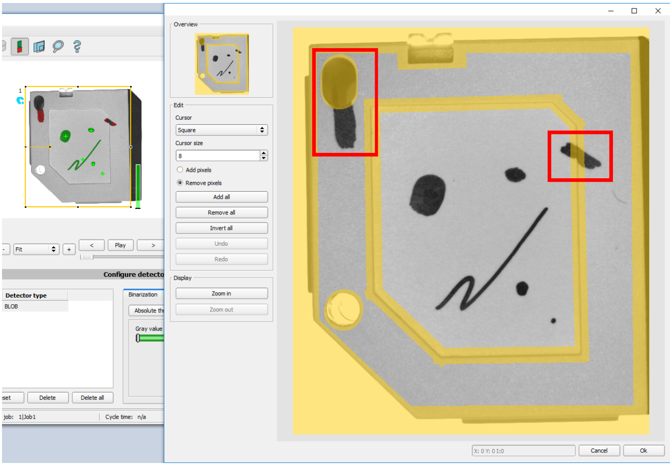



*Fig. 161: Edge objects, example 2.1*

The BLOBs touch the inner, yellow, masked areas. They are not detected as valid BLOBs, as "Boundary objects" is NOT activated.

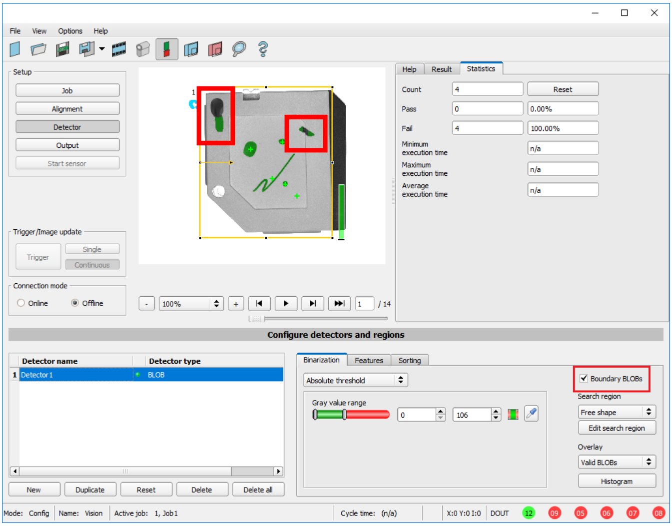



*Fig. 162: Edge objects, example 2.2*

The BLOBs touch the inner, yellow masked areas, but they are now detected / marked as valid BLOBs, because the option "border objects" is ACTIVE.

### 9.3.9.3 Features tab

In the Features tab, the features / filter criteria to distinguish between valid and invalid BLOBs / objects are defined. Only the valid BLOBs are processed further, e.g. for data output.

**Example:** If the thresholds for the feature "Area" have been set to the range 100 ... 150 (pixels), only BLOBs with an area within this area will be recognized as valid (green).

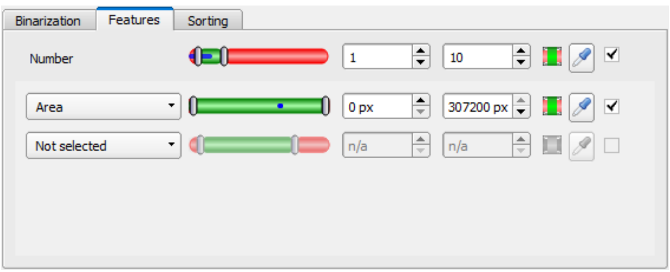



*Fig. 163: BLOB detector, Features tab*

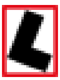
**Parameter description:**

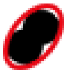
| Parameter | Function |
|---|---|
| Number | The parameter "Number" can be used to check the number of found and valid BLOBs in addition to the characteristics. For this purpose, the upper and lower limit for the accepted number of BLOBs (max. 10,000) is determined.<br>• Detector result positive: Number of valid (filtered) BLOBs is within the range of "Number".<br>• Detector result negative: Number of valid (filtered) BLOBs is outside the range of "Number".<br>If the number of BLOBs is outside the defined limits, the detector result is negative, although valid BLOBs in the image are marked green.<br>If the detector counts more than 10,000 BLOBs (maximum), the detector result is negative and processing is stopped.<br>**NOTE:**<br>Defect detection via number = 0 |
| Invert button / | With the "Invert button" (default: red/green/red), the logic for the evaluation can be inverted. This way, the relevant range can be included or excluded. |
| Pipette symbol (Number) | By clicking this symbol, both limits of "Number" are set to the exact number of BLOBs found in the image. |
| Pipette symbol (Feature) | By clicking on the "Pipette button", the cursor changes into a pipette symbol. When the cursor is moved and a pixel within a valid (green) BLOB is clicked, the thresholds of the selected feature are set to +/-10% of the value of the BLOB that was clicked.<br>Example: If the feature "Area" was selected and clicked with an active pipette on a pixel within a valid BLOB, the two thresholds for Area are set to +/-10% of the calculated number of pixels (= area) of the selected BLOB. |
| Checkbox (Default: Active) | • Active: Feature is calculated, filtered (limits adjustable), and available for data output.<br>• Inactive: Feature is calculated, NOT filtered, but is available for data output. |

**First level features: BLOB type / geometric model**

The features of the first level (Area, Area incl. holes, Contour length, etc.) are calculated directly from the BLOB data, i.e. the pixels belonging to the BLOB. For further features, a geometric model is first fitted to the data via a best-fit line. The features are then based on this model and not directly on the pixels belonging to the BLOB.

| Feature | Function |
|---|---|
| Area | Area of the BLOB, without holes, in pixels. Corresponds to the number of pixels belonging to the BLOB. |
| Area (with holes) | Area of the BLOB, with holes, in pixels. Corresponds to the number of pixels within the outer contour. |
| Contour length | Number of pixels of the outer contour of the BLOB. |
| Compactness | Compactness of the BLOB (ideal circle = 1, all other > 1)<br>The stronger the shape of the BLOB deviates from an ideal circle, the larger the value for compactness will be. Value range 1 ... 100 (limited at 100; BLOBs with higher values are marked as invalid) |
| Center of gravity X | X coordinate of the BLOB center of gravity in pixels.<br>When the calibration is activated in the "Job" setup, the value output can also be stored in world coordinates, e.g. in millimeters. |
| Center of gravity Y | Y coordinate of the BLOB center of gravity in pixels.<br>When the calibration is activated in the "Job" setup, the value output can also be stored in world coordinates, e.g. in millimeters. |
| Gray scale value, average | Average gray scale value of all the pixels that belong to the BLOB. |

| BLOB type / Geometric Model | Function |
|---|---|
| Some features are calculated based on a given geometric model. For example, eccentricity is based on an ellipse fit of the object. | |
|  Rectangle, paraxial (R1) | Enclosing rectangle parallel to Y axis and X axis. Outliers are not eliminated. |
|  Rectangle, minimum area (R2) | Enclosing rectangle with smallest area. Outliers are not eliminated. |
|  Circle, fit (C1) | Circle-fit, not enclosing, outlier correction (robust against outliers) |

| BLOB type / Geometric Model | Function |
|---|---|
|  Ellipse, equivalent (E1) | Equivalent ellipse, based on moments of area. |

**Features / second level: BLOB type parameter**

**[*]** Possibility of value output in world coordinates [mm] when calibration is activated

| Feature | Relevant for | Function | [*] |
|---|---|---|---|
| Center X | R1, R2, C1, E1 | X coordinate of the center of the fitted, geometric element (rectangle, circle, ellipse) | ✓ |
| Center Y | R1, R2, C1, E1 | Y coordinate of the center of the fitted, geometric element (rectangle, circle, ellipse) | ✓ |
| Width | R1, R2, E1 | Width of geometric element. Width ≥ 0, width ≥ height.<br>The orientation is chosen in a way that width is always greater than the height.<br>(Exception: R1, Rectangle, paraxial: Width always in horizontal orientation = parallel to X axis) | ✓ |
| Height | R1, R2, E1 | Height of geometric element. Height ≥ 0, height ≤ width.<br>The orientation is chosen in a way that width is always greater than the height.<br>(Exception: R1, Rectangle, paraxial: Height always in vertical orientation = parallel to Y axis) | ✓ |
| Angle (180) | R2, E1 | Orientation of the width (long axis) of the object in degrees (°), (range -90 ... + 90 °, 0 ° = east, counterclockwise).<br>Please refer to the following as well: Feature Angle (Page 222) | |
| Angle (360) | R2, E1 | Orientation of the width (long axis) of the object in degrees (°), (range -180 ... + 180 °, 0 ° = east, counterclockwise).<br>Please refer to the following as well: Feature Angle (Page 222) | |
| Axial ratio | E1 | Ratio of the long to the short axis (a / b) | |
| Face up / down, area | E1 | Face up / down distinction, based on area, indicated by sign.<br>Please refer to the following as well: Feature Face up / down (Page 227) | |
| Radius | C1 | Specifies the radius of the fitted circle. | ✓ |

| Feature | Relevant for | Function | [*] |
|---|---|---|---|
| Deviation, inside | C1 | Returns the largest deviation between the BLOB contour and the contour of the geometric element (deviation inside the fitted circle).<br>Please refer to the following as well: Feature Deviation (Page 224) | ✓ |
| Deviation, outside | C1 | Returns the largest deviation between the BLOB contour and the contour of the geometric element (deviation outside the fitted circle).<br>Please refer to the following as well: Feature Deviation (Page 224) | ✓ |
| Deviation, mean | C1 | Returns the mean of the absolute "inside" and "outside" deviation values between the BLOB contour and the contour of the geometric element.<br>Please refer to the following as well: Feature Deviation (Page 224) | ✓ |

**Feature Angle**

With the feature "Angle (180)" and "Angle (360)", the orientation of the object can be determined. The angle always indicates the orientation of the width axis (width is the longest side of an object). The angles are specified in [degrees °].

The "Angle (180)" feature has a rotational range of -90° to +90°.

The "Angle (360)" feature depends on the selected geometric model (e.g. E1 Ellipse, R2 Rectangle minimal area, etc.). It has a rotational range of -180° to +180°, as shown in the following figure.

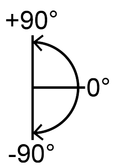



*Fig. 164: Rotational direction of "Angle (180)"*

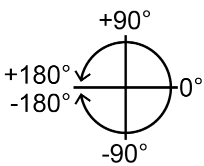



*Fig. 165: Rotational direction of "Angle (360)"*

In contrast to the "Angle (180)" feature, in the "Angle (360)" feature, the orientation of the width axis is set depending on a vector. This vector indicates the direction to the contour point with the longest distance to the center and the orientation (180°) of the vector point to the same side. Whether an object lies in half plane (-90° ... +90°) or in (-180° ... -90°; 90° ... 180°) is determined by the half plane in which the vector lies. The following figures show two examples of the angle determination of the feature "Angle (360)".

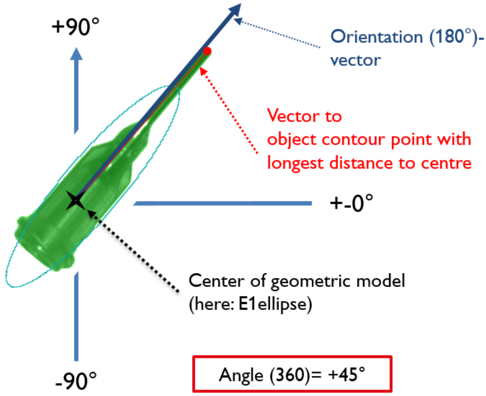



*Fig. 166: Example 1: Angle (360) with +45°*

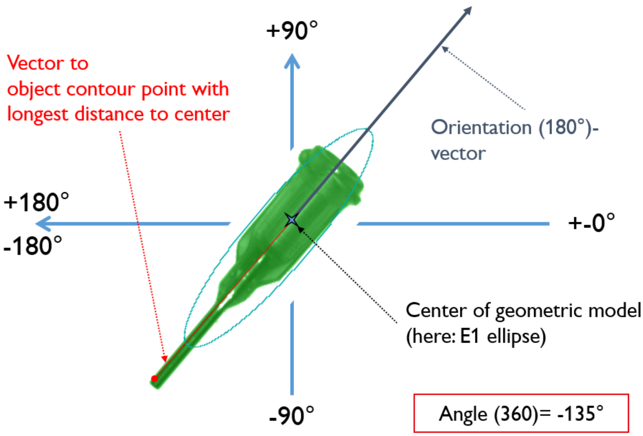
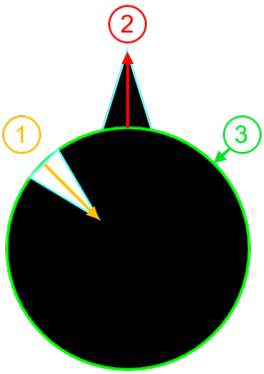



*Fig. 167: Example 2: Angle (360) with -135°*

### Feature Deviation

The deviation feature calculates measures that quantitatively describe the deviation of the actual object from the fitted model. The features "Deviation, inside", "Deviation, outside", and "Deviation, mean" evaluate indentations and outstanding elements of the BLOB / object contour. The deviations always refer to the fitted circle. All indentations inside the fitted circle are "Deviation, inside". All elements that protrude from the fitted circle are judged by the feature "deviation, outside". The orientation directions of the features are shown graphically in the following figure.




**(1)**: Deviation, inside
**(2)**: Deviation, outside
**(3)**: Fitted circle C1

*Fig. 168: Orientation direction "Deviation, inside" and "Deviation, outside"*

In the "Result" tab of the VISOR® software, the value of the largest deviation towards the inside and the value of the largest deviation towards the outside (provided they are "enabled") are always shown for each fitted circle.

The characteristic "deviation, mean" gives the average of the amounts of the deviations to all positions, i.e. to all pixels of the fitted circle.

### Example for the assessment of the mean deviation

Jagged elements are examined for the feature "Deviation, mean", see figure "Deviation, mean".

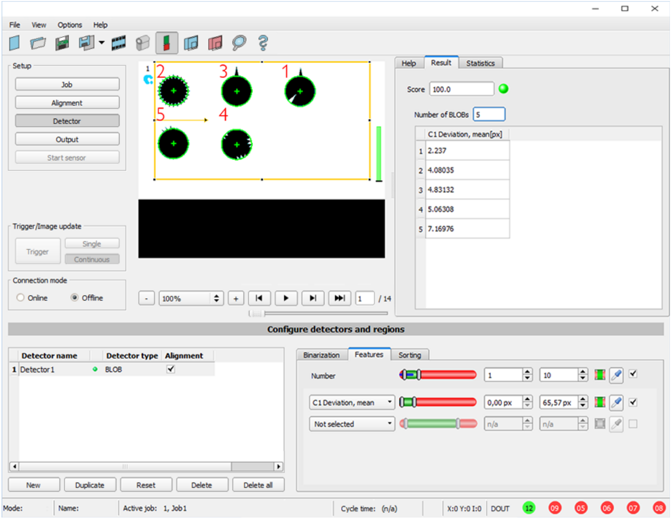


*Fig. 169: Deviation, mean*

The feature "Deviation, mean" calculates all deviations from the fitted circle (green) to the contour of the object / BLOB (cyan) per pixel of the fitted circle inwards and outwards. The following figure shows a zoomed out section of the circle number "2" from the previous figure. The red arrows indicate the deviations per pixel of the fitted circle to the BLOB contour. The amounts of all determined values are averaged and form the result of the "Deviation, mean" feature.

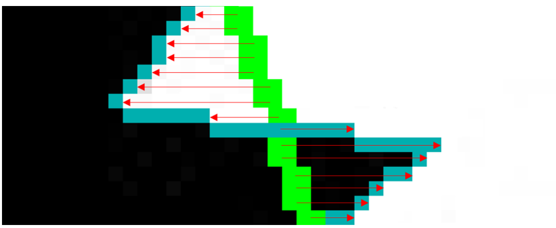


*Fig. 170: Deviation, mean: Circle 2 zoomed*

### Example for assessing deviations from circles inside and outside

Six circles with different notches and protruding elements must be examined for the features "Deviation, inside" and "Deviation, outside".

In order to improve the presentation, "BLOB contour" is selected in the "Binarization" tab of the "BLOB" detector. Now the detector marks the contours of all circles in the search field in cyan.

The following features are selected in the "Features" tab:

- "C1 Circle, fit" (first-level feature), "Deviation, inside" (second-level feature)
- "C1 Circle, fit" (first-level feature), "Deviation, outside" (second-level feature)
- "C1 Circle, fit" (first-level feature), "Deviation, mean" (second-level feature)

In the "Result" tab, the results of the characteristics per circle can now be read, see also the following figure. (Note: The results can be assigned to the circles by moving the mouse over the circles in the field of view.)

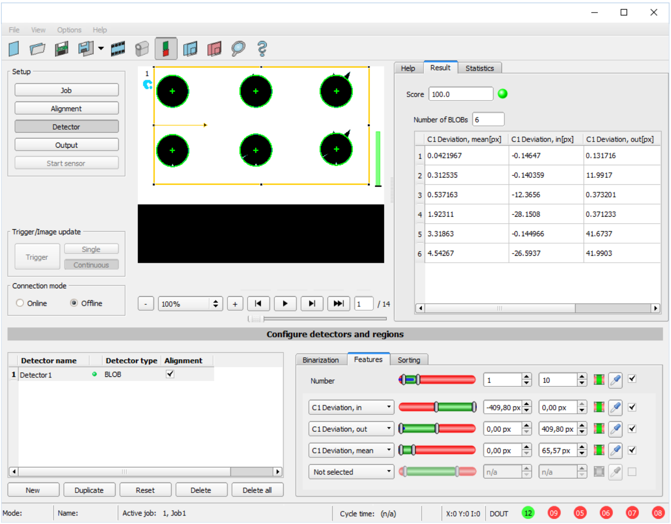



*Fig. 171: "Deviation, inside", "Deviation, outside", and "Deviation, mean" results*

The figure below serves to allocate and interpret the results from the screenshot above.

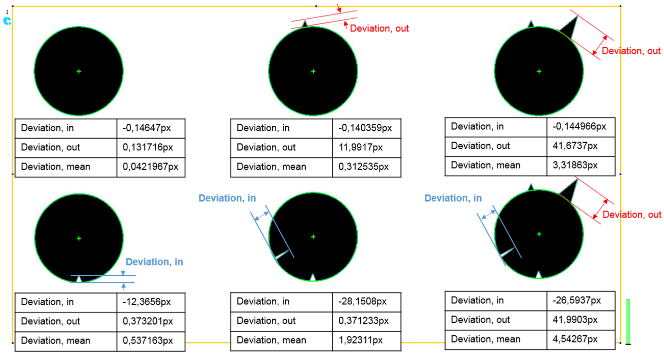



*Fig. 172: Result interpretation of "Deviation, inside", "Deviation, outside", and "Deviation, mean"*

### Feature Face up / down

"Face up / down, area" and "Face up / down, contour" assess the symmetry of the BLOB with respect to an axis determined by the center and the orientation of the BLOB. If a BLOB is fully symmetric with respect to this line, the result value will be 0, otherwise it will deviate from 0. The sign of the value indicates whether the side to the left or right is "stronger".

"Face up / down, area" and "Face up/down, contour" can be used for distinguishing between the face up / down position of an object as necessary in many areas of the delivery technique. Applications can be found, for example, on vibratory conveyors or in robotics.

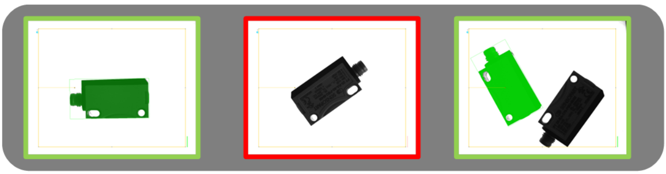


*Fig. 173: Face up / down, area or contour*

The left image shows the searched object e.g. lying on the ventral side. The thresholds are chosen so that this upward-facing position leads to a positive result.
The image in the middle shows the same object face up and is considered "not OK".
The right-hand image displays both objects in an image where only the object facing up is considered "ok".

- "Face up / down, area" uses each pixel associated with the BLOB for the calculation.
- "Face up / down, contour" uses only the contour pixels of the BLOB for the calculation. This method can be used, if e.g. the object varies within the contour or is subject to changes due to reflections or other environmental influences.

The axis used for the calculation is determined by the center and angle (360 °) of the geometric model chosen that was chosen, e.g. smallest enclosing rectangle (R2) or ellipse (E1).

The geometric model for the calculation should be chosen so that the orientation returns a stable and unambiguous value. Thus highly symmetric objects (e.g. rectangles, circles, squares, or point-symmetric objects) cannot be reliably evaluated with this method. For objects where the smallest enclosing rectangle does not provide a unique orientation indication (e.g., "L"-shaped geometries), the ellipse may be a better choice than a geometric model.

### 9.3.9.4 Sorting tab

The characteristics (Features tab (Page 218)) defined in the "Features" tab are calculated for each blob. The results from these calculations can be sent to a PC or PLC in a data telegram, provided that the property has been selected in the corresponding Telegram tab (Page 314). The order in which the results of the individual BLOBs are sent is specified in the "Sorting" tab.

If e.g. the feature "Center of gravity Y" is calculated and there are 5 BLOBs in the image, the output data telegram comprises the results of all 5 BLOBs.

If the sort criterion is "Area" and the sort order is "Descending", the result (here: center of gravity Y) of the BLOB with the largest area is output first.

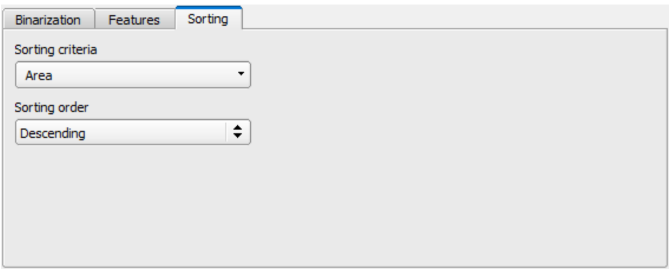



*Fig. 174: Detector BLOB, tab Sorting*

**Parameter description:**

| Parameter | Function |
| --- | --- |
| Sorting criterion | Each feature mentioned under "Detector BLOB / Features tab" can be used as a sorting criterion. |

| Parameter | Function |
| --- | --- |
| Sorting order | Sorting order for the selected sorting criterion. Sorting can be "ascend-ing" or "descending". |

### 9.3.10 Detector Caliper

With this detector, you can control the dimensional accuracy of an object.

#### 9.3.10.1 Color Channel tab

See Chapter: Color Channel tab (Page 147)

#### 9.3.10.2 Probe tab

All caliper parameters can be set here, and the result Histogram can be retrieved.

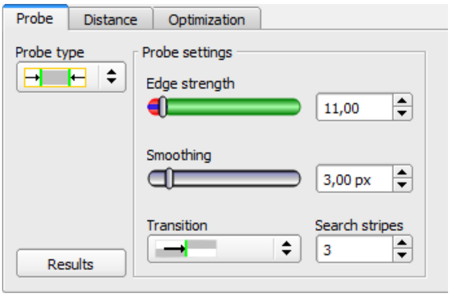
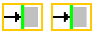



*Fig. 175: Detector Caliper, tab Probe*

**Parameter description:**

| Parameter | Function |
|---|---|
| Probe type | Selection of probe type:<br>• One probe, both sides<br><br>• One probe, one side<br><br>• Two probes, antiparallel (opposite direction)<br><br>• Two probes, same direction<br> |
| Edge thickness | Edge width / contrast from which (0 ... 100) an edge should be detected as an edge. |
| Smoothing | The edge contour is smoothed in the search direction. With larger values, noisy edges, blurred edges, or edges that are not perpendicular to the search direction, are detected more reliably. In addition, light-dark-light or dark-light-dark transitions which are close together with larger values can be ignored. Thus, interfering edges, e.g. scratches, can be hidden. The effect of smoothing can be displayed graphically using the button "Results". |
| Transition | Selection between:<br>• Light → dark<br><br>• Dark → light<br><br>• Both directions (light-dark and dark-light transition)<br> |
| Number of search stripes | Number of parallel search stripes into which the width of the search range is divided. Edge detector is processed in each search stripe over the whole width. The bigger the number of search stripes, the more probable the very first edge will be found. (Finer sampling will result in a longer execution time) |
| Results | Opens result and histogram window |

### 9.3.10.3 Distance tab

Here, all parameters for the desired distance can be set.

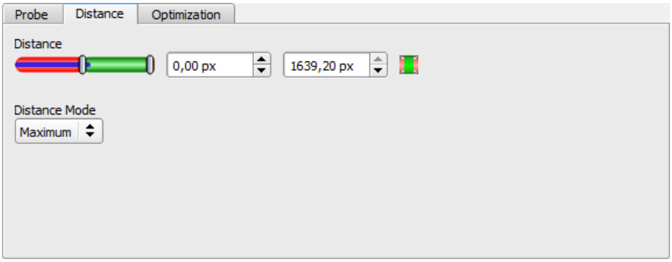



*Fig. 176: Detector Caliper, tab Distance*

**Parameter description:**

| Parameter | Function |
|---|---|
| Distance | Distance range in pixels. Blue bar: Current distance value. |
| Distance mode | For each search stripe, one probe point is determined. If the number of search stripes is greater than 1, there are several possibilities to determine a result from these probe points:<br>• Maximum: The probe point that gives the maximum distance is selected.<br>• Minimum: The probe point that gives the minimum distance is selected.<br>• Mean: All probe points are arithmetically averaged. If there are outliers, they go into the result and falsify it.<br>• Median: The probe points are sorted in ascending order and the mean distance is used. Outliers do not influence the result.<br>Only available for the double-sided probe ""<br>• Maximum by search stripe (of a search stripe): The probe points of the search stripe with the greatest distance are selected.<br>• Minimum by search stripe(of a search stripe): The probe points of the search stripe with the smallest distance are selected. |

Additional information:

**Distance mode**

Information about the structure of the Edge detector can be found at: Structure of the Edge detector (Page 155)

To determine a distance, different distance modes can be selected in the "Distance" tab. The following examples explain the difference between "Minimum" and "Minimum by search stripe" and the difference between "Maximum" and "Maximum by search stripe".

The following object is detected:

*Fig. 177: Demonstration object*

For the examination the double-sided probe type "" is selected in the Probe tab and the number of search stripes is increased to 20.

**Difference between "Minimum" and "Minimum by search stripe"**

With the Distance mode "Minimum by search stripe", in contrast to the Distance mode "Minimum", the probe points are evaluated by only one search stripe. The search stripe with the smallest distance between the probe points is selected (see figure: Results histogram, Distance mode "Minimum by search stripe").

With the Distance mode "Minimum" the smallest distance is determined in which the probe lines with the smallest distance to each other are selected. However, in contrast to the Distance mode "Minimum by search stripe", the probe points of the probe lines can come from two different search stripes (see figure: Results histogram, Distance mode "Minimum").

## Distance mode "Minimum by search stripe"

Minimum by search stripe determined at the probe points of a search stripe.

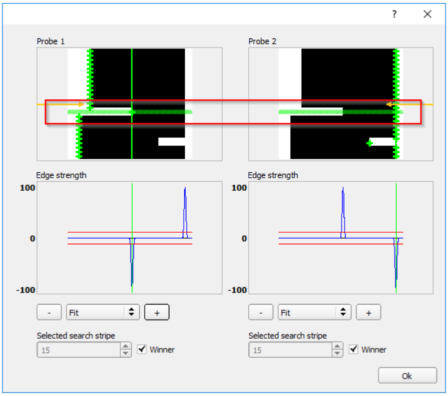



*Fig. 178: Results histogram, Distance mode "Minimum by search stripe"*

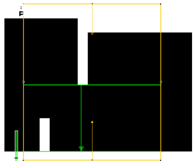


*Fig. 179: Image output with overlay, Distance mode "Minimum by search stripe"*

## Distance mode "Minimum"

Minimum distance determined at the scanning lines with the smallest distance to each other. The probe points of the scanning lines can, but do not have to, come from just one search stripe.

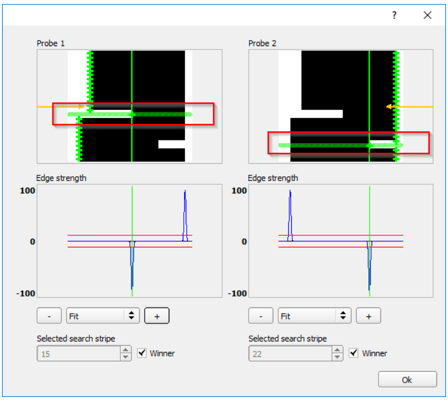



*Fig. 180: Results histogram, Distance mode "Minimum"*

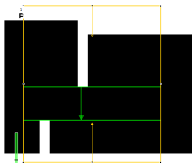


*Fig. 181: Image output with overlay, Distance mode "Minimum"*

### Difference between "Maximum" and "Maximum by search stripe"

The selected example object shows identical image outputs with the Distance mode "Maximum" and the Distance mode "Maximum by search stripe" when determining the distance (see figures in the right columns ofi the tables below). However, the distances are not determined identically.

With the Distance mode "Maximum by search stripe" the probe points are evaluated by only one search stripe. The search stripe with the largest distance between the probe points is selected (see figure: results histogram, Distance mode "Maximum by search stripe").

With Distance mode "Maximum", the maximum distance is determined by selecting the probe lines with the greatest distance to each other. However, the probe points of the probe lines can come from two different search stripes (see figure: Results histogram, Distance mode "Maximum").

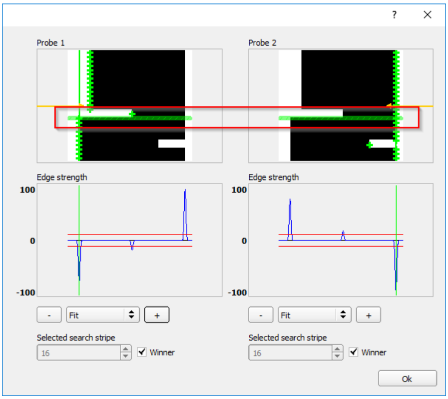
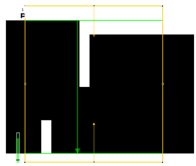
<table>
<tr><th colspan="2">Distance mode "Maximum by search stripe"</th></tr>
<tr><td colspan="2">Maximum by search stripe determined at the probe points of a search stripe.</td></tr>
<tr><td>

Fig. 182: Results histogram, Distance mode "Maximum by search stripe"</td><td>
Fig. 183: Image output with overlay, Distance mode "Maximum by search stripe"</td></tr>
</table>

**Distance mode "Maximum"**

Maximum distance determined at the scanning lines with the largest distance to each other. The probe points of the scanning lines can, but do not have to, come from just one search stripe.

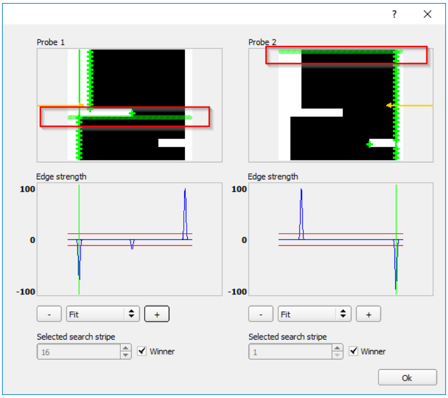



*Fig. 184: Results histogram, Distance mode "Maximum"*

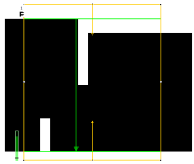


*Fig. 185: Image output with overlay, Distance mode "Maximum"*

### 9.3.10.4 Optimization tab

In the "Optimization" tab, further settings for optimizing the Edge detector can be made.

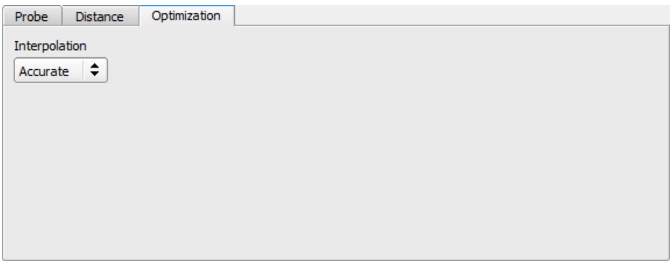



*Fig. 186: Detector Caliper, tab Optimization*

**Parameter description:**

| Parameter | Function |
|---|---|
| Interpolation | The calculation of the edge position can be performed either with sub-pixel accuracy (up to 1/10 pixels) or with simple accuracy. |
| Accurate | Subpixel accuracy |

| Parameter | Function |
|---|---|
| Fast | Simple accuracy: This setting partly provides calculations which are over 50% faster. |

### 9.3.10.5 Results / Histogram window

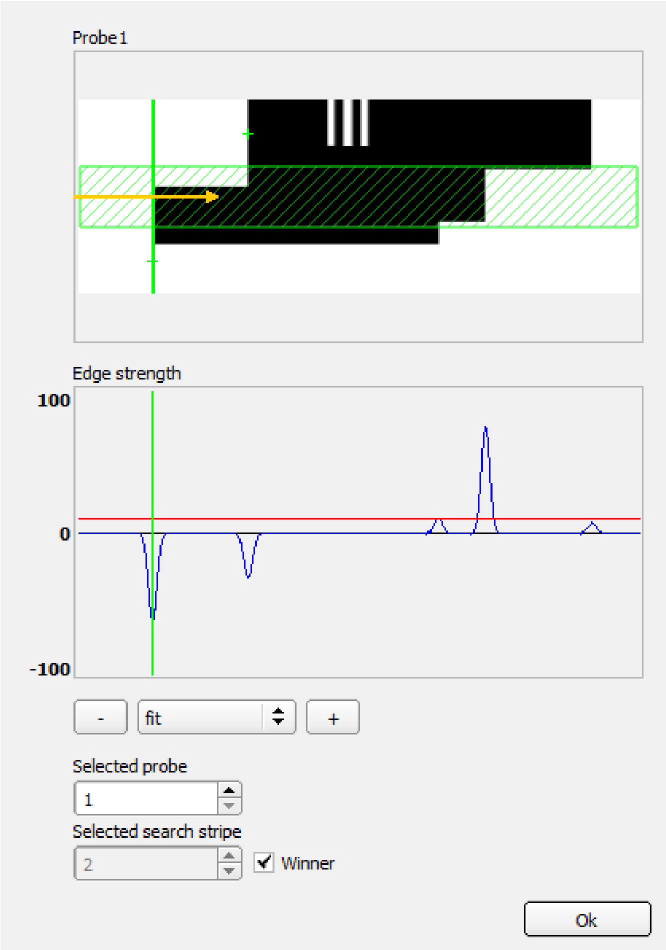



*Fig. 187: Results / Histogram window*

| Parameter | Function |
|---|---|
| Probe (x) | Image of probe (x) with:<br>• Green line: Detected result edge<br>• Green crosses: Determined edge transition (touch point) per search stripe<br>• Light blue or light green zone: Display of "Selected search stripe" |
| Edge thickness | Histogram with:<br>• Blue line: Contrast gradient in the image, depending on the selected search stripe<br>• Red line: Required contrast for edge ("Switching threshold")<br>• Light blue or light green line: Determined edge transition depending on selected search stripe |
| Fit, "+", "-" | Fit or zoom for "edge width" histogram |
| Selected probe | Display of the selected probe |
| Selected search stripe | Selection of the search stripe to be displayed in the probe image<br>• Winner: Winner search stripe (depending on settings in tab "Distance")<br>• "1, 2, .." Number of the displayed search stripe |

## 9.3.11 Detector Barcode

This detector is suitable for barcode reading of 1D codes.

### 9.3.11.1 Code tab

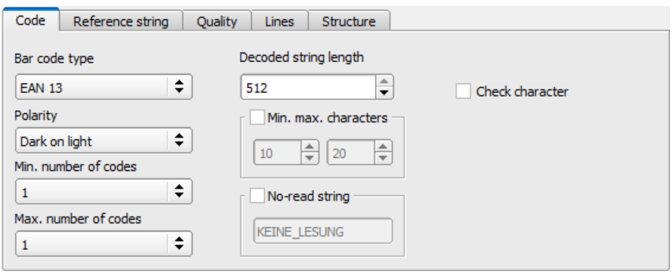



*Fig. 188: Detector Barcode, Code tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Barcode type | Select here the type of barcode to be read with the code reader. |
| Max. string length | Max. length of a barcode If the contents of the code are longer than the maximum length, the rest will be cut off.<br>If more than one code is read simultaneously, this value must be set to the longest length of the longest code. |
| Check digit | This parameter activates the recognition of a check character if it is available in the code. Barcodes with check characters are e.g. Code 39, Codabar, 25 Industrial or 25 Interleaved. If this parameter is not selected, the check character is interpreted as a normal data character and output in the string. |
| Min. Anzahl Codes | Minimum number of codes to be read inside the search range. |
| Max. number of codes | Maximum number of codes to be read inside the search range. If this value is chosen higher than actually necessary, the execution time of the detector may increase slightly. |
| Number of characters | Number of expected characters in the barcode. Codes with a different number of characters are ignored. If the number of characters of the code is known in advance, this increases detection certainty.<br>If codes with a certain number of characters must be found among several codes, the parameter "Max. number of codes" must be set higher than the number of searched codes. |
| Text output for incorrect reading | Specifies the text which is output via the interfaces in case of an incorrect reading. The text does not appear in the result display. |
| Polarity | Choices for ink of code "dark on light" or "light on dark". |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

**Improvement of execution speed**

- Search range for position (yellow frame) only as large as necessary.

**Robust detection**

- Search range (yellow frame) sufficiently large?
- Distinctive contrast present?
- Was the selection "Check character" activated, even though there is no check character in the code?

### 9.3.11.2 Reference string tab

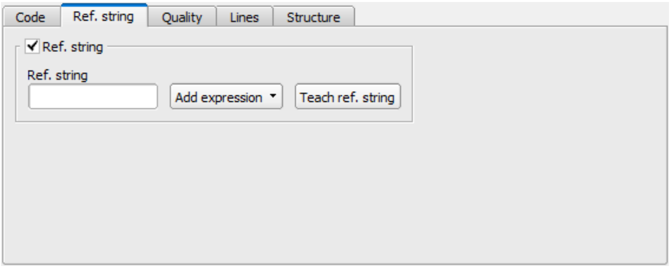



*Fig. 189: Barcode detector , Reference string tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| String comparison | Activates verification of the contents of the read information. The content of the information read is checked on the basis of regular expressions. |
| Reference string | This text or regular expressions is taken for verification. This is where specific characters can be found that are directly compared or regular expressions to check the structure of the read result.<br>If codes with a certain reference string must be found among several codes, the parameter "Max. number of codes" in the "Code" tab must be selected higher than the number of searched codes. |
| Add expression | Opens a list with examples for regular expressions. |
| Teach reference string | Reads the code that is currently under the code reader and accepts the read content as a comparison text. This text can be changed later. |

**Examples of reference character strings defined by regular expressions**

| Reference character string | Hit | Example for hit |
|---|---|---|
| 123 | String containing 123 | 01234 |
| \A123 | String beginning with 123 | 1234 |
| 123\Z | String ending with 123 | 0123 |
| \A123\Z | String matching 123 exactly | 123 |
| [123] | Character string containing one of the characters | 33 |
| [123]{2} | String containing a sequence of 2 of the characters | 23 |

| Reference character string | Hit | Example for hit |
|---|---|---|
| [12]\|[34] | String containing a character of one of both groups | 4 |

**The most important elements of regular expressions:**

| | |
|---|---|
| ^ or \A | Represents the beginning of the character string |
| $ or \Z | Represents the end of the character string, and possibly includes a newline as the last character |
| . | Represents every character apart from newline |
| [...] | Represents any literal listed in the square brackets. If the first character is an '^', the expression is negated. You can use the '-' character, as in '[A-Z0-9]', to specify value ranges. Other characters lose their special meaning within square brackets, except '\'. |
| * | Allows 0 or more repetitions of the preceding literal / group |
| + | Allows 1 or more repetitions |
| ? | Allows 0 or 1 repetitions |
| {n,m} | Allows n to m repetitions |
| {n} | Allows exactly n repetitions |
| \| | Separates alternative search expressions |

### 9.3.11.3 Quality tab

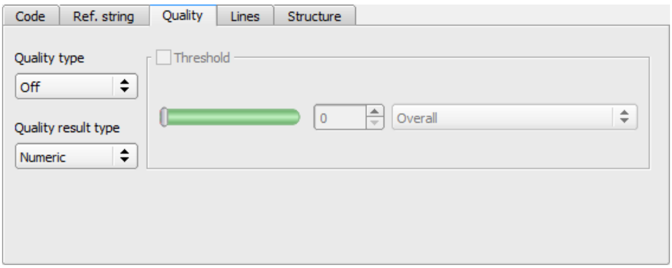



*Fig. 190: Detector Barcode, tab Quality*

The evaluation of the print quality is carried out according to the international standard ISO/IEC 15416.

For a standard-compliant quality assessment, certain minimum requirements for the mapping of the code in the camera (resolution), the arrangement of the camera, and the type and arrangement of the lighting are prescribed. These are printed in the respective standards.

While the overall quality is the final symbol level, the remaining degrees indicate potential quality-reducing causes. A list of frequently occurring defects and their effect on the individual quality grades can be found in the standard ISO / IEC 15416.

**Standard 1D barcodes - Parameter description:**

| Parameter | Function |
|---|---|
| Q1 Overall | Minimum value of all remaining degree values |
| Q2 Not used | - |
| Q3 Not used | - |
| Q4 Decode | Is set to 4 if the examined barcode symbol could be read, otherwise set to 0 |
| Q5 Symbol Contrast | Difference between maximum and minimum reflectance value of the greyscale profile; higher contrast results in a better degree |
| Q6 Minimal Reflect-ance | Is set to 4 if the minimum reflection value of the gray value profile is less than or equal to 0.5 of the maximum reflection value. Otherwise a value of 0 is assigned. |
| Q7 Minimal Edge Con-trast | Evaluates the minimum edge contrast in the greyscale profile. "Edge contrast" is the contrast between two adjacent symbol elements (line-to-hole or hole-to-line). |
| Q8 Modulation | Evaluates the amplitude between the symbol elements. Higher amp-litudes mean that lines and holes can be distinguished more reliably from one another and this degree is rated higher |
| Q9 Defects | Irregularities in the grey tone profile within individual symbol elements or the squeaky zones, the presence of which is indicated by a lower degree |
| Q10 Decodability | Designates deviations of the symbol element widths from their nominal value, which is defined in the corresponding symbology standard. |
| Q11 Additional Requirements | Other symbology specific requirements such as: quiet zone widths, wide/narrow ratio, inter character gaps, guarding patterns or others. |
| Quality parameter Output | There are two presentation formats for quality parameters. Both formats correspond to the norms. The parameters can be specified with values from A-F or 0-4. A and 4 are the best possible grades. The setting made here affects both the display of the quality parameters on the screen and the output of the quality parameters via the interfaces. The assignment is the following:<br>A B C D F<br>4 3 2 1 0 |

**Composite and stacked barcodes - Parameter description:**

The print quality rating of a "composite" barcode includes the following 24 grades:

| Parameter | Function |
|---|---|
| Q1 Overall | Minimum value of all remaining degree values |
| Q2 Overall Linear | Minimum value of Q parameters Q4-Q11; represents the total degree of the linear (1D) part of the composite symbol |
| Q3 Overall Composite | Minimum value of Q parameters Q12-Q24; represents the total degree of the composite (2D) part of the composite symbol |
| LINEAR:<br>Q4 Decode<br>Q5 Symbol Contrast<br>Q6 Minimal Reflect-ance<br>Q7 Minimal Edge Con-trast<br>Q8 Modulation<br>Q9 Defects<br>Q10 Decodability<br>Q11 Additional Requirements | The grades from the group LINEAR correspond to those from the simple 1D barcode case described above. |
| COMPOSITE:<br>Q12 Decode<br>Q13 Rap Overall | The grades from the group COMPOSITE correspond to the PDF 417 quality grades, whereby rap overall is named after the so-called RAP start-stop pattern, which is specific for composite symbols. |
| COMPOSITE RAP:<br>Q14 Contrast<br>Q15 Minimal Reflect-ance<br>Q16 Minimal Edge Contrast<br>Q17 Modulation<br>Q18 Defects<br>Q19 Decodability<br>Q20 Codeword Yield<br>Q21 Unused Error Correction<br>Q22 Modulation<br>Q23 Decodability<br>Q24 Defects | In addition, the COMPOSITE RAP subgroup represents the individual gray level profile grades of the RAP pattern. These are consistent with the simple 1D barcode quality grades. |

| Parameter | Function |
|---|---|
| Quality parameter<br>Output | There are two presentation formats for quality parameters. Both formats correspond to the norms. The parameters can be specified with values from A-F or 0-4. A and 4 are the best possible grades. The setting made here affects both the display of the quality parameters on the screen and the output of the quality parameters via the interfaces. The assignment is the following:<br>A B C D F<br>4 3 2 1 0 |

### 9.3.11.4 Lines tab

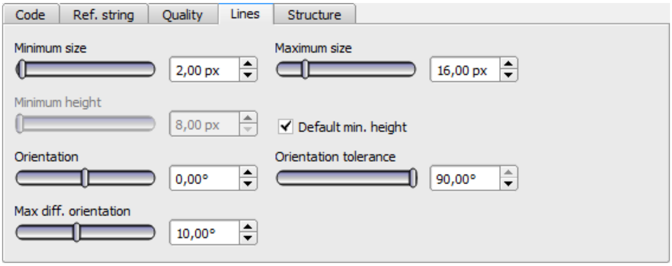



*Fig. 191: Detector Barcode, Lines tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Minimum<br>Line width | The minimum size of an element, i.e. minimum width of all lines and spaces. For very small barcodes, the value should be reduced to 1.5. In the case of huge barcodes, the value should be increased, which results in a shorter execution time. |
| Maximum<br>Line width | The maximum size of an element, i.e the maximum thickness of all lines and spaces. This value should be large enough to find the candidate region for the entire symbol. On the other hand, the value should not be so large that two adjacent barcodes merge into a single candidate. |
| Minimum<br>Line height | The minimum barcode height For very flat barcodes with a height of less than 16 pixels, it makes sense to set the height manually so that the barcode can be found and read. The minimum height is 8 pixels. If the barcode is very high, e.g. 70 pixels and more, manually adjusting to the respective height can lead to quicker reading. |

| Parameter | Function |
| --- | --- |
| Orientation | The Orientation parameter can be used to restrict the angle range for the code reading. Barcodes with rotational positions outside the specified orientation are not read. The parameter is used, for example, if a barcode can lie in different rotational positions in front of the code reader and not all rotational positions should be read. If codes with a certain rotational position need to be found among several codes, then the parameter max. number of codes in the Code tab should be selected higher than the number of searched codes.<br>If the barcodes only appear in the edited images with a certain orientation, then you can reduce the value range accordingly. This will detect wrong candidates sooner. The execution time of the operator is shortened if the orientation angle is restricted. This strategy is especially true when the edited images contain a lot of background texture with misoriented, barcode-like structures. |
| Orientation tolerance | Tolerance of orientation. See "Orientation" for further details. |
| Max. deviation (orientation) | A potential barcode comprises lines, and hence edges with a consistent orientation. The size "Maximum orientation deviation" indicates how strong the difference in the orientation of adjacent edges may be. The maximum orientation deviation is a differential angle in degrees. If a barcode is frayed, i.e. the line edges are interefered, the maximum orientation deviation should be selected as large. However, with small values, the number of wrong barcode candidates can be reduced. |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

**Parameter Orientation**

The following figure illustrates the orientation.

**NOTE:**
The specification of the orientation refers to the image and not to the rotational position of the search range.

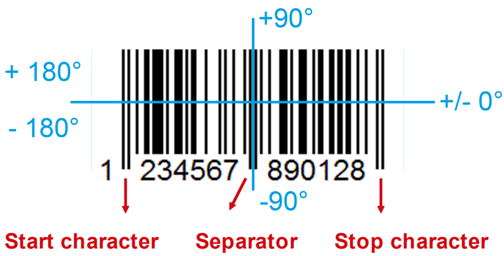



*Fig. 192: Parameter Orientation*

#### Improvement of execution speed

- Search range for position (yellow frame) only as large as necessary.

#### Robust detection

- Search range (yellow frame) sufficiently large?
- Distinctive contrast present?
- Are the thresholds set correctly?
- Code size sufficient in the field of view?
- Is the line width sufficiently large?

### 9.3.11.5 Structure tab

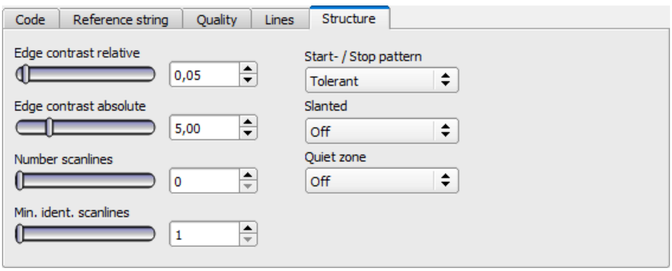



*Fig. 193: Detector Barcode, tab Structure*

**Parameter description:**

| Parameter | Function |
|---|---|
| Edge contrast relative | Edges are found within a scan line using a threshold. The parameter "Edge contrast relative" defines how this threshold is calculated relative to the dynamic range of the gray values along the scan line. For large background noise or noise, edge contrast should be set relatively higher. Typical value range: [0.05 ... 0.2]; Default: 0.05 |
| Edge contrast absolute | "Edge contrast absolute" prevents the misdetections of edges. For high noise images, this parameter should be set higher. For low-contrast, noise-free images, over-reading may interfere with correct edge detection. In such cases, it is advisable to reduce the value or to set it to 0.0. Typical values: [0.0 ... 10.0]; Default: 5.0 |
| Number scan lines | Number of scan lines used during the scanning of a code. Reducing the number of scan lines improves speed. Better quality images require fewer scan lines than lower quality images. For average image quality, a value between 2 and 5 are adequate. If a barcode cannot be found after reducing the scan lines, the number of scan lines must be increased again. Typical values: [0, 5, 10, 20 ...]; Default: 0 |
| Min. ident. Scanlinien | The minimum number of identical scan lines to accept a code as read. If this parameter is not set (value 0), the barcode is read as soon as the scan line has been successfully decoded. Faulty readings can be reduced if this parameter is set to 2 or higher. Typical values: [0, 2, 3, ...]; Default: 0 |
| Start / Stop pattern | Sets the search for start or stop patterns within a scan line to "Tolerant" or "Exact". "Tolerant" increases the general read rate, especially in images with poor contrast. "Exact" increases the robustness against incorrect decoding but can also reduce the general reading rate. Standard value: "Tolerant" |
| Slanted | If "slanted" = "On", improved readability when individual lines of the barcode are aligned obliquely to the main direction of the code, e.g. if the code appears distorted by an uneven surface.<br>If "slanted" = "Off", default setting when all lines of the barcode appear parallel in the image.<br>If "Tilt" = "Auto", first the "Off" position and then the "On" position is tested, which can increase reading time.<br>Values: "Off", "Auto", "On"; Default: "Off" |

| Parameter | Function |
|---|---|
| Quiet zone | Controls the detection of quiet zones of a barcode. With "Quiet zone" = "On", the quiet zones must be at least as wide as specified by the corresponding barcode standard. With "Quiet zone" set to an integer (≥ 1), the quiet zones of at least "Quiet zone" x X pixels must be observed. With "Quiet zone" = "Tolerant", a limited number of edges are allowed in the quiet zone, but at most 1 per 4 module widths. The goal is to prevent you from recognizing only part of the barcode, but still be able to read codes with a simple violation of the quiet zone. With "Quiet zone" = "Off", the detection of the quiet zones is disabled. Detection of the quiet zone prevents simple barcodes from being found within a stroke sequence of a longer and / or more complex barcode. Usually, values between 2 and 4 achieve optimal results because they suppress false barcodes while still tolerating small interferences such as text, label edges, etc. Typical values: "Off" "On", 1, 2, 3, 4, 5; Default: "Off" |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

## 9.3.12 Detector Datacode

This detector is suitable for reading 2D DataMatrix codes.

### 9.3.12.1 Code tab

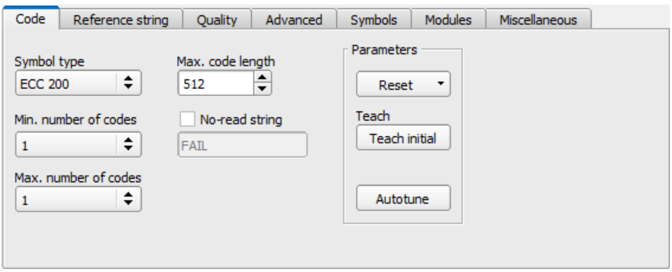



*Fig. 194: Datacode detector, Code tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Code type | Select here the type of the data code to be read. |

| Parameter | Function |
| --- | --- |
| Max. string length | Max. length of a code. If the contents of the code are longer than the maximum length, the rest will be cut off.<br>If more than one code is read simultaneously, this value must be set to the longest length of the longest code. |
| Min. Number of codes | Minimum number of codes to be read inside the search range. |
| Max. number of codes | Maximum number of codes to be read inside the search range. If this value is chosen higher than actually necessary, the execution time of the detector may increase slightly. |
| Reset | Reset parameter is used to reset the learned parameters to the initial state before teaching. There are the options "Standard", "Advanced", and "Maximum". "Standard" sets the limits of the search range so that the majority of the possible codes are recognized during training. If your code is still not recognized, select the setting "Advanced". If the code still cannot be read, set the setting to "Maximum". The settings "Advanced" and "Maximum" may slow down the execution time. The difference to a complete resetting of the sensor is that only the parameters for the Data matrix code are reset. The basic parameters of the sensor, e.g. for lighting, inputs, serial interface, etc. are retained. After resetting the parameters, a new teach-in process can be started again with "Teach-in". |
| Initial teach / Additive teach | Teach-in: The search area of the sensor is searched for a Data matrix code. If a code was found, the parameters are stored for this code. After successful teaching, the found code is marked with a green frame. In the "Run" mode, only this very taught-in code is searched for. After completing the teach-in, the button "Additional teach-in" appears at the same place. This allows the extension of the taught-in parameters to either read several different codes in one configuration or to capture any existing spreads in the print quality of a single code. "Additional teach-in" extends the already taught-in parameter set. |
| Autotune | Automatic setting (pre-processing filter and image settings) for the optimization of code reading. |
| Text output for incorrect reading | Specifies the text which is output via the interfaces in case of an incorrect reading. |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

#### Improvement of execution speed

- Search range for position (yellow frame) only as large as necessary.

### Robust detection

- Search range (yellow frame) sufficiently large?
- Distinctive contrast present?
- Are thresholds set correctly?

## 9.3.12.1.1 Autotune

If you select the "Autotune" function, the vision sensor will automatically configure its settings in order to optimize its code reading functionality.

The function always starts with the parameters already set by the user. So if parameters are roughly set before starting the "Autotune" function, then Autotune fine tunes to optimize the result.

After the optimization run of "Autotune", "OK" or "Cancel" can be selected. In case of "OK", the newly found parameters are used. When "Cancel" is selected, the old parameters are restored to the values from before the "Autotune" execution.

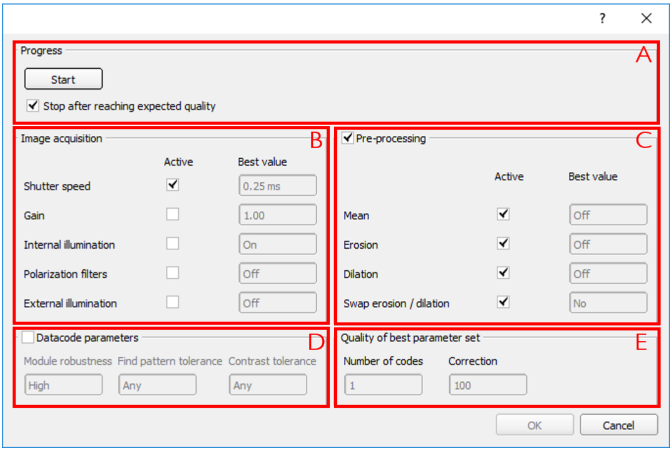



*Fig. 195: "Autotune" window*

The "Autotune" function consists of the following elements:

| A: Fortschritt | |
|---|---|
| start | Start of "Autotune" function. After pressing start, the progress is displayed. |
| Stop after reaching expected quality | The "Autotune" function stops the automatic settings when the minimum required quality is reached. |

| B: Bildaufnahme | |
|---|---|
| Shutter speed | **Enabled:** If the "Enabled" checkbox is enabled, the parameters that the VISOR® vision sensor should determine automatically will be determined. Non-activated parameters remain unchanged.<br>**Best Value:** The "Best Value" field shows the last setting which was determined by the Autotune function. |
| Gain | |
| Internal illumination | |
| Polarization filters | |
| External illumination | |

| C: Vorverarbeitung | |
|---|---|
| Mean | **Enabled:** If the "Enabled" checkbox is enabled, the parameters that the VISOR® vision sensor should determine automatically will be determined. Non-activated parameters remain unchanged.<br>**Best Value:** The "Best Value" field shows the last setting which was determined by the Autotune function. |
| Erosion | |
| Dilation | |
| Inversion order: erosion / dilatation | |

| D: Codeparameter | |
|---|---|
| Module robustness | The best settings found by the Autotune function are displayed. |
| Search pattern tolerance | |
| Contrast tolerance | |

| E: Qualität des besten Parametersatzes | |
|---|---|
| Number of codes | Number of codes in the field of view tested by the Autotune function. |
| Correction | Decode error, which is achieved with activated parameters. |

### 9.3.12.2 Reference string tab

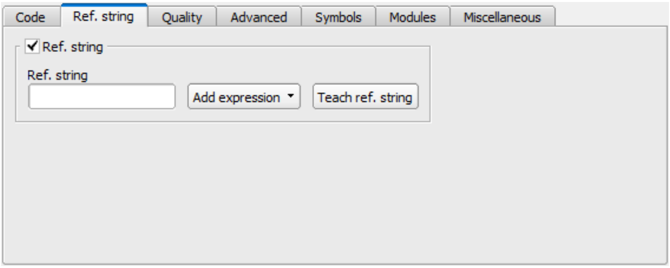



*Fig. 196: Detector Datacode, tab Reference string*

**Parameter description:**

| Parameter | Function |
|---|---|
| Reference string | This text or regular expressions is taken for verification. This is where specific characters can be found that are directly compared or regular expressions to check the structure of the read result.<br>If codes with a certain reference string must be found among several codes, the parameter "Max. number of codes" in the "Code" tab must be selected higher than the number of searched codes. |
| Add expression | Opens a list with examples for regular expressions. |
| Teach reference string | Reads the code that is currently under the code reader and accepts the read content as a comparison text. This text can be changed later. |

**Examples of reference character strings defined by regular expressions**

| Reference character string | Hit | Example for hit |
|---|---|---|
| 123 | String containing 123 | 01234 |
| \A123 | String beginning with 123 | 1234 |
| 123\Z | String ending with 123 | 0123 |
| \A123\Z | String matching 123 exactly | 123 |
| [123] | Character string containing one of the characters | 33 |
| [123]{2} | String containing a sequence of 2 of the characters | 23 |
| [12]\|[34] | String containing a character of one of both groups | 4 |

**The most important elements of regular expressions:**

<table>
<tr><td>^ or \A</td><td>Represents the beginning of the character string</td></tr>
<tr><td>$ or \Z</td><td>Represents the end of the character string, and possibly includes a newline as the last character</td></tr>
<tr><td>.</td><td>Represents every character apart from newline</td></tr>
<tr><td>[...]</td><td>Represents any literal listed in the square brackets. If the first character is an '^', the expression is negated. You can use the '-' character, as in '[A-Z0-9]', to specify value ranges. Other characters lose their special meaning within square brackets, except '\'.</td></tr>
<tr><td>*</td><td>Allows 0 or more repetitions of the preceding literal / group</td></tr>
<tr><td>+</td><td>Allows 1 or more repetitions</td></tr>
<tr><td>?</td><td>Allows 0 or 1 repetitions</td></tr>
<tr><td>{n,m}</td><td>Allows n to m repetitions</td></tr>
<tr><td>{n}</td><td>Allows exactly n repetitions</td></tr>
<tr><td>|</td><td>Separates alternative search expressions</td></tr>
</table>

### 9.3.12.3 Quality tab

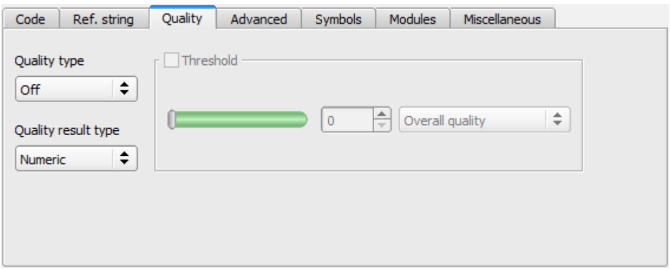



*Fig. 197: Detector Datacode, tab Quality*

Quality parameters are provided as additional information to evaluate the quality of the codes. There are quality parameters according to AIM DPM-1-2006 and ISO / IEC 15415 standard.

For a standard-compliant quality assessment, certain minimum requirements for the mapping of the code in the camera (resolution), the arrangement of the camera, and the type and arrangement of the lighting are prescribed. These are printed in the respective standards.

The quality parameters are composed of the following individual ratings:

**Parameter description:**

| Parameter | Function |
|---|---|
| Q1 Overall quality | Minimum of all following individual ratings |
| Q2 Contrast | Range between minimum and maximum pixel intensity. A strong contrast results in a good grading. |
| Q3 Modulation | Returns a value for the ratio of black to white in the code. Too much, as well as too little, black lead to a reduction of the value. A balanced ratio of black and white simplifies the assignment of the individual modules. |
| Q4 Fixed pattern damage | The fixed pattern of both the ECC200 and QR code is of great importance for recognizing and decoding the codes. Fixed pattern damage provides information about the state of the "finder pattern" and the quiet zones of the code. |
| Q5 Decode | Always has the value 4 if the code was read successfully. Codes that cannot be read cannot be evaluated for quality. Because data matrix codes include error correction, errors in individual modules can be corrected. The sum of the corrected errors is mapped in the value "unused error correction". It is quite possible that codes with an Unused Error score of 0 can still be read. |
| Q6 Axial non-uniformity | Provides information about any horizontal or vertical distortion of the code. |
| Q7 Grid non-uniformity | Gives information about general distortions of the code. |
| Q8 Unused error correction | Quality parameters according to AIM DPM-1-2006 are an extension to the ISO/IEC 15415 Standard, which defines the specific requirements of the gray value settings of the image of the Data matrix code, and thus improves the reproducibility of the quality evaluation between the different manufacturers.<br>The unused error correction capacity of the treated symbol is calculated in the degree Unused error correction. |
| Q9 Mean light | Quality parameters according to AIM consist of one value more than quality parameters according to ISO/IEC 15415.<br>This value is called "Mean light".<br>"Mean light" is not a quality value for the code; it provides information about the quality of the image by calculating the average gray value of the light data code modules.<br>"Mean light" can have values from 0.0 to 1.0, which is 0% to 100% of the maximum gray value.<br>An image has the required gray value properties if the value "mean light" is between 70% and 86% (i.e. 0.70 to 0.86). |

| Parameter | Function |
|---|---|
| Quality parameter Output | There are two presentation formats for quality parameters. Both formats correspond to the norms. The parameters can be specified with values from A-F or 0-4. A and 4 are the best possible grades. The setting made here affects both the display of the quality parameters on the screen and the output of the quality parameters via the interfaces. The assignment is the following:<br>A B C D F<br>4 3 2 1 0 |

### 9.3.12.4 Advanced tab

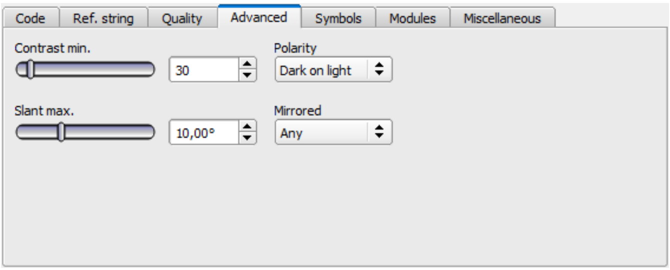



*Fig. 198: Detector Datacode, tab Advanced*

**Parameter description:**

| Parameter | Function |
|---|---|
| Min. Kontrast | Minimum contrast in grayscale between light and dark parts of the code, value range (1 ... 100). |
| Polarity | Possible settings: Should light code be read on a dark background or dark code on a light background? |
| Slanted | Maximum deviation of the angle in the L-shaped finder pattern from the (ideally) right angle. The specification corresponds to perspective distortions that may occur when the symbol is printed or when the image is captured. |
| Mirrored | Setting option, whether the code was applied mirrored or not. Due to the symmetry of the code, this is not visible to the eye. The function is helpful if e.g. codes on a transparent surface should be read from behind. |

### 9.3.12.5 Symbols tab

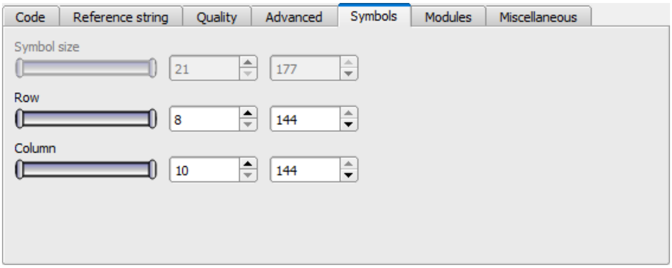



*Fig. 199: Detector Datacode, tab Symbols*

**Parameter description:**

| Parameter | Function |
| --- | --- |
| Symbol size min. / max. | Only QR code: Size of symbol in the image in pixels. |
| Columns min. / max. | Only ECC200 and PDF 417: Number of columns including finder pattern. |
| Rows min. / max. | Only ECC200 and PDF 417: Number of rows including finder pattern. |

### 9.3.12.6 Modules tab

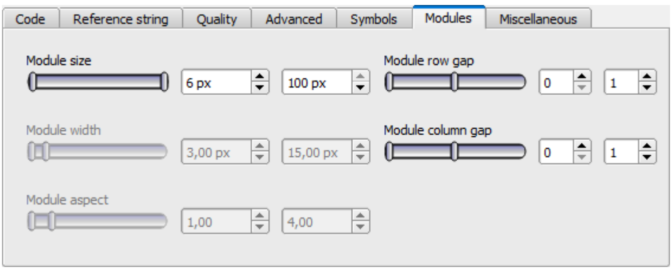



*Fig. 200: Detector Datacode, tab Modules*

**Parameter description:**

| Parameter | Function |
|---|---|
| Module size min. / max. | Size of modules in pixels. |
| Module width min. / max. | Only PDF 417: Width of the modules in the image in pixels. |
| Module aspect min. / max. | Only PDF 417: Minimum aspect of modules in the image (height too wide). |
| Column spacing | Only ECC200 and QR code: Allowable space between two columns, e.g. with nailed codes that have no area-wide modules. |
| Row spacing | Only ECC200 and QR code: Allowable space between two rows. |

### 9.3.12.7 Miscellaneous tab

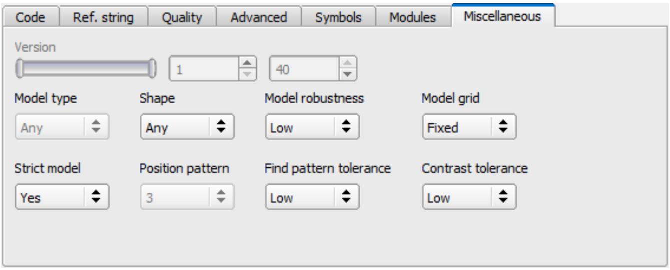



*Fig. 201: Datacode detector, Miscellaneous tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Version | Only QR code: Version of the code, with Model 1 range of values 1 ... 14, with Model 2 range of values 1 ... 40. Version indicates the size of the code. Version 1 means 21 x 21 modules, Version 2 means 25 x 25 modules . . . (4 more modules per version) |
| Search strategy | Determines the number of candidates for reading attempts. If this parameter is set to "Fault tolerant", more potential candidates are taken into account and the probability increases of finding codes that are difficult to read. The "Fast" setting produces a faster reading result for clearly printed codes. |
| Model type | Only QR code: Model 1 or Model 2. Model 2 is the newer version, supporting larger codes. |

| Parameter | Function |
|---|---|
| Shape | Only ECC200 and QR code: Used to specify whether rectangular or quadratic codes should be read. |
| Module robustness | Robustness of the decoding of data codes with very small module size. Setting the parameter to 'high' increases the likelihood of being able to decode data codes with very small module sizes. In addition, in this case, the minimum module size should also be adjusted accordingly, i.e. be set to the assumed minimum module size or module width. |
| Module tab | Only ECC200: Indication of whether the size of the modules may vary within a certain range. Depending on this parameter, different algorithms are used to calculate the module positions. In one case ('fixed'), a fixed grid in which the distances between the module centers are all the same is used. In the other case ('variable'), the grid is aligned on the alternate side of the finder pattern. With 'any', both variants for the grid are tried one after the other. Note that the value of 'module_ grid' is ignored if 'finder_pattern_tolerance' is set to 'high'. In this case, a solid grid is always assumed.<br>List of values: 'fixed', 'variable', 'any' Default: 'fixed' (enhanced: 'any'). |
| Strict model | Specifies whether the entered parameters must be exactly adhered to. If you select "Yes", codes outside of the parameter limits will be ignored. |
| Position pattern | Only QR code: Number of position detection patterns that must be clearly visible in the image in order to search for a code. |
| Search pattern tol-erance | Only ECC200: Tolerance of the search with respect to a disturbed or missing finder pattern. The finder pattern encompasses both the L-shaped and the opposite alternating sides. In one case ('low'), it is assumed that the finder pattern is present to a high degree and shows almost no disturbances. In the other case ('high'), the finder pattern may be heavily disturbed or missing completely without influencing the recognition. It should be noted, however, that in this variant, an increased computing time should be expected. |
| Contrast tolerance | Tolerance in code search for strong local contrast fluctuations. |

### 9.3.13 Detector OCR

This detector is suitable for locating and testing busbars. It is suitable, for example, for reading dot print, as in the automotive industry, and fonts for the pharmaceutical, semiconductor and food industries are also pre-installed.

#### 9.3.13.1 Procedure

The following describes the procedure for setting up an OCR detector for clear text reading step by step. Since the setup steps are based on the results of the previously performed steps, this sequence must be followed for a correct, process-reliable function.

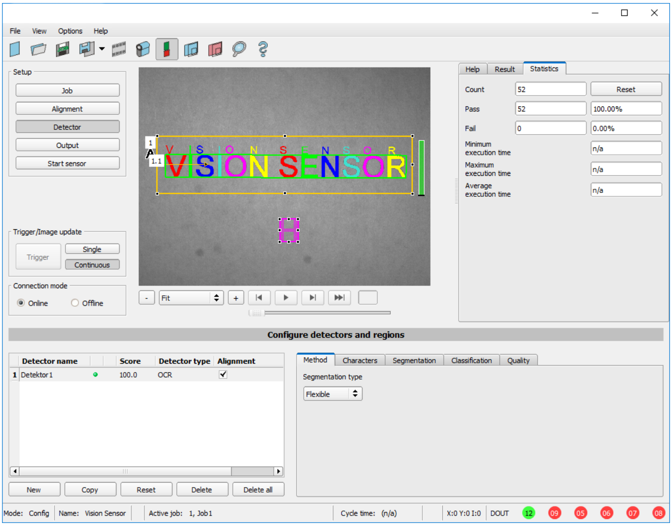


*Fig. 202: Detector OCR*

**Basic sequence of setting parameters**

1. Optimization of the output image; use of the "Pre-processing" tab under "Job".
2. Segmentation (separating characters from the background) using the "Characters" and "Segmentation" tabs.
3. Classification (reading of characters) using the "Classification" tab: selection of a character set, specification of a reference string, optimization of reading results with regular expressions .
4. "Quality" tab: Discard characters that could not be classified with sufficient quality.

**NOTE:**
For the OCR detector, it is not sufficient to adjust the parameters on one single image. Process reliability can only be judged on the largest possible image series - the more images are used, the more stable the result! Save typical image sequences and load them into simulation mode for parameter optimization. Use at least a few dozen images that show the variability of the process.

### Step 1: Optimization of the original image

1. Setup step "Job" / Tab "Image acquisition": Optimize contrast, optimize brightness
2. Adjust external lighting if necessary. If standard illumination is not sufficient for raised or recessed fonts, a better result can be achieved with the "Multishot" option if necessary.
3. The segmentation can be improved using preprocessing filters under "Job" / "Pre-processing" tab.
   E.g. smoothing filters "Gauss", "Mean" (for stable segmentation) or "Dilatation" / "Erosion" or a combination of these.
   (Additional information: Pre-processing tab (Page 93))
4. Display characters to be read as large as possible in the image

### Step 2: Segmentation

1. Select the segmentation method in the "Method" tab: "Flexible" or "Fast" (Additional information: Method tab (Page 261)).
2. Segmentierung optimieren mit den Reitern „Characters“ (Methode: Flexible / Methode: Fast) sowie Reiter „Segmentation“ (Methode Flexible) bzw. "Threshold" (Methode Fast).
   Each segment is displayed in a different color.
   Result: all desired characters must be cleanly segmented.
   **Note:** Welches Symbol jedem segmentierten Zeichen zugeordnet wurde (Ergebnis der Klassifizierung), spielt hier noch keine Rolle.
3. Check correct segmentation of all characters before classification.
   **Note:** Die Klassifizierung hat keinen Einfluss auf die Segmentierung. **Incorrectly segmented** characters are **classified incorrectly**. If the segmentation is unstable despite correct settings, return to step 1 (optimizing the original image).

### Examples: Segmentation

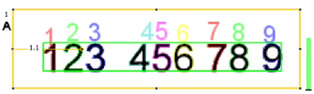



*Fig. 203: Segmentation without default for parameter "Grouping of characters":*

All characters are found.

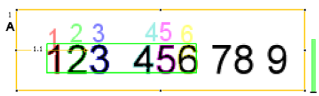
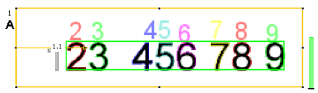
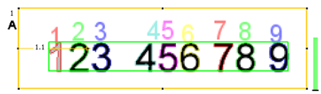
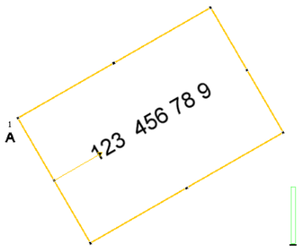
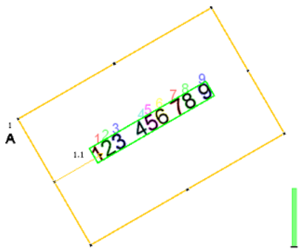
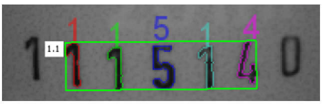
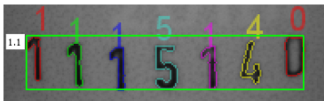



*Fig. 204: Segmentation with default for parameter "Grouping of characters" "3 3":*

Only the two groups of three are segmented.

| | |
|---|---|
| <br>*Fig. 205: Segmentation without default for parameter "Grouping of characters":*<br>The first character "1" is not segmented because its brightness differs too much from the other characters. | <br>*Fig. 206: Segmentation with default for parameter "Grouping of characters" "3 3 2 1":*<br>The character that differs in brightness is also segmented. |
| <br>*Fig. 207: Segmentation with parameter "Text alignment" = "Font horizontal in the image":*<br>No segmentation because characters do not lie horizontally in the image. | <br>*Fig. 208: Segmentation with parameter "Text alignment" = "Font horizontal in search area":*<br>Segmentation works because characters lie horizontally in the search **area**. |
| <br>*Fig. 209: Segmentation with value 15% for parameter "Max. deviation baseline":*<br>Only the inner characters are segmented. | <br>Segmentation with value 25% for parameter "Max. deviation baseline":<br>All characters are segmented. |

**Step 3: Classification**

1. Reiter „Classification“: Geeignete Schrift („Zeichensatz“) auswählen.
2. Each character set is offered with a different number of characters (e.g. numbers, capital letters, special characters).
   Select the character set that best suits the application.
   **Note:** Je größer der Zeichensatz, umso größer die Wahrscheinlichkeit für Fehllesungen – daher kleinsten möglichen Zeichensatz verwenden!
3. Defining Reference string, adding regular expressions.
   The Reference string has two functions:

- **Influencing the classification** using the quality value (reliability)
- **Influence on the detector** result based on the specified minimum quality for the entire character string (threshold).

**Step 4: Quality**

- If the reliability of one of the classified characters is below the threshold (minimum reliability), the detector result becomes negative.
- Low reliability indicates that the character was not classified safely. High reliability, however, is no guarantee for a safe classification!

### 9.3.13.2 Method tab

Settings in the Method tab: Definition of type of segmentation.

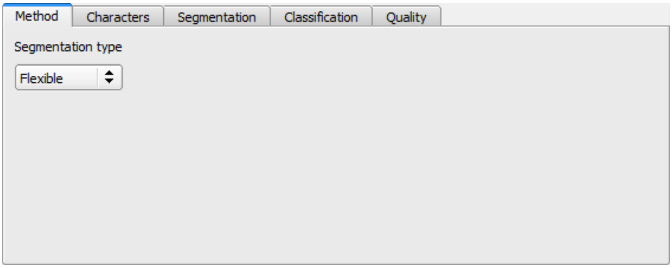



*Fig. 210: OCR Detector, Method tab*

**NOTE:**
The available tabs differ depending on the selected segmentation method (Flexible / Fast).

| Parameter | Function |
| --- | --- |
| Segmentation type: Flexible | **Flexible:**<br>• Only a few parameters adjustable, well suited for users with little experience<br>• Covers a wide range of variations between font and background; also suitable for low-contrast fonts with changing lighting conditions or challenging dot prints<br>• Segmentation searches for character strings, requires at least 3 characters<br>• Background noise can interfere with segmentation and reading quality<br>• Better performance under simple conditions |
| Segmentation type: Fast | **Fast:**<br>• Requires knowledge of image processing<br>• Segmentation takes place via binarization thresholds to separate the characters from the background (BLOB analysis principle)<br>• Also works from 1 character<br>• Depending on the application approx. factor 2-8 faster than "flexible"<br>• only limited suitability for low-contrast lettering or under changing lighting conditions<br>• Background noise can be filtered out by setting parameters<br>• Better performance under difficult conditions |

### 9.3.13.3 Characters tab (Method: flexible)

Settings in the Characters tab: Define the basic settings of the characters to be read.

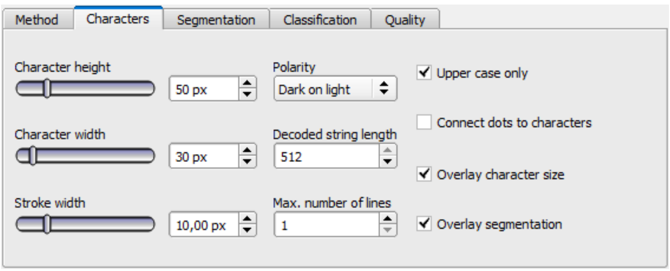



*Fig. 211: Detector OCR, tab Characters*

**Parameter description:**

| Parameter | Function |
|---|---|
| Character height | Max. height of a character in pixels. |
| Character width | Max. width of character in pixels. |
| Stroke width | Average width of lines of characters in pixels. |
| Polarity | Possibility to select between dark characters on bright background or vice versa. |
| Max. number of lines | Max. number of lines to be read |
| Upper case only | Limitation to capital letters only. |
| Connect dots to characters | Connects single dots, e.g. of dot print or a blurry printed font to complete characters |
| Overlay character size | Switches on and off overlay rectangle for size of letters. |
| Overlay segmentation | Switches on and off colored overlay for segmentation of characters |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

**Improvement of execution speed**

- Search range for characters (yellow frame) only as large as necessary

## 9.3.13.4 Segmentation tab (Method: flexible)

Settings in the Segmentation tab: Define the basic settings of the characters to be read.

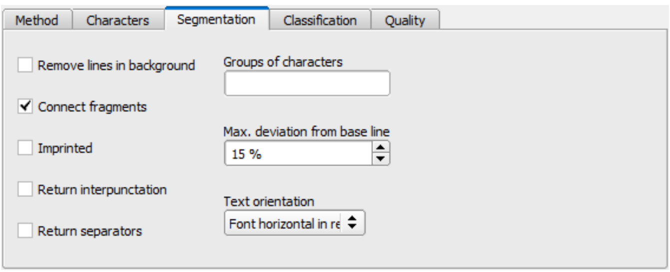



*Fig. 212: OCR Detector, Segmentation tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Suppress back-ground lines | Can be used to hide disturbing lines in the background. |
| Connect fragments | Connects characters that are e.g. fragmented in two parts due to unclean pressure or damage. |
| Raised / imprinted font | Enables the reading of raised or imprinted writing, e.g. if, due to light-ing, the characters e.g. appear in white with a black border (shadow) or vice versa. |
| Return punctuation | Activates output of special characters such as periods or commas. |
| Return separators | Activates output of special characters e.g. dashes. |
| Groups of characters | Allows to specify how the characters in the string to be read are grouped (number of characters per character group). If the characters are e.g. always printed in two groups of four, this can be specified by entering "4 4".<br>This function should be used if different lengths of strings are read in the same picture for several evaluations. |
| Max. deviation from base line | Maximum permissible vertical shift of characters relative to the line (the straight line between the first and last character); specified as a per-centage of the character height. This function can be used when the characters are not printed on a horizontal line. |
| Text orientation | "Font horizontal in image": the font must always be horizontal in the image. Rotated text will not be read or read wrong.<br>"Font horizontal in region": The rotation angle of the search area can be used to specify the rotation of the font with respect to the horizontal. |

### 9.3.13.5 Threshold tab (Method: fast)

Settings in the Threshold tab: Define the basic settings of the characters to be read.

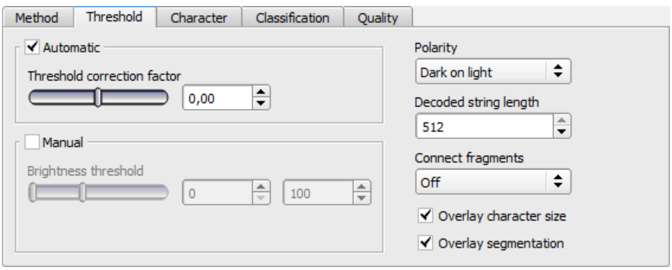



*Fig. 213: OCR detector, Threshold tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Automatic | Activate automatic binarization. The binarization threshold is calculated automatically from the brightness distribution of foreground and back-ground. |
| Correction factor for threshold | The binarization threshold can be shifted in the direction of the fore-ground or the background brightness. |
| Manual | Activate manual binarization. |
| Brightness threshold | Fixed entry of the binarization threshold. |
| Polarity | Possibility to select between dark characters on bright background or vice versa. |
| Max. string length | Maximum allowed length of character string. |
| Connect fragments | Connect fragmented characters from several fragments.<br>Selection: "Off" / "1" / ... / "20" [px]. Number of neighboring pixels: If another segment is found within these pixels, they are combined. |
| Overlay character size | Switches on and off overlay rectangle for size of letters. |
| Overlay segmentation | Switches on and off colored overlay for segmentation of characters |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

### 9.3.13.6 Characters tab (Method: fast)

Settings in the Characters tab: Define the basic settings of the characters to be read.

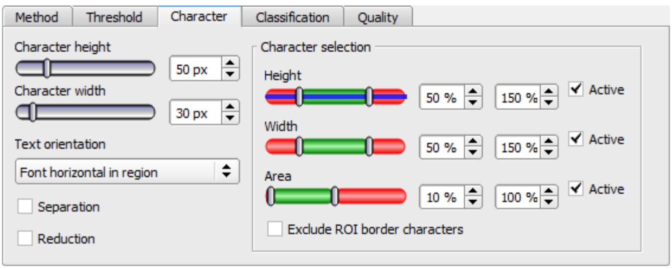



*Fig. 214: OCR Detector, Characters tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Character height | Typical height of a character in pixels. |
| Character width | Typical width of a character in pixels. |
| Text orientation | "Horizontal in the image": The font must always be horizontal in the image. Rotated text will not be read or read wrong.<br>"Horizontal in the search range": The rotation angle of the search range can be used to specify the rotation of the font in relation to the horizontal. |
| Separation | Separates horizontally connected characters that differ significantly from the set character width. |
| Reduction | Reduces segmented areas which have been enlarged by "Connect fragments" to original size. |
| Character selection | Specification of tolerances for the entered character sizes. |
| Height | Specification of tolerances for the defined character height (50% to 150%). Check "Active" to activate this setting. |
| Width | Specification of tolerances to the entered character width (50% to 150%). Check "Active" to activate this setting. |
| Area | Specifying tolerances for the drawing area resulting from the settings (10% to 100%). Check "Active" to activate this setting. |
| No characters on border search range | Discard characters that extend beyond the boundary of the search range. |

**Example: Evaluation of the type face**

Say you want the following type face to be read:

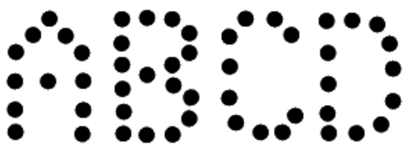
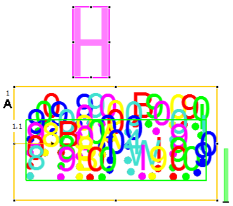
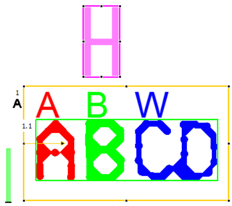
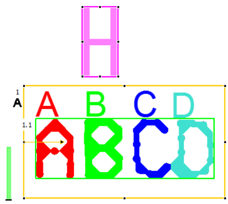


*Fig. 215: Type face*

The following table describes the individual steps used to evaluate the type face.

| Step 1 | Step 2 | Step 3 |
|---|---|---|
| If you place a OCR Detector over the text image and select the "Fast" mode for segmentation in the "Method" tab, an evaluation will be run for each fragment of a letter. | In order for the individual fragments to be put together in order to form letters / characters, you need to select the right value for the "Connect fragments" in the "Threshold" tab. In this example, the "Connect fragments" parameter has been set to 14. | The letters "C" and "D" are still being recognized as the letter "W". The configured character width (pink "H"), however, corresponds to the character width of a letter. To separate the characters, the "Separation" parameter needs to be enabled in the "Characters" tab. |
|  *Fig. 216: "Fast" segmentation mode* |  *Fig. 217: Connect fragments* |  *Fig. 218: Separation* |

### 9.3.13.7 Classification tab

Settings in the Classification tab: Define the basic settings of the characters to be read.

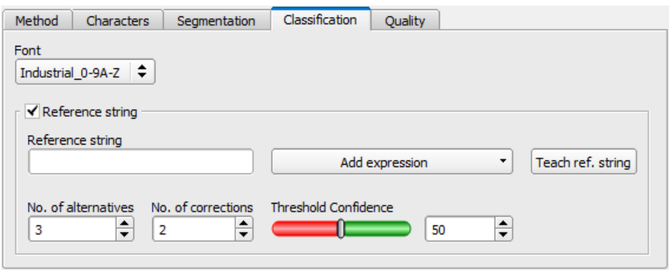



*Fig. 219: Detector OCR, tab Classification*

**Parameter description:**

| Parameter | Function |
|---|---|
| Character set | Available fonts. Additional information: OCR - available fonts<br>0-9 Only digits<br>0-9+ Numbers and special characters<br>A-Z+ Capital letters and special characters<br>0-9A-Z Numbers and capital letters<br>no extension all characters |
| Reference string (Checkbox) | Activates verification of the contents of the read information. The content of the read characters is checked on the basis of regular expressions. |
| Reference string | This text or regular expressions is taken for verification. This is where specific characters can be found that are directly compared or regular expressions to check the structure of the read result.Characters which look very similar as number or as letter, e.g. "8" and "B" can be corrected automatically by using regular expressions in the reference string. Additional information: see below. |
| Add expression | Opens a list with examples for regular expressions. |
| Teach reference string | Reads the code that is currently under the code reader and takes over the read content as a comparison text. This text can be changed later. |
| Number of alternatives | Specifies how many possible alternatives can be searched to find and automatically replace a character according to the regular expression in the reference string. |
| Number of corrections | Max. number of characters that may be changed after being checked by the regular expression. Example: Segmentation is: Day, three-digit (MON / TUE / WED / etc). The decoding outputs the letters "W6O" instead of "WED". With a setting of '2' in this field, the camera software will automatically 'correct' the (number) 6 and (letter) O to become (letter) E and D. If the setting in this field was 1, then the detector would fail. |
| Threshold | Threshold for good/bad decision: If, based on the set threshold, the number of corrections is too high, the entire text is rated as "not read". |

**Reference string: Details**

The Reference string has two functions:

**1. Beeinflussung der Klassifikation**, d.h. der erkannten Zeichen. For each segmented character, a quality value (reliability) is determined in relation to each character contained in the

character set.
Without specification of the reference string, the character with the highest quality value (reliability) is output.
If the reference string is specified, the *n* best alternatives are taken into account (**number of alternatives**).
A maximum of *m* times one character may be selected for the reference string (**number of corrections**), which did not have the maximum reliability.

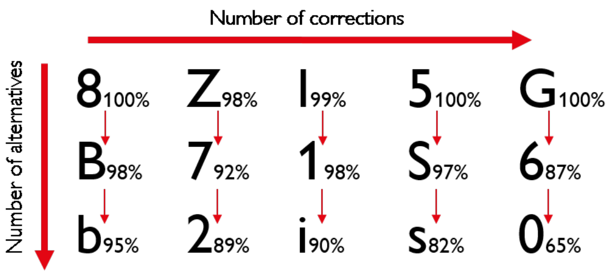



*Fig. 220: Reference string operating mode*

**2. Beeinflussung des Detektorergebnisses**:
A minimum quality for the entire character string (threshold) is specified. If this value is undershot, the detector result is negative.

**Examples of reference character strings defined by regular expressions**

| Reference character string | Hit | Example for hit |
|---|---|---|
| 123 | String containing 123 | 01234 |
| \A123 | String beginning with 123 | 1234 |
| 123\Z | String ending with 123 | 0123 |
| \A123\Z | String matching 123 exactly | 123 |
| [123] | Character string containing one of the characters | 33 |
| [123]{2} | String containing a sequence of 2 of the characters | 23 |
| [12]\|[34] | String containing a character of one of both groups | 4 |

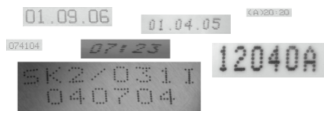
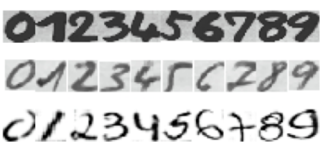
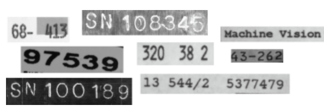
**The most important elements of regular expressions:**

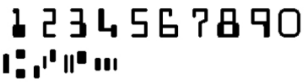
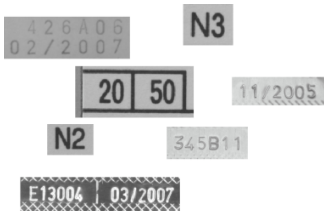
| | |
|---|---|
| ^ or \A | Represents the beginning of the character string |

<table>
<tr><td>$ or \Z</td><td>Represents the end of the character string, and possibly includes a newline as the last character</td></tr>
<tr><td>.</td><td>Represents every character apart from newline</td></tr>
<tr><td>[...]</td><td>Represents any literal listed in the square brackets. If the first character is an '^', the expression is negated. You can use the '-' character, as in '[A-Z0-9]', to specify value ranges. Other characters lose their special meaning within square brackets, except '\'.</td></tr>
<tr><td>*</td><td>Allows 0 or more repetitions of the preceding literal / group</td></tr>
<tr><td>+</td><td>Allows 1 or more repetitions</td></tr>
<tr><td>?</td><td>Allows 0 or 1 repetitions</td></tr>
<tr><td>{n,m}</td><td>Allows n to m repetitions</td></tr>
<tr><td>{n}</td><td>Allows exactly n repetitions</td></tr>
<tr><td>|</td><td>Separates alternative search expressions</td></tr>
</table>

### 9.3.13.7.1 OCR - available fonts

**Overview of fonts:**

<table>
<tr><th>Semi</th><th>Dot print</th></tr>
<tr><td>
</td><td>
</td></tr>
<tr><th>Handwritten</th><th>Industrial</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

<table>
<tr><th>MICR</th><th>OCRA</th></tr>
<tr><td></td><td>0123456789<br>ABCDEFGHIJKLM<br>NOPQRSTUVWXYZ<br>abcdefghijklm<br>nopqrtsuvwxyz<br>-?!/\=+<>.#$%&()@*</td></tr>
<tr><th>OCRB</th><th>Pharma</th></tr>
<tr><td>0123456789<br>ABCDEFGHIJKLM<br>NOPQRSTUVWXYZ<br>abcdefghijklm<br>nopqrtsuvwxyz<br>-?!/\=+<>.#$%&()@*</td><td></td></tr>
</table>

### 9.3.13.8 Quality tab

Definition of basic settings of characters to read.

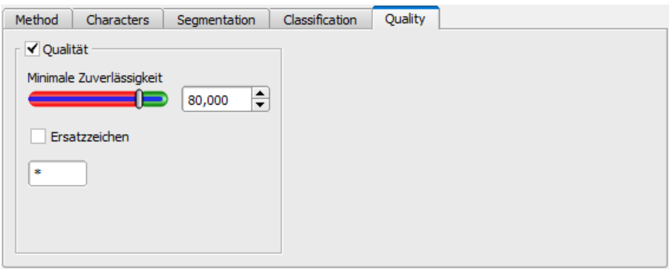



*Fig. 221: Detector OCR, tab Quality*

**Parameter description:**

| Parameter | Function |
|---|---|
| Quality | Quality of each character gets a value of 0-100%. The higher the value, the safer the read character was determined. Small values indicate a rather uncertain assignment. |
| Minimum reliability | If the reliability is below the set threshold, the character is rated unread and replaced with a placeholder. |
| Replacement character | Placeholder in case the minimum reliability was not reached. |

### 9.3.13.9 OCR Result

In the Result tab (next to the help tab or, in run mode, at the bottom of the screen) evaluation results are displayed.

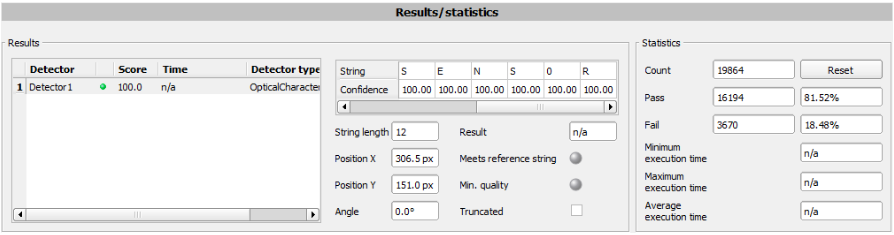



*Fig. 222: Detector OCR, Result display*

**Parameter description:**

| Parameter | Function |
|---|---|
| Score | Detector result: 0% (NOK) or 100% (OK) |
| Text | Characters read |
| Security | Value from 0-100%, indicates how reliably the detector could evaluate a character. |
| String length | Length of string read |
| Position X | X position of the read string in pixels |
| Position Y | Y position of the read string in pixels |
| angle | Angle to the horizontal line |

| Parameter | Function |
|---|---|
| Result | Indication for the quality of a result. If no characters had to be replaced according to the reference string, this value is at 100%. The value decreases with rising number of corrections. |
| Result comparison | Indicates if the output string meets the reference string. |
| Min. Qualität | Indicates if minimum reliability was reached. |
| Truncated | Indicates if a part of the string was truncated. |

## 9.3.14 Detector Color Value

 This detector determines mean color values RGB / HSV / LAB for output via the interfaces.

### 9.3.14.1 Color Channel tab

The Color Channel tab is used to select the Color models (Page 354)/color channel(s) on which the detector is to operate.

An image recorded with a color chip contains more information than a monochrome image due to the color component. This feature can be used with the color channel selection. By selecting the color channels, specific areas can be intensified or weakend. The display of the image depends on the image chip and the selected detector.

- Monochrome chip: Display always black/ white
- Color chip + Color detector: Display always colored
- Color chip + Object detector: Monochrome image, display depending on selected color model and color channel

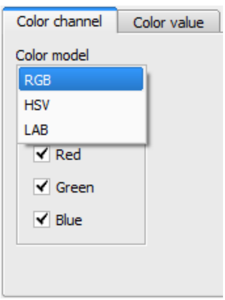



*Fig. 223: Color Channel*

| Parameter | Function |
|---|---|
| Color space | Color spaces:<br>RGB, Color model RGB (Page 355),<br>HSV, Color model HSV (Page 355),<br>LAB, Color model LAB (Page 356) |
| Color channel | One or more channels can be selected. |

### 9.3.14.2 Color Value tab

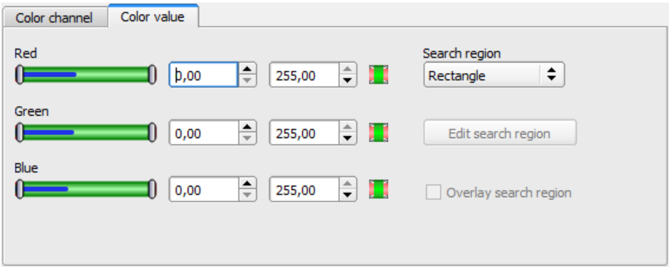



*Fig. 224: Color Value tab*

**Parameter description:**

| Parameter* | Function |
|---|---|
| *differs depending on the setting in color channel | |
| Red (Hue / Lumin-ance) | Schwellen für gewählten Kanal min. / max. |
| Green (Saturation / A) | Schwellen für gewählten Kanal min. / max. |
| Blue (Value / B) | Schwellen für gewählten Kanal min. / max. |
| Search range (shape) | The shape of the search range can be set as Rectangle, Circle, or Free shape. If Freeform is selected, "Edit search range" is active. |
| Edit search range | The parameter "Edit search range" can be used to hide areas of the search area. As with an eraser, the areas that are not needed for the evaluation can be removed in the search range. These marked areas can also be inverted, i.e. the areas that are important for the execution, etc. are marked. |
| Display search range | Enable / disable the display of search range edits |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

**Predestined applications:**

- Output of calculated color parameters via one of the data interfaces for further processing.

## 9.3.15 Detector Color Area

This detector determines the area covered by a color or a color range. Depending on the proportion of the area, a good / bad result can be produced.

### 9.3.15.1 Color Channel tab

For VISOR® Color: See Chapter: Color Channel tab (Page 273)

### 9.3.15.2 Color Area tab

Determines the area covered by a color or color range. Depending on the proportion of the area, a good / bad result can be produced.

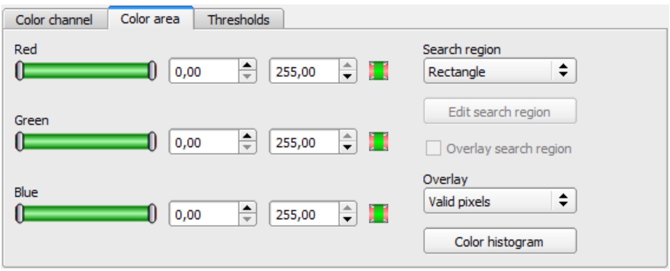



*Fig. 225: Color Area*

**Parameter description:**

| Parameter* | Function |
|---|---|
| *differs depending on the setting in color channel | |
| Red (Hue / Luminance) | Schwellen für gewählten Kanal min. / max. |
| Green (Saturation / A) | Schwellen für gewählten Kanal min. / max. |
| Blue (Value / B) | Schwellen für gewählten Kanal min. / max. |
| Search range (shape) | The shape of the search range can be set as Rectangle, Circle, or Free shape. If Freeform is selected, "Edit search range" is active. |

| Parameter* | Function |
|---|---|
| Edit search range | With the parameter "Edit search range", areas of the search range can be hidden. As with an eraser, the areas that are not needed for the evaluation can be removed in the search range. The marked areas can also be inverted. This marks the areas that are important for the execution. |
| Display search range | Enable / disable the display of search range edits |
| Overlay | Color marking of pixels inside or outside of specified color range. This is an aid during setup to visualize detector results and to set thresholds more accurately. |
| Color histogram | Allows graphical adjustment of the thresholds using a histogram |

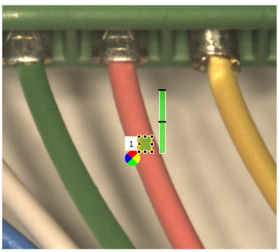
For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

**Predestined applications:**

- Colored object with certain size and variable position in the ROI

### 9.3.15.2.1 Color histogram

Depending on the selected color model, the histograms for RGB, HSV, or LAB are displayed. The histogram shows the distribution of colors in the search range. Using the buttons, single channels can be switched on or off. Small markings below the histogram can be used to move the color detection limits. The marked area is highlighted in the corresponding color. Crossing the limits results in inversion of the selection. If a color can be reliably detected with only one channel, the limit values of the other channels must be set to the lower or upper end value so that they do not interfere with detection.



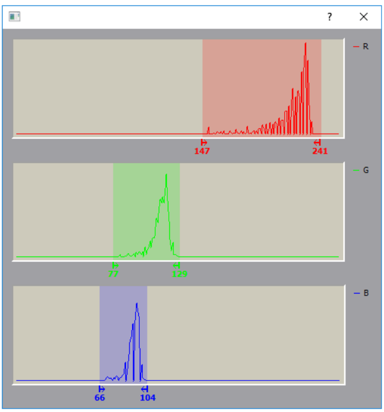


*Fig. 226: Color histogram*

#### 9.3.15.3 Thresholds tab

Determines the area covered by a color or color range. Setting of thresholds.

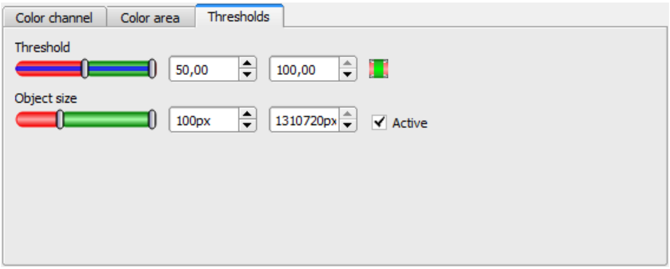



*Fig. 227: Color Area, Thresholds*

**Parameter description:**

| Parameter | Function |
|---|---|
| Threshold | Schwellen für Flächenanteil min. / max. |
| Object size | Min. / Max. Objektgröße (zusammenhängender Farbbereich) |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

### 9.3.16 Detector Color List

This detector compares a color with a list of known colors. Result: Number or name of the closest color. This enables sorting of parts by color.

#### 9.3.16.1 Color Channel tab

For VISOR® Color: See Chapter: Color Channel tab (Page 273)

### 9.3.16.2 Color List tab

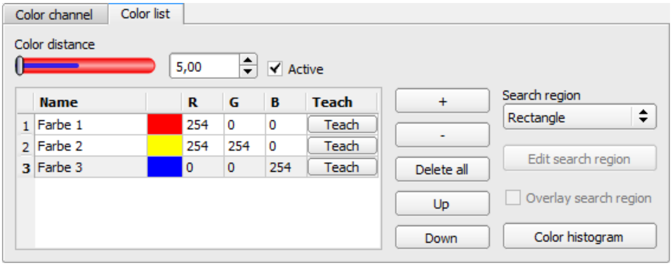



*Fig. 228: Color List*

**Parameter description:**

| Parameter | Function |
|---|---|
| Color distance | Distance of the current color versus the taught-in color.<br>The metric of the color distance depends on the Color models (Page 354) used; only the selected color channels are considered. *1) |
| Name | Name of the color, can be changed by double-clicking on the name, e.g. red, yellow, blue |
| Pattern color | Representation of the taught-in color as a pattern and in numerical values. Differs depending on the setting in the color channel (RGB / HSV / LAB) |
| Teach-in | If you learn the color or the color range in the search range, and if several different colors are to be taught-in, a small search range must be moved to the color to be taught in each case. |
| + | New line at end of the table |
| - | Delete selected line |
| Delete all | Delete all entries in the list |
| Up | Move marked line one line down |
| Down | Move marked line one line down |
| Search range (shape) | The shape of the search range can be set as Rectangle, Circle, or Free shape. If Freeform is selected, "Edit search range" is active. |
| Edit search range | With the parameter "Edit search range", areas of the search range can be hidden. As with an eraser, the areas that are not needed for the evaluation can be removed in the search range. The marked areas can also be inverted. This marks the areas that are important for the execution. |

| Parameter | Function |
|---|---|
| Display search range | Enable / disable the display of search range edits |
| Color histogram | Allows graphical adjustment of the thresholds using a histogram |

1*) In the RGB and LAB color model, the color distance is the Euclidean distance.
In the color model LAB, the color distribution over the entire space is nearly homogeneous, i.e. color differences of the same amount lead to a very similar perception of the color difference over the entire model. Therefore in this model, it can be said that a distance of ≥ 5 leads to the perception of another color.

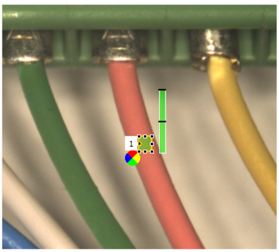
For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

**Predestined applications:**

- Sorting of colored object via the list index
- Simple testing of homogeneous color areas (color is averaged over search range, learn color, set small color distance (tolerance band) ... done)

### 9.3.16.2.1 Color histogram

Depending on the selected color model, the histograms for RGB, HSV, or LAB are displayed. The histogram shows the distribution of colors in the search range. Using the buttons, single channels can be switched on or off. Small markings below the histogram can be used to move the color detection limits. The marked area is highlighted in the corresponding color. Crossing the limits results in inversion of the selection. If a color can be reliably detected with only one channel, the limit values of the other channels must be set to the lower or upper end value so that they do not interfere with detection.



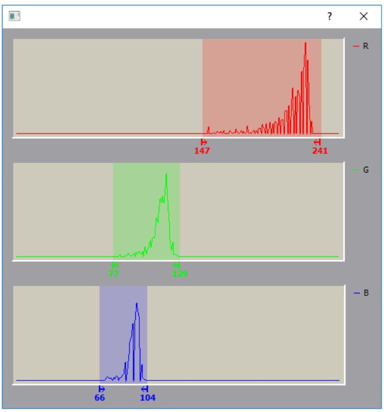


*Fig. 229: Color histogram*

## 9.3.17 Detector Result processing: Text, numbers

This detector allows the calculation and evaluation of detector results from previously executed detectors. This means that the calculation can be performed directly on the VISOR® and the correct functioning does not have to be distributed among different systems (e.g. on the PLC).

In addition to simple arithmetic operations, more complex operations can also be performed, such as sorting output vectors, calculating distances and angles, or logical operations.

The score value of the detector is 100 if **all** expressions are valid. Otherwise, the score value is 0.

### 9.3.17.1 Expressions tab

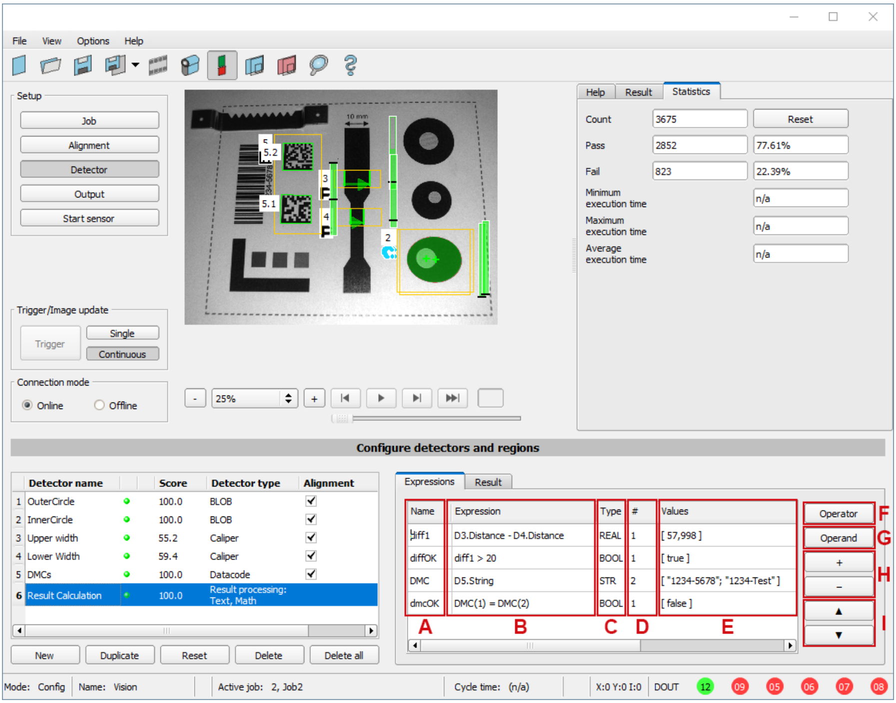



*Fig. 230: Result processing detector, Expressions tab*

The columns and buttons of the detector are described below. The expression in column **B** is evaluated and its result is written into the variable (column **A**). This variable can be accessed in the setup step Output / Telegram.

### Columns

**A**: Name of the expression (changeable), used as a variable. Numbers 0-9, letters A-Z, underscores are allowed, no special characters.

**B**: Expression: consists of operators and operands, can be typed in manually or inserted using the buttons.

**C**: Data types: BOOL (Boolean number), INT (integer), REAL (floating point number), STRING (text).

**D**: Number of components of the result, e.g. for vectors.

**E**: Result of the expression and possible errors that may occur (e.g. "Invalid expression name", "Unexpected symbol", "Undefined variable").

### Buttons

**F**: Operators: divided into groups. Additional information: Chapter Operators.

**G**: Operands: variables, constants and access to detector results (by detector number). Additional information: Chapter Operands.

**H**: + / - adds a new line at the end / deletes the currently selected line.

**I**: ▲ / ▼ moves the current line one position up / down.

### Syntax

| Eingabe | Function |
|---|---|
| D1.Score | Access e.g. to score value of detector no. 1 1 |
| [ ] | Vectors or scalars, spaces possible |
| ; | Separator for vectors |
| "xyz" | Strings |
| (n) | Access to the *n*-th component of a vector |

### Scalars and vectors

The input data and the results can be scalars (single values) or vectors with several components (fields).

Examples:

| Data type | Scalar | Vector |
|---|---|---|
| REAL | [10,543] | [10,543; 2,000; 8,500] |
| INT | [23] | [23; 45; 6] |
| BOOL | [true] | [true; true; false] |
| STRING | ["Object"] | ["Object"; "Code"] |

Individual components of a vector can be accessed by specifying the index. The numbering of the indices begins with 1.

Example: **v1** = [11; 12; 13; 14]

**Single component:**
v1(2) = [12]

**Multiple components:**
v1(2; 4) = [12; 14]

**Component range:**
v1(2:4) = [12; 13; 14]

**Component range to the end:**
v1(2::) = [12; 13; 14]

**Component ranges:**
v1(1; 3:4) = [11; 13; 14]

**Invalid index:**
v1(5) = []

### 9.3.17.1.1 Operators

The operators are divided into the Algeb-raTextgrLogicGeometryouTrigonometryRoundingpsVectorVector properties Sorting, **,** , **and** , **,** , **and** . Within the groups of operators, there are "Advanced" groups containing more complex and less common operators. These subgroups can be expanded by clicking on the arrow (▼).

**NOTE:**
In general, the following Data types can be processed: BOOL (Boolean number), INT (integer), REAL (floating point number), STRING (text). If there are restrictions, these are listed in the respective group or operator.

Operators can be applied:

- to 2 scalars (e.g. 2 + 1 = [3])
- to 2 vectors: Operator acts component by component (e.g. ["a"; "b"] + ["x"; "y"] = ["ax"; "by"].
- to a vector and a scalar: Scalar affects every component of the vector. (e.g. [2; 3] + [1] = [3; 4])

The different operators are described below.

#### 9.3.17.1.1.1 Algebra group

**IN:/OUT:** Data types INT / REAL

| Operator name | Description | Example: |
|---|---|---|
| + | Addition of scalars or of vector components. Corresponding components are added. | [2;3;5]+[4;6;7]<br>=[6;9;12] |
| | Addition of scalar to vector. In this case, the scalar is added to each component of the vector. | [2;3;5]+5<br>=[7;8;10] |
| - | Subtraction of scalars or of vector components. | [4;6;7]-[2;3;5]<br>=[2;3;2] |
| | Subtraction of a scalar from a vector. In this case, the scalar is subtracted from each component of the vector. | [4;6;7]-2<br>=[2;4;5] |
| * | Multiplication of scalars or of vector components. | [4;6;7]*[2;3;5]<br>=[8;18;35] |
| | Multiplication of scalars and vectors. In this case, the scalar is multiplied by each component of the vector. | [4;6;7]*2<br>=[8;12;14] |
| / | Division of scalars or of vector components. | [4;6;7]/[2;3;5]<br>=[2;2;1.4] |
| | Division of vector by scalar. In this case, each component of the vector is divided by the scalar. | [4;6;7]/2<br>=[2;3;3.5] |
| sqr | Square the scalar or vector components. | sqr([2;3;5])<br>=[4;9;25] |
| sqrt | Square root of the scalar or vector components. | sqrt(9)<br>=[3] |
| pow | Power of the scalar or vector components.<br>**IN: base** (base), **expn** (exponent) | pow (2;3)<br>=[8]<br>pow ([2;3;5];3)<br>=[8;27;125] |
| log | Logarithm of the scalar or vector components with base 10. | log(100)<br>=[2] |
| abs | Absolute value of the scalar or vector components | abs(-3.4)<br>=[3.4] |
| min | Returns the smallest vector components (component by component) | min([1;5];[2;4])<br>=[1;4] |

| Operator name | Description | Example: |
|---|---|---|
| max | Returns the largest vector components (component by component) | max([1;5];[2;4])<br>=[2;5] |
| () | Access vector components via their indexes<br>**IN:** Indexes of the desired vector components, Data type INT | [2;4;6;8;10](2:4)<br>=[4;6;8]<br>[2;4;6;8;10](4::)<br>=[8;10] |
| div | Integer division with remainder (outputs integer result)<br>**IN: x** (dividend), **y** (divisor), (Data type INT)<br>**OUT:** Data type INT | div(5;2)<br>=[2] |
| mod | Remainder of an integer division<br>**IN: x** (dividend), **y** (divisor), (Data type INT)<br>**OUT:** Data type INT | mod(5;2)<br>=[1] |

#### 9.3.17.1.1.2 Text group

**IN:/OUT:** Data type STRING

| Operator name | Description | Example: |
|---|---|---|
| "" | Create string variable | "Object"<br>=["Object"] |
| + | Add string vectors | ["V10-"; "V20-"] + ["Solar";"Code Reader"]<br>=["V10-Solar";"V20-Code Reader"] |
| | Add a single string to a vector. In this case, the string will be added to each of the vector components. | ["pick";"place"] + [" object"]<br>=["pick object";"place object"] |
| str_sub | Extract substring. Range refers to the position according to UTF-8 symbols (not to bytes).<br>**IN: v** (String from which a substring is to be extracted),<br>**pos1** (start position, Data type INT),<br>**pos2** (end position, optional, Data type INT)<br>If **pos2** is not specified, the partial string is output to the end. | str_sub ("object";1;3)<br>=["obj"]<br>str_sub ("object";4)<br>=["ect"] |

| Operator name | Description | Example: |
|---|---|---|
| str_insert | Insert a string at a predefined position<br>**IN: v** (String into which a string is to be inserted),<br>**pos1** (position at which the text is to be inserted, Data type INT),<br>**str** (Text to be inserted, Data type STRING) | str_insert ("xxzz";3;"yy")<br>=["xxyyzz"] |
| str_delete | Delete part of a string<br>**IN: v** (String of which a part is to be deleted),<br>**pos** (position at which the text is to be deleted, Data type INT),<br>**len** (number of characters to be deleted, Data type INT) | str_delete ("abcde";4;2)<br>=["abc"] |
| str_replace | Replace parts of a text.<br>**IN: v** (String whose text is to be replaced),<br>**str1** (string to be replaced),<br>**str2** (string to be inserted instead) | str_replace ("abcde";"abc";"x")<br>=["xde"] |
| str_search | Search forward for string and return position of the first result. If string is not found or input string is empty, the result is -1.<br>**IN: v** (string to be scanned)<br>**w** (String to search for) | str_search ("xy-ab-xy";"xy")<br>=[1] |
| str_length | String length: Number of characters (not number of bytes).<br>**IN: v** (string whose length is to be output)<br>**OUT:** Data type INT | str_length(["abcde"])<br>=[5] |
| str_length_byte | String length: Number of bytes for UTF-8 representation (not number of characters).<br>**IN: v** (string whose length is to be output) | str_length_byte("►")<br>=[3] |
| str_correction | Perform text correction using the Reed-Solomon algorithm. Number of check digits: 2.<br>**IN: code** (string to be checked)<br>**OUT:** corrected string, Data type STRING | str_correction ("0110UOL5MI5")<br>=["0100UOL5MI5"] |
| str_correction_errors | Perform text correction using the Reed-Solomon algorithm and output corrected position. Number of check digits: 2.<br>**IN: code** (string to be checked)<br>**OUT:** Index of the corrected position (Data type INT). If no correction is made, the result is an empty vector [ ]. | str_correction_errors ("0110UOL5MI5")<br>=[3]<br>str_correction_errors ("0100UOL5MI5")<br>=[ ] |

| Operator name | Description | Example: |
|---|---|---|
| to_upper | Convert all letters to capital letters<br>**IN: v** (String) | to_upper ("Object")<br>=["OBEJCT"] |
| to_lower | Convert all letters to lower case letters<br>**IN: v** (String) | to_lower ("Object")<br>=["object"] |
| to_string | Convert value / number of type BOOL, INT, REAL to string.<br>**IN: v** (Data type BOOL /INT /REAL),<br>optional: **width** (minimum width of the output string (right-justified), if necessary filled up with blanks, Data type INT),<br>**precision** (accuracy of rounding to decimal places, Data type INT)<br>**OUT:** Data type STRING | to_string ([2.22; 9.99; 5.1]; 4; 1)<br>=[" 2.2"; "10.0"; " 5.1"] |
| to_number | Convert string to number. If the string contains letters, the result is an empty vector [ ].<br>**IN: v** (String)<br>**OUT:** Data type INT, REAL | to_number ("000.123")<br>=[0.123] |
| vec_sum | Link (concatenate) components of the string vector<br>**IN: v** (String vector, Data type STRING),<br>optional: separator (Data type STRING) | vec_sum<br>(["ab";"cd";"ef"];"_")<br>= ["ab_cd_ef"] |

### 9.3.17.1.1.3 Logic group

**Part 1**

When comparing a vector with a scalar, the scalar is compared with each component of the vector.

**IN:** Data type BOOL / INT / REAL / STRING

**OUT:** Data type BOOL

| Operator name | Description | Example: |
|---|---|---|
| < | "Less than" operation of scalars or vector components | [2;5;4] < [2;4;5]<br>=[false;false;true]<br>["A"]<["B"] = [true] |
| <= | "Less than or equal to" operation of scalars or vector components | [2;5;4] <= [2;4;5]<br>=[true;false;true] |
| > | "Greater than" operation of scalars, or vector components | [2;5;4] > [2]<br>=[false;true;true] |

| Operator name | Description | Example: |
|---|---|---|
| >= | "Greater than or equal to" operation of scalars or vector components | [2;5;4] >= [2]<br>=[true;true;true] |
| = | "Is equal" operation of scalar or vector components | ["OK";"NOK"] = ["OK"]<br>=[true;false] |
| != | "Is unequal" operation of scalar or vector components | ["OK"] != ["NOK"]<br>=[true] |

**Part 2**

For the following operators: **IN:/OUT:** Data type BOOL
Example vectors: **v1**=[true;true] **v2**=[true;false] **v3**=[true]

<table>
<tr><th>Operator name</th><th>Description</th><th>Example:</th></tr>
<tr><td rowspan="2">&</td><td>AND operation of scalars or vector components</td><td>v1&v2<br>=[true;false]</td></tr>
<tr><td>AND operation of scalar and vector. In this case, the scalar is combined with each component of the other vector.</td><td>v2&v3<br>=[true;false]</td></tr>
<tr><td rowspan="2">|</td><td>OR operation of scalars or vector components</td><td>v1|v2<br>=[true;true]</td></tr>
<tr><td>OR operation of scalar and vector. In this case, the scalar is combined with each component of the other vector.</td><td>v2|v3<br>=[true;true]</td></tr>
<tr><td>!</td><td>Negation (NOT) of scalars or vector components</td><td>!v2<br>=[false;true]</td></tr>
<tr><td>if</td><td>Check a condition and issue a value accordingly<br>IN: b (condition to be tested, Data type BOOL, must be scalar),<br>v1 (Then value), v2 (Otherwise value), Data types BOOL / INT / REAL / STRING.<br>Output of v1 if b=true, output of v2 if b=false<br>Note: Both v1 (then value) and v2 (otherwise value) must be executable. Otherwise, the expression cannot be executed.</td><td>if(v3; "OK"; "NOK")<br>=["OK"]</td></tr>
</table>

#### 9.3.17.1.1.4 Geometry group

**IN:/OUT:** Data types INT / REAL

**Overlays for geometry operators**

For some Geometry operators there are graphical overlays in the image. The overlays are displayed for the expression that is currently active (if there are several Geometry operators in one line, only the first one is displayed).

These overlays are also only displayed if **no** calibration is active or a **2D** calibration (i.e. "Scaling (Measurement)", "Calibration plate (Measurement)" or "Calibration plate (Robotics)").

The input positions are shown with green crosses and a small number next to them (this number indicates the parameter, e.g. "1" for x1, y1).

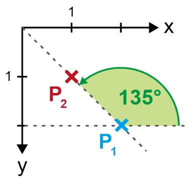
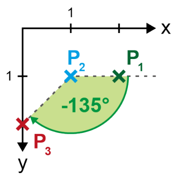
For distance calculations ("distance" and "nn_distance") the distances are drawn in.

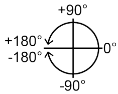
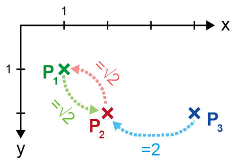
For angles, the respective straight lines and the angle are drawn in.

- For the "angle_2points" angle: X axis and the line defined by the 2 entered points.
- For the "angle_3points" angle: the two straight lines defined by the 3 entered points.

If a calibration is active, the units set in the calibration are used and the corresponding coordinate system is used for angle calculation.

| Operator name | Description | Example: |
|---|---|---|
| distance | Calculate distance between 2 points<br>**IN: x1**, **y1**, **x2**, **y2** | distance(1;1;4;1)<br>=[3] |

| Operator name | Description | Example: |
|---|---|---|
| angle_ 2points | Angle between the X axis and the line given by 2 points. Value range -180° to +180°.<br>(Direction of rotation: see graphic below)<br>If the calibration is active, the calculation is performed in the World coordinate system. If the calibration is inactive, the calculation is performed in the Image Coordinate System.<br>**IN: PosX1**, **PosY1**, **PosX2**, **PosY2**<br>**POSX1=2** **POSX2=1**<br>**POSY1=2** **POSY2=1**<br>**P1 (2\|2)** **P2 (1\|1)**<br><br>angle_2points (2;2;1;1) = [135]<br>*Fig. 231: Example for angle_2points without calibration (IF)* | angle_2points (2;2;1;1) =[135] |
| angle_ 3points | Angle between 2 straight lines given by 3 points. Value range -180° to +180°.<br>(Direction of rotation: see graphic below)<br>If the calibration is active, the calculation is performed in the World coordinate system. If the calibration is inactive, the calculation is performed in the Image Coordinate System.<br>**IN: PosX1**, **PosY1**, **PosX2**, **PosY2**, **PosX3**, **PosY3**<br>**POSX1=2** **POSX2=1** **POSX3=0**<br>**POSY1=1** **POSY2=1** **POSY3=2**<br>**P1 (2\|1)** **P2 (1\|1)** **P3 (0\|2)**<br><br>angle_3points (2;1;1;1;0;2) = [-135]<br>*Fig. 232: Example for angle_3points without calibration (IF)* | angle_3points (2;1;1;1;0;2) =[-135] |

| Operator name | Description | Example: |
|---|---|---|
| angle_diff | Difference between 2 angles. The function returns the smaller distance within the circle (or the specified period) with a sign. Direction of rotation (IF):<br><br><br>**IN: a1**, **a2** (angles 1 and 2),<br>optional: **period** (period: possible values 90, 180 or 360; 360 is the default value)<br>For period = 180, the results are between -90 and 90.<br>For period = 90, the results are between -45 and 45. | angle_diff (200;10;360) =[170]<br>angle_diff (200;10; 90) =[-10] |
| nn_distance | Calculate distance to nearest neighbor for each of the points<br>**IN: x**, **y** (vectors with the same number of vector components (≥2))<br>**x = [1;2;4]**<br>**y = [1;2;2]**<br>→ $P_1$ (1\|1) $P_2$ (2\|2) $P_3$ (4\|2)<br><br><br>nn_distance (x,y) = [√2;√2;2] | nn_distance (x,y) = [ 1.414; 1.414; 2.000 ] |
| nn_ distance_idx | Calculate the index of the nearest neighbor for each of the points: "At which position is the nearest point to this one?"<br>**IN: x**, **y** (vectors with the same number of vector components (≥2))<br>**OUT:** Data type INT | x=[1;2;4] y=[1;2;2] nn_distance_idx (x;y) =[2;1;2] |

### 9.3.17.1.1.5 Trigonometry group

Note: The calculations are performed based on degree values (not in radians).

**IN:/OUT:** Data type REAL

| Operator name | Description | Example: |
|---|---|---|
| sin | Sine of the scalar or vector components | sin(90)<br>=[1] |

| Operator name | Description | Example: |
|---|---|---|
| cos | Cosine of the scalar or vector components | cos(180)<br>=[-1] |
| tan | Tangent of the scalar or vector components | tan(45)<br>=[1] |
| arcsin | Arc sine of the scalar or vector components<br>**IN:** Data type REAL -1 ≤ **v** ≤ 1 | arcsin(0,5)<br>=[30] |
| arccos | Arc cosine of the scalar or vector components<br>**IN:** Data type REAL -1 ≤ **v** ≤ 1 | arccos(0,5)<br>=[60] |
| arctan | Arc tangent of the scalar or vector components | arctan(1)<br>=[45] |
| arctan2 | Arc tangent of the scalar or vector components with 2 arguments | arctan2(1;0)<br>=[90] |
| to_degree | Convert radian to degrees | to_degree (pi)<br>=[180] |
| to_radian | Convert degrees to radian | to_radian (180)<br>=[3.142] |

### 9.3.17.1.1.6 Rounding group

**IN:** Data type REAL

**OUT:** Data types REAL / INT

| Operator name | Description | Example: |
|---|---|---|
| round | Rounding of the scalar or vector components<br>**IN:** optional: **prec** (precision of rounding to decimal places, Data type INT; no specification or "0" leads to rounding to integer) | round (3.667;1)<br>=[3.7] |
| ceil | Ceiling function: INT value greater than or equal to the scalar or vector components | ceil ([2.3;-3.5])<br>=[3;-3] |
| floor | Floor function: INT value less than or equal to the scalar or vector components | floor ([2.3;-3.5])<br>=[2;-4] |
| trunc | Create INT-value by truncating the scalar or the vector components (truncating the decimal places) | trunc ([2.3;-3.5])<br>=[2;-3] |

### 9.3.17.1.1.7 Vector group

**IN:**/**OUT:** Data types BOOL / INT / REAL / STRING

Example vector: **v1**=[2;3;5] **v2**=[20;30;50]

| Operator name | Description | Example: |
|---|---|---|
| [ ] | Create vector | [2;3;5] |
| | Link (concatenate) vectors | [[2;3];[4;6]]<br>=[2;3;4;6] |
| : | Create vector with components from "**i**" to "**j**"<br>**IN: i**, **j** (vector components from/to, Data type INT, values ≥ 0) | 0:3<br>= [0;1;2;3] |
| :: | Return range of vector components up to the end of the vector<br>**IN:** Index of the desired vector components, Data type INT | v1(**2**::)<br>=[3;5] |
| new | Create new vector by specifying length and default value<br>**IN: length** (number of vector components, Data type INT)<br>**v0** (value) | new(3;1.2)<br>=[1.2; 1.2; 1.2]<br>new(3;"xy")<br>["xy";"xy";"xy"] |
| size | Return the number of components of the vector<br>**OUT:** Data type INT | size(v1)<br>=[3] |
| concat | Link (concatenate) vectors<br>**IN: v**, **w** (vectors) | concat (v1;[0;0;0])<br>=[2;3;5;0;0;0] |
| interleave | Interleave vector components<br>**IN: v**, **w** (vectors) | interleave (v1;v2)<br>=[2;20;3;30;5;50] |
| () | Access vector components via their indexes<br>**IN: v** (vector), **w** (index of the desired vector components, Data type INT) | v1(3)<br>=[3] |
| bound | Return values that are within the specified range. If the lower or upper limit is irrelevant, use **[ ]** for it.<br>**IN: v** (Data type REAL), **vmin**, **vmax** (upper/lower limit, Data type REAL)<br>**OUT:** Data type REAL | bound (v1; 3; 6)<br>=[3; 5]<br>bound (v1; 3.5; [ ])<br>=[5] |

| Operator name | Description | Example: |
|---|---|---|
| lowerbound | Return vector components which are ≥ the lower bound<br>**IN: v** (vector or scalar, Data type REAL), **vmin** (lower bound) | lowerbound ([2;4;6];5)<br>=[6] |
| upperbound | Return vector components which are ≤ the upper bound<br>**IN: v** (vector or scalar, Data type REAL), **vmax** (upper bound) | upperbound ([2;4;6];5)<br>=[2;4] |
| bound_idx | Return indexes of vector components whose values are within the specified range. If the lower or upper limit is irrelevant, use **[ ]** for it.<br>**IN: v** (Data type REAL), **vmin**, **vmax** (upper/lower bound, Data type REAL)<br>**OUT:** Data type INT | bound_idx (v1;3;5)<br>=[2;3]<br>bound_idx ([2;7;5;3;4]; 4; [ ])<br>=[2;3;5] |
| select | Access vector components via their indexes | select(v1;1)<br>=[2] |

### 9.3.17.1.1.8 Vector properties group

This group provides operators to combine ("aggregate") all components of a vector. These operators start with the abbreviation "v_" to distinguish them from operators of the same name, which process vectors component by component.

**IN:/OUT:** Data type REAL
Example vector: **v1**=[2;4;5]

| Operator name | Description | Example: |
|---|---|---|
| vec_sum | Sum of the vector components | vec_sum(v1)<br>=[11] |
| vec_product | Product of the vector components | vec_product(v1)<br>=[40] |
| vec_mean | Mean of the vector components | vec_mean(v1)<br>=[3.667] |
| vec_stddev | Standard deviation of the vector components | vec_stddev(v1)<br>=[1,528] |
| vec_median | Median of the vector components | vec_median(v1)<br>=[4] |

| Operator name | Description | Example: |
|---|---|---|
| vec_ median_idx | Index of the median of the vector components<br>**OUT:** Data type INT | vec_median_idx ([4;2;5])<br>=[1] |
| vec_min | Returns the smallest of the vector components (by aggregating) | vec_min(v1)<br>=[2] |
| vec_min_idx | Index of the minimum of the vector components<br>**OUT:** Data type INT | vec_min_idx(v1)<br>=[1] |
| vec_max | Returns the largest of the vector components (by aggregating) | vec_max(v1)<br>=[5] |
| vec_max_ idx | Index of the maximum of the vector components<br>**OUT:** Data type INT | vec_max_idx(v1)<br>=[3] |
| vec_and | AND operation within a vector<br>**IN:/OUT:** Data type BOOL | vec_and([true;false])<br>=[false] |
| vec_or | OR operation within a vector<br>**IN:/OUT:** Data type BOOL | vec_or([true;false])<br>=[true] |

### 9.3.17.1.1.9 Sorting group

Example vector: **v1**=[2;4;5]

| Operator name | Description | Example: |
|---|---|---|
| sort | Sort vector in ascending order. | sort (["z";"x";"y"])<br>=["x";"y";"z"] |
| sort_idx | Return indexes of the vector components according to their size (ascending order): "At which position is the smallest / ... / greatest vector component?"<br>**OUT:** Data type INT | sort_idx ([4;5;2])<br>=[3;1;2] |
| sort_by_idx | Sort vector by specified index vector<br>**IN:** **v** (Vector ), **idx** (Vector: Data type INT) | sort_by_idx([v1];[2;3;1])<br>=[4;5;2] |
| invert | Inverse order of vector components | invert ([2;5;4;1])<br>=[1;4;5;2] |

### 9.3.17.1.2 Operands

The "Operands" button can be used to access detector results and variables, and constants can be inserted.

Further information on the results of the individual detectors can be found in the Communications manual: Chapters Data output in ASCII / Data output in BINARY

| Operand | Description |
|---|---|
| Detector (detector name) | Access to all detector results, these can also be accessed in the telegram (setup step Output / Telegram)<br>Only those detectors that are listed **before** / **above** the current detector in the detector list can be accessed by the Result processing (i.e. which were first created or moved up).<br>To access a result, the detector number is used in the expression, e.g.: "D1.Score". |
| Variables | Access to results of expressions (only from this Result processing detector), which are **before** / **above** the current expression (by default "v1", "v2", ...) |
| Constants | Inserting the constants "true", "false", "pi", "e". |

### 9.3.17.2 Result tab

The "Result" tab defines how the detector result (green or red LED) is generated.


*Fig. 233: Detector Result processing, "Result" tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Result expression | • "All expressions valid" (default): By default the detector is "OK" (green) if all expressions are valid.<br>• Select an expression from the drop-down list (only expressions of Data type BOOL are displayed):<br>The detector can thus be assigned the result of a Boolean variable. It is displayed accordingly:<br>for "true" = green / "OK"<br>for "false" = red / "NOK". |

### 9.3.17.3 Application examples: "Result processing" detector

### 9.3.17.3.1 Examples "Result processing: Math"

**Example 1: Simple calculations and checks**

- Check whether the label of a package is centered and the angle of the label is correct (in the example demonstrated here: position X)
- Detect the position of the packaging and the label with "Contour" detector.
- Determine difference and check thresholds


*Fig. 234: Case 1: Label correctly applied*

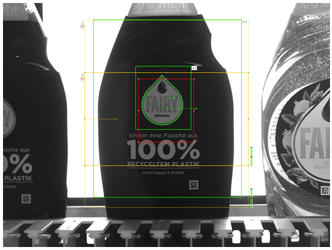



*Fig. 235: Case 2: Label not applied correctly*

Requirements and settings in SensoConfig:

- Detector 1 (**D1**): Contour detector for detecting the bottle
- Detector 2 (**D2**): Contour detector for detecting the label
  Both detectors output values for X position.
- Detector 3: Result processing: Math
- Then select the expression "Result" in the "Result" tab, so that the overall detector result is output accordingly.

Result processing - Expressions:

Case 1: Label correctly applied

| Name | Expression | Type | # | Values |
|---|---|---|---|---|
| xOffset | abs(**D1**.PosX - **D2**.PosX) | REAL | 1 | [3,745] |
| Result | xOffset < 4 | BOOL | 1 | **[true]** |

Case 2: Label not applied correctly

| Name | Expression | Type | # | Values |
|---|---|---|---|---|
| xOffset | abs(**D1**.PosX - **D2**.PosX) | REAL | 1 | [18,178] |
| Result | xOffset < 4 | BOOL | 1 | **[false]** |

**Example 2: Calculate 2D distances**

- Check if the white circle is centered within the black circle
- Detecting center of gravity with "BLOB" detector
- Calculate distance

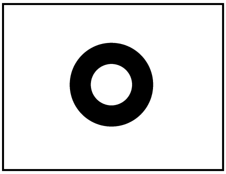


*Fig. 236: Case 1: White circle centered*

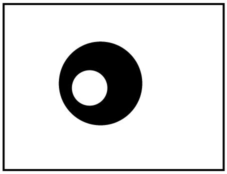


*Fig. 237: Case 2: White circle not centered*

Requirements and settings in SensoConfig:

- Detector 1 (**D1**): BLOB detector for detecting the black circle
- Detector 2 (**D2**): BLOB detector for detecting the white circle
- For D1 and D2, in the "Features" tab: Activate C1 Circuit > Pos. X and C1 Circuit > Pos. Y
- Detector 3: Result processing: Math

Result processing - Expressions:

Case 1: White circle centered

| Name | Expression | Type | # | Values |
|---|---|---|---|---|
| CenterDistance | distance(**D1**.C1_PosX; **D1**.C1_PosY; **D2**.C1_PosX; **D2**.C1_PosY) | REAL | 1 | [0,045] |
| Threshold | CenterDistance < 1 | BOOL | 1 | **[true]** |

Case 2: White circle not centered

| Name | Expression | Type | # | Values |
|---|---|---|---|---|
| CenterDistance | distance(**D1**.C1_PosX; **D1**.C1_PosY; **D2**.C1_PosX; **D2**.C1_PosY) | REAL | 1 | [77,822] |
| Threshold | CenterDistance < 1 | BOOL | 1 | **[false]** |

Then select the expression "Threshold" in the "Result" tab, so that the overall detector result is output accordingly.

### 9.3.17.3.2 Examples "Result processing: Text"

**Example 3: Text comparison**

- Check whether the content of the DataMatrix-Code matches the content of the barcode
- Output result as result on digital switching output


*Fig. 238: Comparison of DataMatrix code and barcode*

Requirements and settings in SensoConfig:

- Detector 1 (**D1**): Datacode detector
- Detector 2 (**D2**): Barcode detector
- Detector 3: Result processing: Text

Result processing - Expressions:

| Name | Expression | Type | # | Values |
|---|---|---|---|---|
| DMC_ Result | **D1**.String | STR | 1 | ["543-11024"] |
| Barcode_ Result | **D2**.String | STR | 1 | ["548-11024"] |
| Result | DMC_Result == Barcode_Result | BOOL | 1 | **[false]** |

Then select the expression "Result" in the "Result" tab, so that the overall detector result is output accordingly.

**Example 4: Sorting the Results Output Based on Position**

- Output the results of several codes based on their Y-position from top to bottom

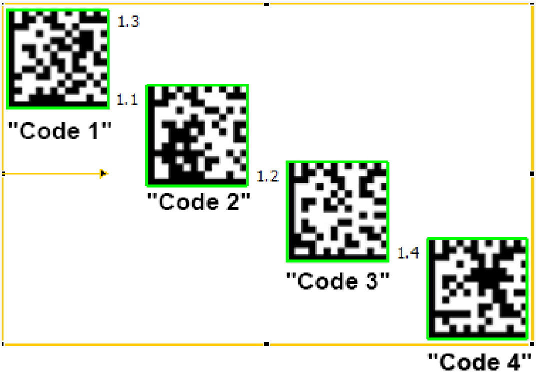



*Fig. 239: Sorting of data codes*

Requirements and settings in SensoConfig:

- Detector 1 (**D1**): Datacode detector
  Here: "Max. number of codes" Parameter is set to 4.
- Detector 2: Result processing: Text

Result processing - Expressions:

| Name | Expression | Type | # | Values |
|---|---|---|---|---|
| ResultString | **D1**.String | STR | 4 | ["Code 2"; "Code 3"; "Code 1"; "Code 4"] |
| yPosition | **D1**.PosY | REAL | 4 | [359; 564; 154; 772] |
| IndexPos | sort_idx(yPosition) | INT | 4 | [3; 1; 2; 4] |

| Name | Expression | Type | # | Values |
|---|---|---|---|---|
| Result | sort_by_idx(ResultString; IndexPos) | STR | 4 | ["Code 1"; "Code 2"; "Code 3"; "Code 4"] |

## 9.3.18 Detector Wafer

This detector is suitable for inspecting breakouts on the edges of wafers or cells during production and for measuring geometric parameters such as width, height, position, angle of rotation, etc. It is extremely accurate in measuring the size and position of the wafer and can be used as a tool to pick up and place robotic systems.

**NOTE:**
The tabs "Binarization", "Rectangle fit" and "Miscellaneous" are only accessible in the expert mode. Activation via menu bar "Options / Expert Mode".

**The VISOR® Solar is characterized by the following properties:**

- Automatic recognition of wafer and cell geometry
- Reliable detection of imperfections on straight and curved contours
- Flexible setting of the test criteria: e.g. tolerances for wafer size and rotational position, size, and number of allowed contour defects
- Easy optimization of the sensor with regard to evaluation speed and inspection accuracy (subpixel method)
- Free cutting function for larger defects
- Distortion removal

### 9.3.18.1 Wafer tab

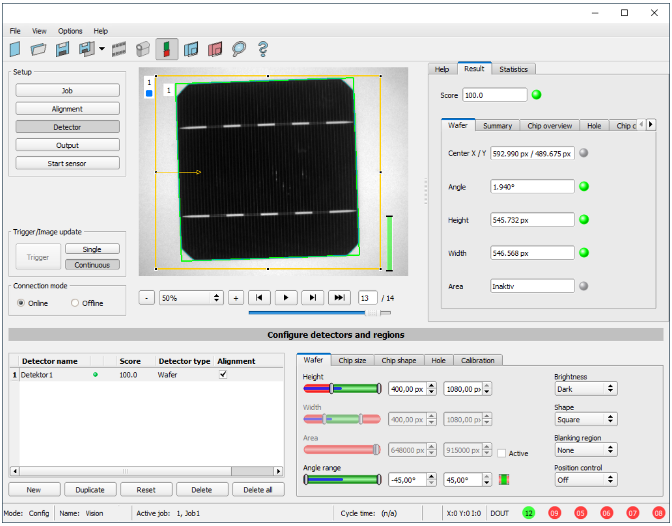



*Fig. 240: Detector Wafer, Wafer tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| height | Range to accept the height of a wafer. |
| Width | Range to accept the width of a wafer. |
| Area | Range to accept the area of a wafer. |
| Angle range | Range to accept the current value of rotation. |
| Brightness | Select the brightness of your object comparing to the background. |
| Shape | Choice between rectangular or square wafer shape |
| Blanking region | This selection box allows up to 12 rectangular areas to be positioned freely in the image. The areas within these zones are not used for wafer control. |
| Position control | To control the position of the wafer center of gravity, a rectangle or an ellipse can be freely positioned in the image area as the desired area. |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

#### 9.3.18.2 Chip size tab

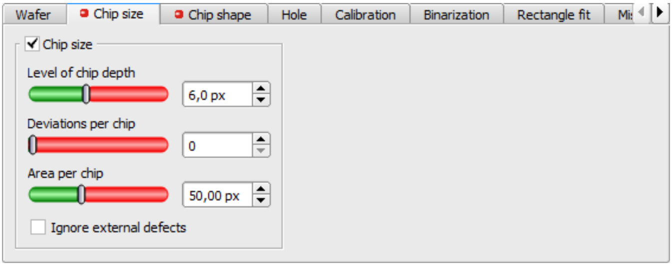



*Fig. 241: Detector Wafer, Chip size tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Chip size | Activate chip size. |
| Threshold of chip depth | From the detected contour points, an ideally approximated, symmetrical right-hand contour is calculated. Based on this, the distances are calculated for all contour points. The threshold "max. deviation" defines the value for a faulty distance. |
| Deviations per chip | Defines the max. number of faulty distances for GOOD / BAD detection. |
| Area per chip | Defines the threshold of a faulty area for GOOD / BAD detection. |
| Ignore external defects (Expert mode) | All detected error contour points that lie outside the box enclosing the wafer (color: turquoise) are not taken into account during the contour check. |

#### 9.3.18.3 Chip shape tab

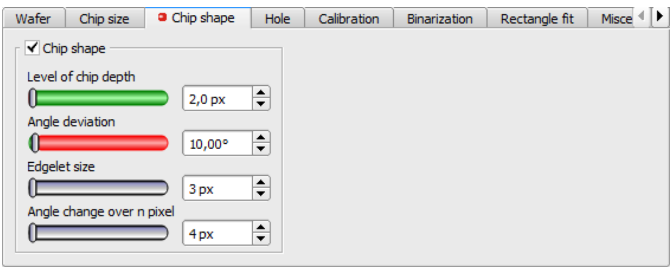



*Fig. 242: Detector Wafer, tab Chip shape*

**Parameter description:**

| Parameter | Function |
|---|---|
| Chip shape | Activate chip shape. |
| Threshold of chip depth | If a protrusion has been detected and there is a chip in the fault region, then the internal errors below the set deviation (in pixels, relative to the enclosing box) are classified as faults. |
| Angle deviation | A contour point is detected as an error if the calculated angle change is above the threshold value. |
| Edge segment (Expert mode) | For each individual contour point, an edge segment (length, rotational position in the image) is determined on the basis of two adjacent contour points. Parameter: Distance to the neighboring points. |
| Angle change over n pixels (Expert mode) | For each individual contour point, the maximum difference of the rotational positions of the associated n edge segments is determined from n adjacent contour points. |

### 9.3.18.4 Hole tab

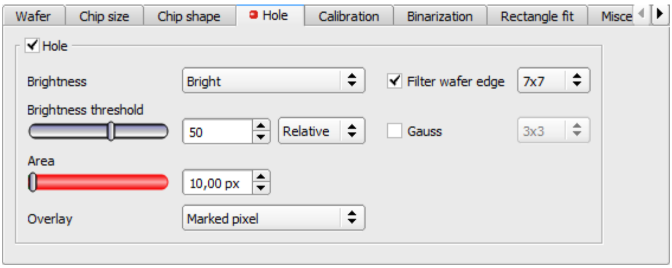



*Fig. 243: Detector Wafer, tab Hole*

**Parameter description:**

| Parameter | Function |
|---|---|
| Hole | Activate hole recognition. |
| brightness | Select the brightness of the object based on the brightness of the wafer. |
| Brightness threshold / absolute | Define the intensity threshold to detect an faulty object as a fixed gray value. |
| Brightness threshold / relative | Defining the brightness threshold for object recognition as an offset value in addition to the dynamically determined, average gray value of the wafer. |

| Parameter | Function |
|---|---|
| Area | Definition of the minimum hole / object area to be detected. Value in pixels * pixels or calibrated in mm * mm. |
| Overlay | (De)activation of the overlay of detected objects. |
| Edge filter (Expert mode) | Extending dark areas, eliminating bright pixels in dark areas, eliminating artifacts, separating bright objects. |
| Gauss (Expert mode) | Reduction of noise, suppression of spurious details and artifacts, smoothing of edges. |

### 9.3.18.5 Calibration tab

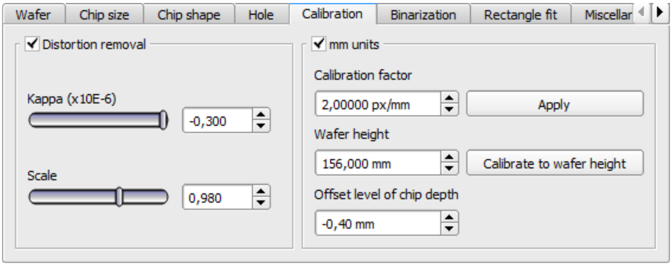



*Fig. 244: Detector Wafer, tab Calibration*

**Parameter description:**

| Parameter | Function |
|---|---|
| Distortion removal | Activate distortion removal. |
| • Kappa (x10E-6) | Distortion coefficient for modeling the radial distortion. |
| • Scaling | Multiplicative correction. |
| mm units | Activate mm units. |
| • Calibration factor | Pixel per mm; calibration factor to convert image data into world data. |
| • Apply | By pressing "Apply", the dimensions in the other tabs are automatically adjusted to the new calibration factor. |
| • Wafer height | Program automatically adjusts the calibration factor based on the measured wafer height in pixels. |
| • Calibrate to wafer height | Calibration factor is calculated from the value "Wafer height". |

| Parameter | Function |
|---|---|
| • Offset chip depth | Correction factor for the measured value of chip depth. The factor is added to the actual measured value. |

### 9.3.18.6 Binarization tab

**NOTE:**
The tabs Binarization, Rectangle fit and Miscellaneous are only accessible in the expert mode. Activation via menu bar "Options / Expert Mode".

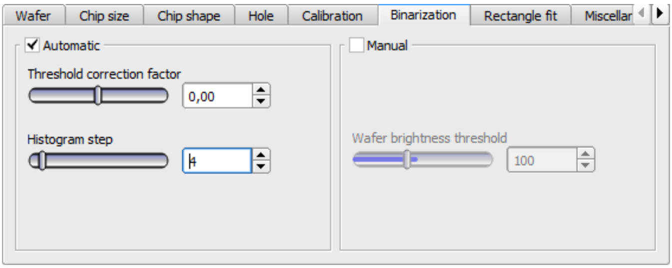



*Fig. 245: Detector Wafer, tab Binarization*

**Parameter description:**

| Parameter | Function |
|---|---|
| Automatic | Activate automatic binarization. |
| Correction factor for threshold | The threshold value for contour detection is calculated automatically from the brightness distribution of foreground and background. The position of the contrast threshold in the histogram can be shifted in the direction of the foreground or the background color. |
| Histogram increment | Gray scale value resolution of histogram history. |
| Manual | Activate manual binarization. |
| Wafer brightness threshold | Fixed entry of brightness threshold. |

### 9.3.18.7 Rectangle fit tab

**NOTE:**
The tabs Binarization, Rectangle fit and Miscellaneous are only accessible in the expert mode. Activation via menu bar "Options / Expert Mode".

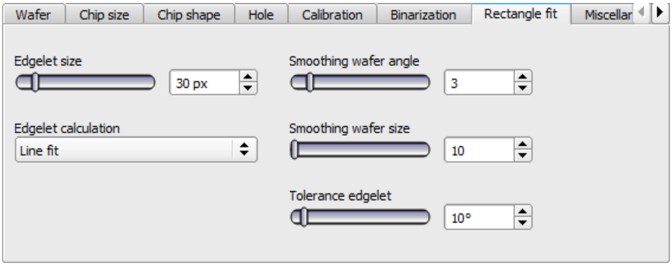



*Fig. 246: Detector Wafer, Rectangle fit tab*

Parameter description:

| Parameter | Function |
|---|---|
| Edge segment | Increment or number of contour elements for calculating local contraction. |
| Calculation edge segment | There are two modes for rectangular calculation: Secant or straight line adaptation. |
| Smoothing wafer angle | +/- range, based on the maximum of the Gaussian distribution of the individual angles, which is used for the angle calculation. |
| Smoothing wafer size | +/- range, based on the maximum of the Gaussian distribution of the individual quantities, which is used for the size calculation. |
| Tolerance edge segment | +/- angular range, based on the actual angle of the wafer of the local continuous line, which are taken into account for the calculation of the wafer size. |

### 9.3.18.8 Miscellaneous tab

**NOTE:**
The tabs Binarization, Rectangle fit and Miscellaneous are only accessible in the expert mode. Activation via menu bar "Options / Expert Mode".

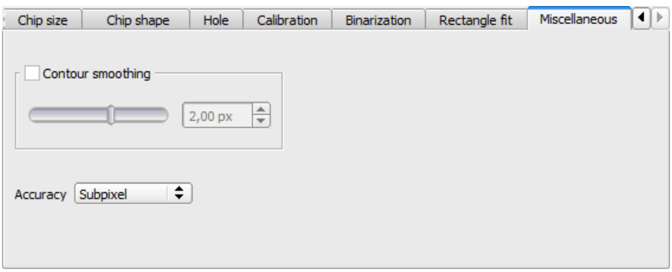


*Fig. 247: Detector Wafer, tab Miscellaneous*

**Parameter description:**

| Parameter | Function |
|---|---|
| Contour smoothing (positive / negative) | With the help of the downstream contour smoothing, errors can be retrieved later:<br>• increased (OPENING, parameter <0) or<br>• reduced (CLOSING, parameter> 0). |
| Accuracy | De(activate) subpixel method for evaluation of all wafer / cell data. |

### 9.3.18.9 Threshold value settings for differentiating between good and bad parts

**Excerpt from: VISOR® SolarUserManual1WIP 05-14 V.pdf**

Whether test parts are recognized as good or bad parts depends largely on the threshold settings. A typical setup will be illustrated by the following task: All good parts pass the test and all bad parts are recognized as bad parts and sorted out. In order to achieve this objective, some parts (pass and fail parts) need to be checked and the threshold limits must be adjusted until the results meet the relevant production requirement. Yes

In order to prevent borderline bad parts from passing through the test as good parts, the thresholds may need to be tightened, which increases the number of rejected parts but reduces the risk of downtime by passing borderline bad parts. If the thresholds are set too narrow, there is a risk that too many parts will be tested as bad parts and thrown out.

To ensure a high output, it may be necessary to set the thresholds less sharply. This carries the risk that a bad part will not be ejected, with all the negative consequences for further production.

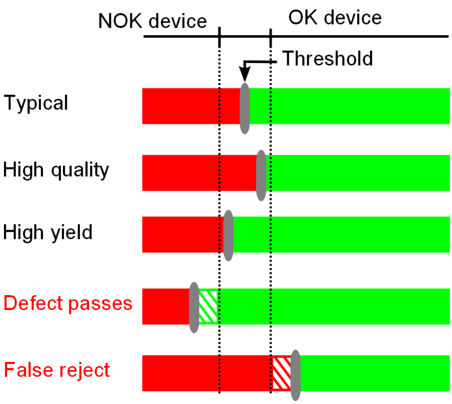


*Fig. 248: Settings of the switching threshold*

## 9.3.19 Detector Busbar

This detector is suitable for locating and testing busbars.

### 9.3.19.1 Busbar tab

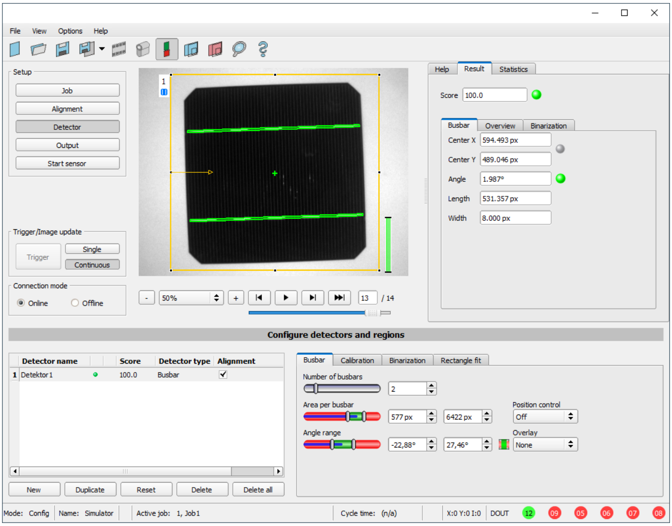



*Fig. 249: Detector Busbar, tab Busbar*

**Parameter description:**

| Parameter | Function |
|---|---|
| Number of busbars | Setting for expected number of busbars. |
| Area per busbar | Range to accept the area of a busbar. The surface criterion is an abort criterion, i.e. if the calculated area is larger or smaller, then any further calculation is aborted. The area of the busbars is calculated from the total number of selected pixels. |
| Angle range | Range for the accepted rotational position |
| Position control | To control the position of the wafer center of gravity, a rectangle or an ellipse can be freely positioned in the image area as the desired area. |
| Overlays | Activate graphical overlays for busbar pixels. |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

### 9.3.19.2 Binarization tab

**NOTE:**
The tabs Binarization and Rectangle fit are only accessible in the expert mode. Activation via menu bar "Options / Expert Mode".

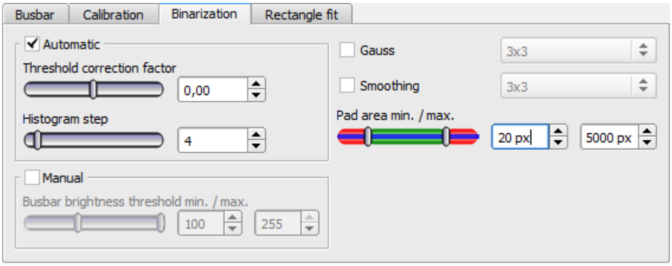



*Fig. 250: Detector Busbar, tab Binarization*

**Parameter description:**

| Parameter | Function |
|---|---|
| Automatic | Activate automatic binarization. |
| Correction factor for threshold | The threshold value for contour detection is calculated automatically from the brightness distribution of foreground and background. The position of the contrast threshold in the histogram can be shifted in the direction of the foreground or the background color. |
| Histogram increment | Gray scale value resolution of histogram history. |
| Manual | Activate manual binarization. |
| Busbar Helligkeitsschwelle min. max. | Min. max. Grauschwelle für Pixel die zum Busbar gehören. |
| Contour smoothing | With the help of the downstream contour smoothing, errors can subsequently be:<br>• increased (OPENING, parameter <0) or<br>• reduced (CLOSING, parameter> 0). |
| Gauss | Reduction of noise, annoying details, and rounding of the corners. |
| Pad area min. max. | Minimum and maximum area to detect a single pad. |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

### 9.3.19.3 Calibration tab

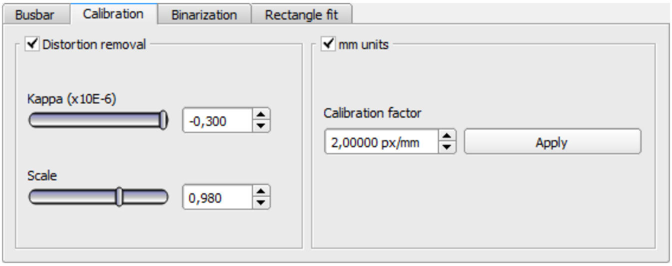



*Fig. 251: Detector Busbar, tab Calibration*

**Parameter description:**

| Parameter | Function |
|---|---|
| Distortion removal | Activate distortion removal. |
| Kappa (x10E-6) | Distortion coefficient for modeling the radial distortion |
| Scaling | Multiplicative correction. |
| mm units | Activate mm units. |
| Calibration factor | Pixel per mm; calibration factor to convert image data into world data. |
| Apply | By pressing "Apply", the dimensions in the other tabs are automatically adjusted to the new calibration factor. |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

### 9.3.19.4 Rectangle fit tab

**NOTE:**
The tabs Binarization and Rectangle fit are only accessible in the expert mode. Activation via menu bar "Options / Expert Mode".

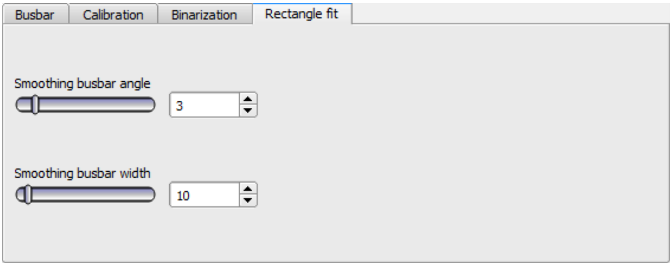



*Fig. 252: Detector Busbar, tab Rectangle fit*

**Parameter description:**

| Parameter | Function |
|---|---|
| Smoothing busbar angle | +/- range, based on the maximum of the Gaussian distribution of the individual angles, which is used for the angle calculation. |
| Smoothing busbar width | +/- range, based on the maximum of the Gaussian distribution of the single widths, which is used for the size calculation. |

For newly generated detectors, all parameters are preset as standard values, which are suitable for many applications.

## 9.4 Setup step Output

The Output setup step can be used to define the assignment and logic gates for the digital signal outputs, as well the interfaces and output data for the VISOR®.

### 9.4.1 Interfaces tab

**NOTE:**
The settings in this tab will affect the job set globally.

In this tab, you select and activate the used digital inputs / outputs and the interfaces for data output. In the "Active" column, the outputs and interfaces can be activated or deactivated separately.

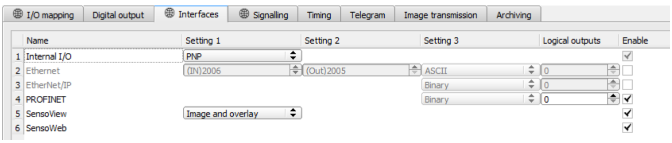



*Fig. 253: Output, Interfaces tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Internal I/O | Selection of internal I /O function: PNP or NPN |
| Ethernet | Ethernet TCP/IP for data output. The sensor is always a socket server. Two different ports are used, which can be defined by the user. Default setting: Port 2006 (IN) for commands to the sensor (control commands and response protocol) and port 2005 (OUT) for the actual data output.<br>Setting 3 can be used to select whether data should be output in binary (hex) or ASCII format.<br>For more information, please refer to the VISOR® Communications manual. |
| EtherNet/IP | Field bus EtherNet/IP for data output.<br>For more information, please refer to the VISOR® Communications manual. |
| PROFINET | Field bus PROFINET for data output and PLC communication.<br>**NOTE:**<br>The sensor starts the PROFINETPROFINET stack as soon as a job with is selected.<br>This reduces the evaluation speed slightly. Switching to another job without PROFINET does not stop the PROFINETstack. In order to stop the stack, the device must be turned off.<br>For more information, please refer to the VISOR® Communications manual. |

| Parameter | Function |
|---|---|
| SensoView | Activation or deactivation of the module "SensoView".<br>When the checkbox is deactivated, the button "View" in SensoFind can no longer be used to access SensoView.<br>If the checkbox "SensoView" is activated (default), the following settings can be selected for image transferal:<br>• **Overlay**<br>When "Overlay" is selected, only the overlay is transferred to SensoView. Image and pre-processing settings are not transferred.<br>• **Image and overlay**<br>For the setting "Image and Overlay", the image and the overlays are transferred to SensoView. Pre-processing settings are not transferred.<br>• **Image with pre-processing and overlay**<br>For this setting, both the image with the pre-processing settings and the overlay will be transferred to SensoView.<br>Additional information: Configure overlay (Page 172) and Pre-processing tab (Page 93)Additional information: |
| SensoWeb | Turns on the web server on the vision sensor. Similar to the locally installed module "SensoView", images and results can be visualized via a web browser via "SensoWeb".<br>The following browsers are supported: Microsoft Internet Explorer® from IE10, Google Chrome®, and Mozilla Firefox®.<br>To start SensoWeb, proceed as follows:<br>• Activate SensoWeb, under Output/Interfaces/SensoWeb<br>• "Start sensor" (button in SensoConfig)<br>• Open browser<br>• Enter the IP address of the sensor (visible in SensoFind) in the address bar of the browser, in the format: "http://your sensor IP", e.g. "http://192.168.100.100" (Default).<br>With http://192.168.100.100/zoom.html (or alternatively the IP address of the sensor), the enlarged view can be accessed directly.<br>Please refer to the following as well: VISOR® – SensoWeb (Page 366) |

For further information, see User Manual, Chapter "Communication"

**Logical outputs:**

Using the Ethernet and EtherNet/IP interfaces makes it possible to define additional logic outputs that only exist logically and can only be communicated via the data output interface.

Logical outputs can e.g. be associated with a detector result or a logical expression (formula).

## 9.4.2 Telegram tab

Configuration of data output for interfaces and archiving in .csv files. Result data that will be output via the previously activated interface can be configured here.

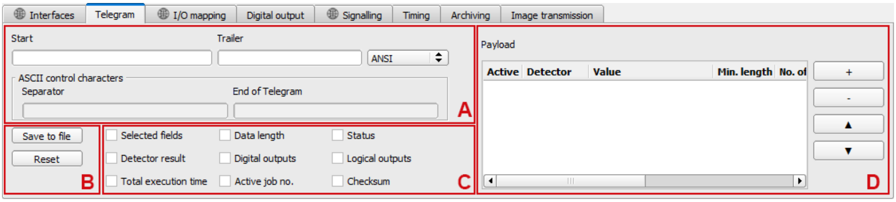



*Fig. 254: Output, Telegram tab*

**A**: Control characters

**B**: General

**C**: Checkboxes

**D**: Detector-specific payload

**A: Control characters**

**Protocol standard contents (start .... to .... check sum)**

| **Parameter description:** | **Length ASCII [Byte]** | **Length Binary [Byte]** | **Data type** | **Available for** |
|---|---|---|---|---|
| Start | | | | |
| Characters that are inserted at the beginning of the output data (prepended) | 0 ... 8 | 0 ... 8 | – | All types |
| Start | | | | |
| Characters that are added at the end of the output data (appended) | 0 ... 8 | 0 ... 8 | – | All types |
| Separator | | | | |
| Separates the values of detectors and selected fields in the output data | 0 ... 4 | na | – | All types |
| End of telegram | | | | |
| Characters that are added to the response to a request | 0 ... 4 | na | – | All types |

### B: General

| Parameter description: |
|---|
| Export |
| Exportation of file format with current results as .csv.<br>Detailed data format output for the custom output string as a .csv file with: Byte position (start position in string), data type, field name, detector name, value, length (in bytes), detector number, and detector type. |
| Reset |
| Resets all the input in this tab. |

### C: Checkboxes

Standard contents that are used frequently can be added to the output string by simply filling them out or enabling the checkboxes.

| Parameter description: | Length ASCII [Byte] | Length Binary [Byte] | Data type | Available for |
|---|---|---|---|---|
| Selected fields | | | | |
| This checkbox displays all selected fields. The checkbox for selected fields is not displayed. | 16 | 2 | **ASCII**: The order of output is from left to right and from top to bottom, i.e. one byte is set per active checkbox starting with the LSB.<br>**Binary**: The output sequence is from left to right and from top to bottom, i.e. one bit is set per active checkbox, starting with the least significant. | All types |
| Telegram length | | | | |
| Number of characters including the characters for the telegram length itself. | 1 ... 10 | 2 | **ASCII**: E.g. output string with 10 characters; telegram length 10 + 2 characters (one byte per decimal place) = 12 | All types |

| **Parameter description:** | **Length ASCII [Byte]** | **Length Binary [Byte]** | **Data type** | **Available for** |
|---|---|---|---|---|
| Status byte | | | | |
| Returns the Trigger mode. | 3 | 2 | **ASCII**:<br>PPF = Trigger; PFP = Free run<br>**Binary**:<br>0X06 0x00 = Trigger mode<br>0X05 0x00 = Free run mode | All types |
| Detector results | | | | |
| Output of overall result for each detector. | 4 ... 261 | 3 ... 35 | **ASCII**:<br>Byte 1 = AND conjunction of all detectors<br>Byte 2 = Overall Alignment result<br>Byte 3 = Overall current job result<br>Followed by the number of detectors; one byte per decimal place<br>Following one byte per detector<br>P = Detector pass<br>F = Detector fail | All types |
| Digital outputs | | | | |
| Returns the logic gate result for each digital output. | 2 ... 7 | N | **ASCII**:<br>Byte 1 Number of active outputs (logic gate result assigned)<br>Followed by bytes 2 – 7; one byte per output<br>P = Detector pass<br>F = Detector fail<br>0 = Inactive output (gap between two active outputs)<br>**Binary**:<br>Bytes 1 and 2: Number of active Outputs<br>Bytes 3 – n: Outputs, bit-coded | All types |

| Parameter description: | Length ASCII [Byte] | Length Binary [Byte] | Data type | Available for |
|---|---|---|---|---|
| log. Outputs | | | | |
| Returns the logic gate result for each logic output. | 1 ... 259 | N | **ASCII**:<br>From byte 1 ... n: Number of active outputs to which a logic result has been assigned.<br>Length: 1 byte per decimal place.<br>The following bytes n ... m: 1 byte per logical output<br>P = Detector pass<br>F = Detector fail<br>0 = Inactive output (gap between two active outputs)<br>**Binary**:<br>Byte 1 - 2: Number of active outputs to which a logic result was assigned.<br>Byte 3 ... n: all active logic outputs<br>1 = Detector Pass<br>0 = Detector Fail | All types |
| Execution time | | | | |
| Returns the execution time for the last evaluation. | 1 ... 3 | 4 | Signed integer | All types |
| Active job | | | | |
| Returns the job for the last evaluation. | 1 ... 3 | 1 | Unsigned int U8 | All types |
| Check sum | | | | |
| XOR check sum of all bytes in the telegram. Is transmitted as the last byte. | 1 | 1 | Unsigned int | All types |

**D: Detector-specific payload**

The check boxes can be used to flexibly add detector-specific payload, in any order you want, to the data telegram.

1. Use the "+" button to generate new entry.

**Function of the buttons:**

- "+": Insert new entry
- "-": Delete marked entry
- "Up", "Down": Displace marked entry

2. Select the detector you want in the "Detector" column.

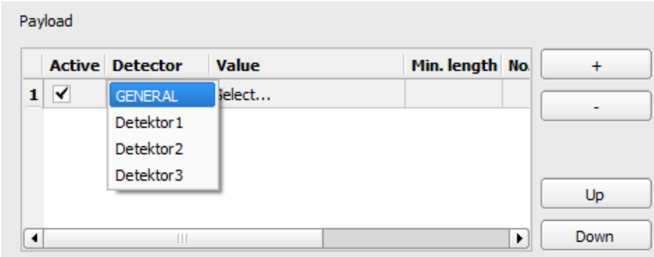



*Fig. 255: Output, Detector-specific payload*

3. Select the detector value you want in the "Value" column so that this value will be output through the enabled interface.
For more information on the available interfaces, please refer to: Interfaces tab (Page 311)

| Column | Function |
|---|---|
| Active | Enables / disabled the selected output value |
| detector | Detector name (select from drop-down list) |
| value | Available detector results (select from drop-down menu):<br>For an overview of detector-specific payload see Communications manual, Chapter Data output ASCII / BINARY |
| Min. Length | Used to define the minimum length for the "Value" cell. If the actual length is smaller than the set value, the field will be padded with spaces (ASCII) or zeros (binary). |
| No. of results | Only available for detectors BLOB as well as Contour and Pattern matching with Multiple objects.<br>Number of results of a detector that found several objects.<br>Example: Filtered by feature "Area" and found 10 BLOBs / objects.<br>Now up to 10 of these area values can be transferred as output data in a sequence.<br>All available output data see Communications manual, Chapter Data output ASCII / BINARY |
| Factor | Factor by which the result data is multiplied (determines the number of decimal places) |

<table>
<tr><th>Column</th><th>Function</th></tr>
<tr><td>Bit depth (binary)</td><td>Specifies the length in bits and thus the value range of the transmitted value.<br>If the value is outside the value range, it is displayed with the highest / lowest possible value.<br>Usable value ranges:
<table>
<tr><th>Bit</th><th>Signed</th><th>Unsigned</th></tr>
<tr><td>8</td><td>-127 to +126</td><td>0 to 254</td></tr>
<tr><td>16</td><td>-32.767 to 32.766</td><td>0 to 65.535</td></tr>
<tr><td>32</td><td>-2.147.483.647 to 2.147.483.646</td><td>0 to 4.294.967.295</td></tr>
</table></td></tr>
<tr><td>Sign (binary)</td><td>Specifies whether the transmitted value is signed or unsigned (Signed / Unsigned)</td></tr>
</table>

For an overview of detector-specific payload see Communications manual, Chapter Data output ASCII / BINARY

### 9.4.3 I/O mapping tab

**NOTE:**
The settings in this tab will affect the job set globally.

Here, the following settings can be made:

- Determination of which of the variably usable I/Os are to be used as input or output.
  Pin 05 pink, pin 06 yellow, pin 07 black (LED B), and pin 08 gray (LED C) can be used as input or output.
- Assignment of functions to the inputs / outputs.
  The respective list boxes show the functions available for this input or output and can also be defined here. The functions listed under "Sole functions" are **only** available via this pin / line.

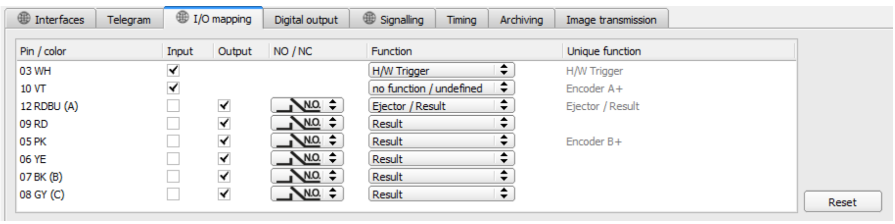



*Fig. 256: Output, I/O mapping tab*

### Functions of inputs

| Function | Description |
| --- | --- |
| H/W trigger | Hardware Trigger (only available via Pin 03 white) |
| Encoder A+ | Input for encoder, Track A+ (only available via Pin 10 purple) |
| Encoder B+ | Input for encoder, Track B+ (only available via Pin 05 pink) |
| Enable trigger | Function for enabling or suppressing trigger signals. Reading this function takes about 1 ms. This creates a pause in which a trigger signal is ignored, even though the Enable Trigger signal is present. |
| Job 1 or 2 | Job change between Job 1 and Job 2, depending on level at this input.<br>Low = Job 1, High = Job 2. |
| Teach temporarily / Teach permanently | Teaching of all detectors. As soon as a high signal is applied **and** a trigger occurs, teach-in starts.<br>Temporary: Storage in RAM, i.e. fleeting after reset,<br>Permanent: Storage in Flash, i.e. permanent, even after reset |
| Job switch (BitX), binary coded | Job switching via binary bit pattern to up to 8 inputs, which can be defined for this purpose, i.e. switching between 1 to 255 jobs. Ranking of the bits according to assigned ascending input designation 1 - 8. Bit 1 = LSB.<br>Please refer to section: Job 1 … 255 via a binary input bit patternSee Chapter: |
| Repeat Mode Enable | Images are captured and evaluated as long as:<br>There is a high level at this input and none of the following termination criteria is met:<br>- "Overall job result" = positive (adjustable under Output / Output Signals)<br>- "Max. cycle time" is not elapsed (if active).<br>If the input Repeat Mode Enable is used, it also acts as Trigger Enable. I.e., triggers will be accepted and processed only if there is a high level at this input; see below: Input: Repeat Mode Enable, with Trigger (Page 325) |
| Multishot trigger (only if Multishot is active) | Default setting if Multishot is active, instead of H/W Trigger |
| No function, undefined | No function, not used |

## Connection Encoder

Functions that are already fully exploited appear pale gray in the listbox because they are no longer available. All input signals must have a minimum signal length of 2 ms.

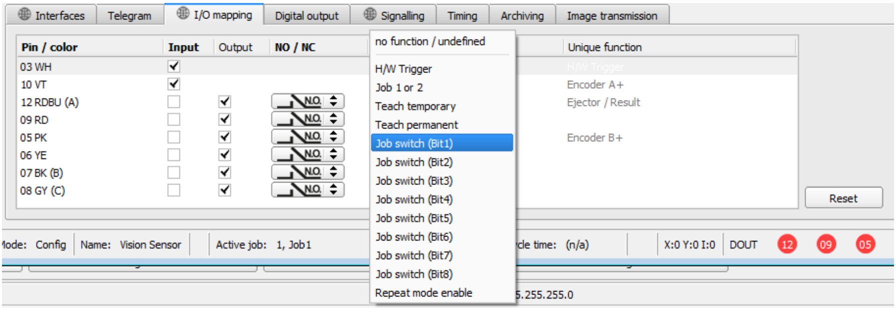



*Fig. 257: Output, I/O mapping tab, Inputs*

If both tracks A+ and B+ are used, forward/backward differentiation or counting is possible. The encoder inputs can process a maximum frequency of 40 kHz.

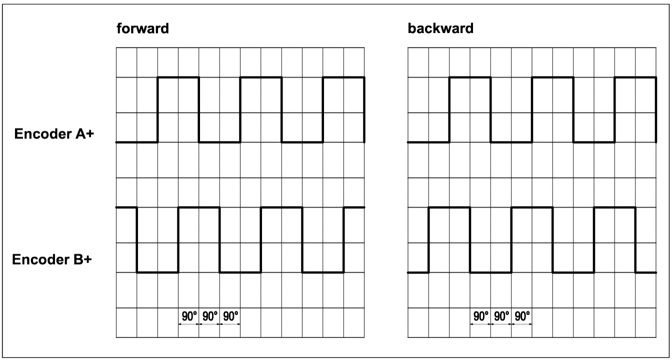



*Fig. 258: Encoder tracks A+ / B+*

## Functions of outputs

The output's default state is defined with NO / NC:

- NO (normally open) = Open if logical expression = False
- NC (normally closed) = Closed if logical expression = False

| Function | Description |
| --- | --- |
| Ejector | Special ejector output can be loaded with up to 100 mA (all other outputs = 50 mA), only available via Pin 12 RDBU (corresponds to indicator LED "A"). |
| result | Result output; each of the result outputs defined here can be assigned a detector result or a combination of detector results in the tab "I /O Logic". |
| Confirmation job change | When changing jobs with the digital I/O ("Job pin X, binary-coded"), a falling / rising edge can be configured here in order to confirm a successful change. The high edge is set after the new job content is loaded and active, i.e. at the same time as the high edge of the ready signal after switching (see Timing ...). The high level stops for 20 ms and is then deleted again.<br>If the switchover was unsuccessful, no high level is output and the signal is permanently low. |
| External illumination | If this setting is selected (only available via Pin 09 RD), external lighting can be connected / triggered here. |
| No function, undefined | No function, not used |

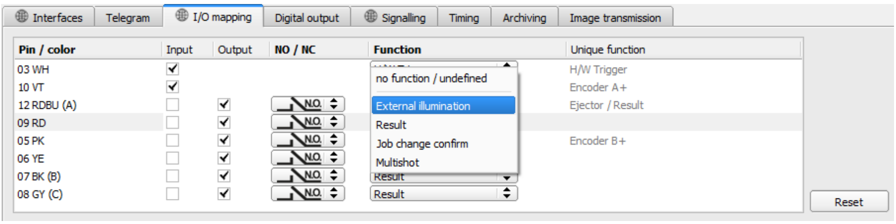



*Fig. 259: Output, I/O mapping tab, Output*

**There are two permanently defined outputs:**

- Ready: Indicates whether the sensor is ready to receive a trigger / next evaluation.
- Valid: Indicates whether the data is valid at the outputs.

**N.O. / N.C.**

For each output, you can define which switching function should be used: Normally open (N.O.) or normally closed (N.C.).

**Programmable functions of the digital inputs:**

In operation with a process control, the following functions can be performed via the inputs:

- Inactive
- Enable / Disable
- Load job (binary coded)
- Teach temporarily
- Teach permanently

**Description of different cases with signal diagram.**

All signals shown here are based on the setting "PNP"

### Input: "Enable trigger"

Activates the trigger input of the sensor (high signal) or blocks the hardware trigger (low signal).

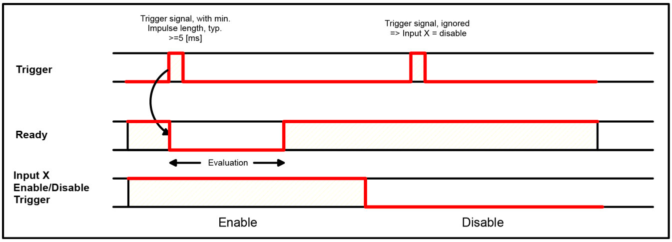



*Fig. 260: Input timing, Enable trigger*

### Input: Job change via binary signals, or via function Job 1 or 2

### Binary job change via up to 5 inputs (Job 1 - max. 255):

When changing the binary input signals, Ready is set to Low. Ready remains Low until the switchover to the new job occurs. If the optional job change confirmation signal is used, it will occur after the job change, and Ready will only become High again after that. During job switching, no trigger signals may be sent. The level change of the associated inputs must be made simultaneously (within a maximum of 10 ms, all levels must be stable). If the level changes of individual inputs are further apart, several job switches are executed one after the other if necessary).

### Job change through function Job 1 or 2:

When changing the level of the correspondingly defined input, Ready is set to Low. Ready remains Low until the switchover to the new job occurs. If the optional job change confirmation signal is used, it will occur after the job change, and Ready will only become High again after that.

During job switching, no trigger signals may be sent. With Job 1 or 2, low level switches to Job 1 and high level to Job 2.

### Difference between binary signals and Job 1 or 2:

When using the switchover via binary signals, the desired job number must always be coded in binary code. At least 2 inputs must be used for 2 jobs.

With Job 1 or 2, low level switches to Job 1 and high level to Job 2. In this way, two jobs can be selected via one input.

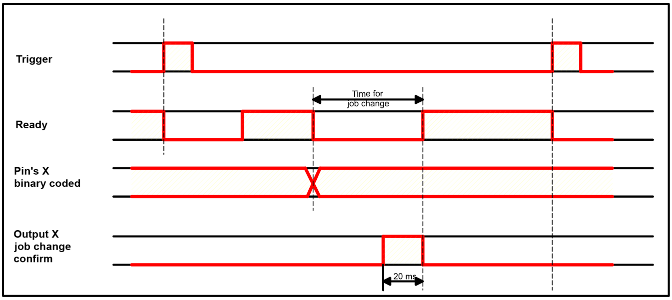



*Fig. 261: Input timing, Job change via Binary / Job 1 or 2*

### Input: Teach temp. / perm.

For re-teaching samples of all detectors of the current job. A rising edge initiates teaching, whereby the high level must be present at least until the next trigger so that an image of a test piece can be recorded in the correct position. Ready is set to Low and remains low until teaching is complete. Depending on the setting, storage is either temporary (in RAM only) or permanent (in Flash).

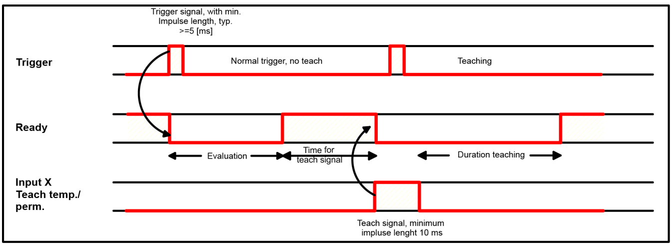



*Fig. 262: Input timing, Teach*

**NOTE:**
The Job 1 or Job 2 and temp./perm. teaching functions can only be used in trigger mode.

### Input: Repeat Mode Enable, with Trigger

Images are acquired and evaluated as long as a high level is present at this input and none of the following criteria is fulfilled:

- "Overall job result" = positive (adjustable under Output / Output Signals)
- "Max. cycle time" is not elapsed (if active).

If the Repeat Mode Enable input is used, it also acts as a Trigger Enable. This means only if a high level is applied to this input, Trigger will be accepted and processed

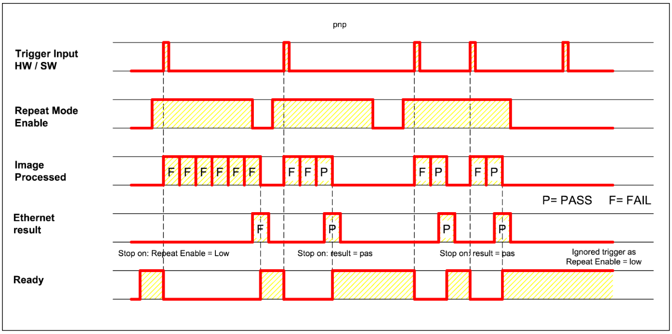



*Fig. 263: Input: Repeat Mode Enable, with Trigger*

### Input, Repeat Mode Enable, in Freerun

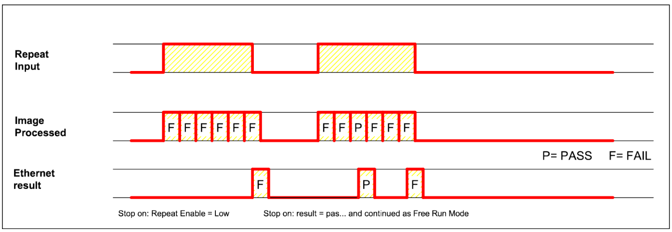



*Fig. 264: Input, Repeat Mode Enable, in Freerun*

## 9.4.4 Digital output tab (Digital outputs / logic)

In this tab you can define the switching behavior and the logical connection of the individual detectors with the digital outputs. The number of outputs depends on the settings under the tab I/O mapping. Additionally, an I/O extension can be controlled via the serial interface.

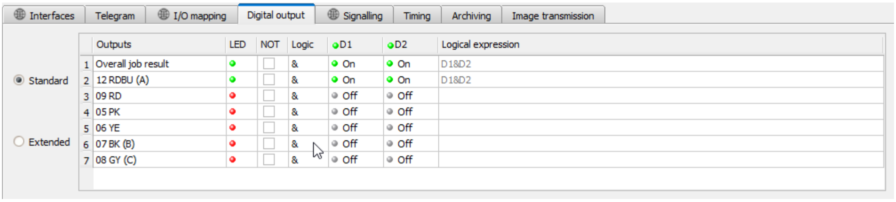


*Fig. 265: Output, Digital output tab, Logic*

**Select logical combination of detectors for each output:**

For each pin (output), there are the following possibilities:

| Parameter | Function |
|---|---|
| Overall job result | No physical output. Affects logic for recorder, statistic, and archiving functions |
| Invert | Invert total result from the following settings for this pin (output) |
| Mode | Standard: Several detectors can be combined into a logical expression using the logical operators AND (&) / OR (\|) / NOT (!).<br>Advanced: The logical formula for combining the detectors can be freely created. |
| NOT | Select: Operator NOT (!) |
| Logic | Select: Operator AND (&) / OR (\|) |
| D1 - D ... | Depending on the number of activated detectors, all detectors are inserted in this list. These can be logically assigned to each listed output. Each detector can be switched on, inverted, or off for the respective pin (output). |
| Logical Expression | Either the logical expression compiled in standard mode is displayed, or the logical expression can be compiled here in advanced mode. |

**Defining logical connection:**

Define the logical link between the test results of the individual detectors and the status of the selected output. You have two input possibilities:

- Standard mode (checkboxes and operators)
- Extended mode (formulas)

**NOTE:**
If an external multishot illumination is connected, the pins: 09, 06, 07, and 08 can no longer be assigned with output signals.

### 9.4.4.1 Logical connection – Standard mode

In standard mode, connection of the detector test results for the selected output is made via the radio buttons Operator and the Checkboxes in the detector drop-down list. The result is displayed in the field "Logical formula" (not editable).

**Connecting results:**

1. In the Operator field, select the logical operator for linking the detectors in the drop-down list.

2. In the drop-down list, activate the detectors that should contribute to the result (tick in the Active column).

By activating the column "Inverted", you can invert the respective detector result.

The entry in the Result column changes accordingly.

**Examples:**

Here, the detector results can only be linked by a logical operation such as:

- (D1&D2&D3) or
- !((!D1)|D2|D3), etc.

**NOTE:**
If a detector is assigned to an image acquisition (see "Multiple Image Acquisition" Chapter Cycle time tab (Page 141)), its result in the other images does not affect the result of the combination.

### 9.4.4.2 Logical connection - extended mode

In extended mode, the combination of detector test results for the selected output is defined by direct input of a logical formula. For this you have the operators AND, OR, and NOT as well as parentheses.

To edit the formula, please use the following characters for the logical operators:

- "&" for AND
- "|" for OR
- "!" "!" for NOT

**Examples:**

Here, logical expressions of any complexity can be created, e.g.:

- (D1&D2)|(D3&D4)
- !((D1|D2)&(D3|D4))
- (D1|D2)&(D3|D4)&(D5|D6)

etc.

**NOTE:**
If a detector is assigned to an image acquisition (see "Multiple Image Acquisition" Chapter Cycle time tab (Page 141)), its result is set to logical "0" in the remaining image acquisitions. The result of the combination must be adjusted accordingly.

## 9.4.5 Signalling tab

**NOTE:**
The settings in this tab will affect the job set globally.

The Signalling tab can be used to configure the settings for statistics and for the digital outputs.

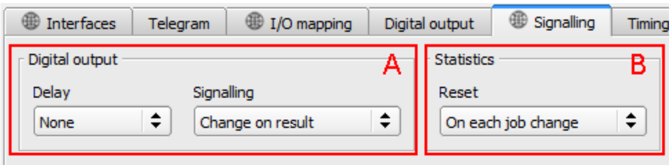



*Fig. 266: Output, Signalling tab*

### A: Digital outputs

| Parameter | Function |
|---|---|
| Delay | You can choose to either delay all outputs or only the ejector output. |

| Parameter | Function |
|---|---|
| Signaling | Result outputs can be reset based on various settings / events:<br>• Change on next result (default):<br>The output changes its level according to the logical result only when the next result is obtained. Typically used in switch control for e.g. sorting, etc.<br>• Change on trigger:<br>The output will be set to "inactive" (low when using PNP mode) when the next trigger occurs. Typically used for operation on a PLC.<br>• Duration of result:<br>The output will switch back to an inactive state after the result duration (in ms) configured here elapses. Typically used for e.g. pneumatic ejectors (blower) |

### B: Statistics

| Parameter | Function |
|---|---|
| Reset | Used to select whether the statistics will be reset with ever job change or only with "Start sensor." |

**ATTENTION:**
For job change and change from run to config mode, the following special states apply: The buffer of the delayed outputs is deleted.

### Digital outputs:

These are reset to the default settings (defaults) when changing jobs and changing the operating mode from "Run" to "Config". The basic settings are defined by "Invert" in the SensoConfig/Output/Digital output tab. The selection "Inverse" inverts the basic setting of the digital output and the result at the same time.

### Ready and Valid

- Ready signals when High, readiness for new image acquisition.
- Valid signals when High that results at the outputs are valid.

### PNP or NPN operating mode

All examples described here are executed in operating mode "PNP". If the setting "NPN" is set, the examples apply analogously with reversed levels.

### 9.4.6 Timing tab

This tab can be used to configure the time response characteristics for the selected output signal. All timing parameters are specified either in milliseconds or (if enabled) encoder pulses. Encoders can be enabled in the I/O mapping tab.

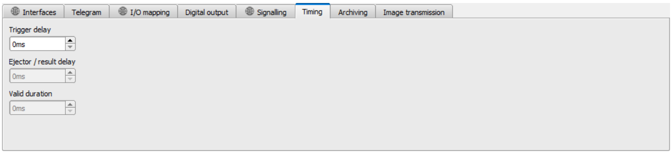



*Fig. 267: Output, Timing tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Trigger delay | Time between the trigger and the start of the image acquisition operation in ms or encoder pulses.<br>The maximum possible value is 3000 ms / encoder pulses.<br><br>How the delay setting will work if you use:<br>• H/W Trigger (digital input): This delay is effective.<br>• Trigger (via Ethernet, PROFINET): The delay will not be applied. The image acquisition operation will be carried out directly after the trigger. |
| Ejector / result delay | Time between the trigger and the presence of the event level, in ms or encoder pulses. The maximum number of components between the trigger and the ejector is 20 (buffer size).<br>The maximum possible value is 3000 ms / encoder pulses.<br><br>How the delay setting will work if you use:<br>• H/W Trigger (digital input): This delay is effective and starts with the H/W trigger.<br>• Trigger (via Ethernet, PROFINET): This delay is effective but only starts after the image is processed (not with the trigger!).<br>• Please select Delay in the Signalling tab. |
| Duration of result | Duration of result signal in ms or encoder pulses.<br>Maximum value of 3000 ms / encoder pulses.<br>Please select Result duration in the Signalling tab. |

> **ATTENTION:**
> When there are job changes, and when there is a change from Run mode to Config mode, the delayed output buffer will be cleared.

**Ready and Valid**

- Ready signals when High, readiness for new image acquisition.
- Valid signals when High that results at the outputs are valid.

**The following cases in the time behavior can be distinguished:**

All examples described here are executed in operating mode "PNP". If the setting "NPN" is set, the examples apply analogously with reversed levels.

**Normal trigger without the use of delay times:**

**Flow:** (in this case: Change on next )

- Rising edge at Trigger input (Pin03 WH)
- Consequence of Trigger = High: Ready = Low, and Valid = Low
- After the vision sensor has evaluated the image and the corresponding results are available, all defined outputs switch to the corresponding logical states and Ready and Valid go back to High level (outputs valid, vision sensor ready for next evaluation)

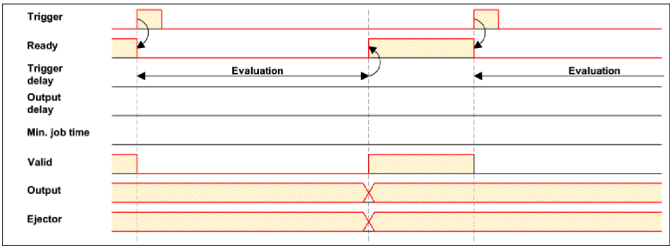



*Fig. 268: Digital outputs timing, standard sequence with normal trigger*

**Trigger delay active**

The trigger delay will only be applied to hardware trigger

This setting is used to selectively delay image acquisition / start of the evaluation compared to the actual physical trigger, which is e.g. triggered by a trigger photocell or the machine control. This allows the fine adjustment of the trigger time without any changes to the mechanics or the control program.

**Procedure:**

Image is acquired after the trigger delay time is elapsed. The cycle time is: Trigger delay time + evaluation time.

See SensoConfig/Output/Timing/Trigger/Delay

- Rising edge at Trigger input (Pin03 WH)
- As a result of Trigger = High: Ready = Low, Valid = Low, all defined result outputs = Low (signaling = change at trigger)
- Before the image is taken for evaluation, the set trigger delay time (trigger delay) elapses.
- Now the evaluation starts. As soon as the corresponding results are available, all defined outputs change to the corresponding logical states. Ready and Valid will switch back to a high state (outputs valid, VISOR® ready for next evaluation).

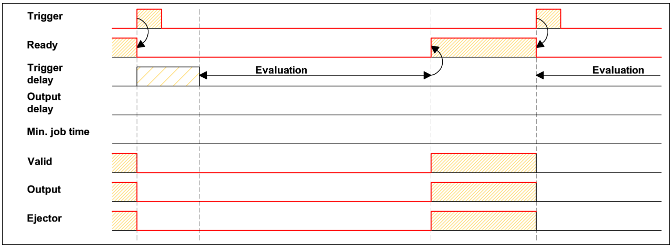



*Fig. 269: Digital outputs timing, Trigger delay*

**Trigger delay + Result delay acting on ejector:**

(ejector only in this case)

The trigger delay will only be applied to hardware trigger

The result delay (whether for all outputs or only ejectors) is used to fine tune e.g. the ejector time regardless of the evaluation time, especially since this may also have slight fluctuations.

**Procedure:**

Image is acquired after the trigger delay time is elapsed. In addition, the result delay works. In this example, however: only on the ejector output (Pin 12 RDBU).

For the defined result outputs except the ejector output, the cycle time is: trigger delay + evaluation time

The cycle time for the ejector output is: The result delay itself (counted from trigger time, only meaningful if longer than sum times!). See SensoConfig/Output/Timing/Digital Output/Delay.

- Rising edge at Trigger input (Pin03 WH)
- As a result of Trigger = High: Ready = Low, Valid = Low, all defined result outputs = Low. Except ejector, a fixed duration of results is defined for this,
- Before the image is taken for evaluation, the set trigger delay time (trigger delay) elapses.
- Now the evaluation starts. As soon as the corresponding results are available, all defined outputs (except ejector) change to the corresponding logical states. Ready and Valid return to a high level.
- In this operation mode, only the Ejector output is set after the Result delay has elapsed. The ejector output is also provided with a result duration in this example and is therefore set to Inactive after this result duration.

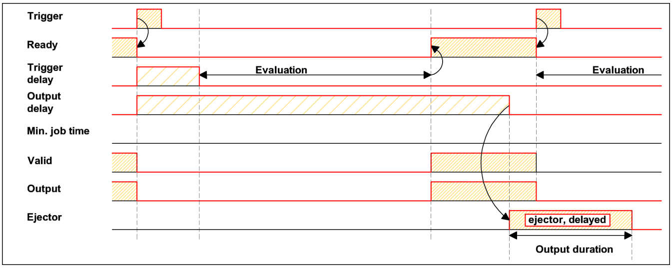



*Fig. 270: Digital outputs timing, Result delay ejector*

### Trigger delay + Result delay acting on all outputs:

The trigger delay will only be applied to hardware trigger

The result delay (whether for all outputs or only ejectors) is used to fine tune e.g. the ejector time regardless of the evaluation time, especially since this may also have slight fluctuations.

### Procedure:

Image is acquired after the trigger delay time is elapsed. Furthermore, in this example, the result delay affects ALL defined result outputs.

The cycle time is the same for all result outputs: The result delay itself (counted from trigger time, only meaningful if longer than sum of trigger delay + evaluation time!). See SensoConfig/Output/Timing/Digital Output/Delay.

- Rising edge at Trigger input (Pin03 WH)
- Consequence of Trigger = High: Ready = Low, Valid = Low.
- Before the image is taken for evaluation, the set trigger delay time (trigger delay) elapses.

- Now the evaluation starts. After the results are available, only the Ready signal is set to High again (ready for the next evaluation). Otherwise, the delay in the result delay is still awaited. Only then do all defined outputs change to the corresponding logical states. Valid goes back to high level (Valid = High: result outputs valid, Signaling = Change in next result).

In this operating mode, only the "Ready" signal changes after elapse of the trigger delay + image acquisition + evaluation time. Ready = High: Ready for the next evaluation. The reason this makes sense is that the VISOR® is already ready for the next evaluation regardless of whether the outputs are set later on.

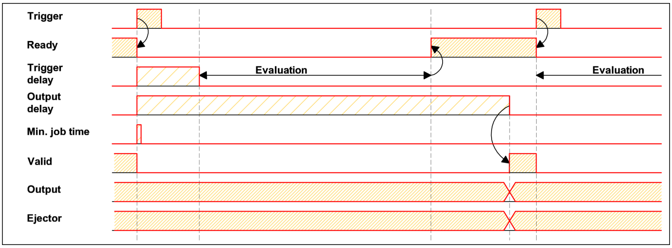



*Fig. 271: Digital outputs timing, Result delay all outputs*

### Result duration, e.g., acting on all outputs:

This time setting is used to achieve an output pulse with a defined length, such as for controlling a pneumatic ejector (blower) in the case of a bad part, or the like.

All defined result outputs are reset to low level (inactive in PNP mode) after the result duration in ms is elapsed.

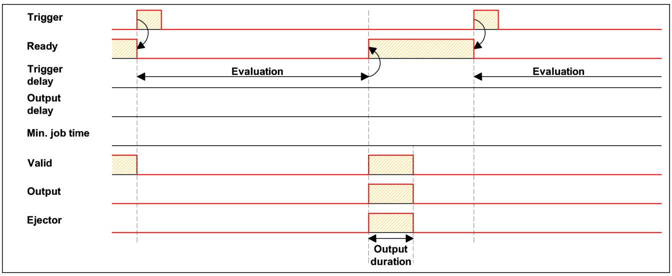



*Fig. 272: Digital output timing, Result duration*

**Cycle time (min., max.) active:**

(Here: Signaling: Change at Trigger)

Parameters for controlling the execution time of a job.

The minimum execution time can be used to suppress multiple triggers and can affect the LED performance. (i.e. if another trigger is received within the minimum job time, it will be ignored)

The maximum execution time is used to cancel a job after a defined time. The result of the job after termination is always "not OK". The maximum execution time should always be greater than the time required for an evaluation.

The cycle time measures the time from the trigger to the setting of the digital switching outputs. If the cycle time should be limited, e.g. because the machine cycle must not be exceeded, the value for the maximum cycle time must be limited accordingly. The result of all unfinished detectors up to this time is set to faulty. When choosing the maximum cycle time, it must be taken into account that this is not adhered to strictly. However, depending on the detector that has just been executed, it can take several milliseconds before it can break off. It is recommended that this maximum cycle time over the actual execution time is checked and the set maximum cycle time is reduced accordingly.

**Procedure:**

All outputs and the signal "Valid" (Outputs valid) are set directly after evaluation.

However, the signal "Ready" (Ready for next evaluation) is set only after the min. Job time has elapsed, and thus triggers are only accepted for the next evaluation from this moment.

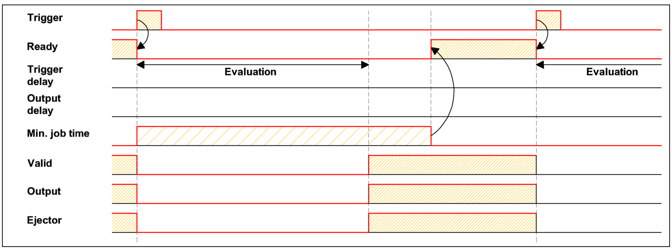



*Fig. 273: Digitale Ausgänge Timing, Min. Job Zeit*

**Multiple result delay for ejector**

This operating mode is used if, between the trigger/evaluation for test part A and its ejection, there is so much time / such a large distance that the VISOR® already has to check n (max. of 20) additional test parts and manage their accordingly belated ejection time.

(only available in mode: SensoConfig/Output/Timing/delay: "ejector / ejector / result delay only")

Here: Signaling = Result duration (alternatively also "Change with next result" applicable)

A maximum of 20 components may fit between the trigger and the ejector.

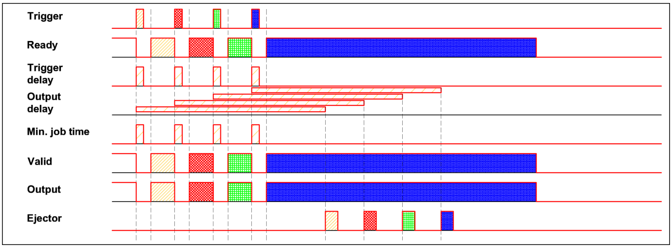



*Fig. 274: Digital output timing, Multiple result delay ejector*

## 9.4.7 Archiving tab

In the Archiving tab, you can configure the archiving of the data.

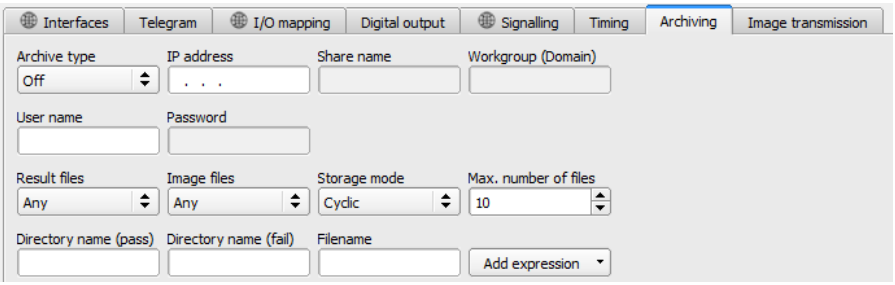



*Fig. 275: Output, Archiving tab*

**Parameter description:**

| Parameter | Function and setting options |
|---|---|
| archive type | Off: No archiving<br>FTP: Archiving to the FTP server<br>SFTP: Archiving on an SFTP server with the SSH FTP protocol<br>SMB: Archiving on a drive with the SMB service (Server Message Block, up to Version 3.1.1)<br>**NOTE:**<br>When using archive servers in other subnets, first set the gateway in SensoFind. |

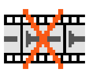
| Parameter | Function and setting options |
| --- | --- |
| IP address | IP address of target server / client |
| Sharing name | Sharing name which was defined at the folder release in the PC in the dialog: "extended release". |
| Workgroup (domain name) | Optional!, Workgroup / Domain name of target server / client. |
| User name | Username for FTP / SFTP / SMB connection. |
| password | Password for FTP / SFTP / SMB connection. |
| Result files | If the protocol file is activated, all data specified under "Output / Data Output" will additionally be logged in a .csv file. A file is created for each evaluation (trigger). The files are numbered consecutively. |
| Image files | Activates archiving of images.<br>**NOTE:**<br>• Images are stored without preprocessing settings, but with the settings for the arrangement (e.g. rotated or mirrored)<br>• FTP, SFTP, and SMB always save images without overlays only. To store images with overlays, please use SensoView. Additional information: VISOR® Software – SensoView |
| Storage mode | Limit: After reaching maximum number of files, transferal is stopped.<br>Unlimited: Files are written until the target drive is full.<br>Cyclic: After reaching maximum number of files, the older files are replaced by the newer ones. |
| Max. number of files | Maximum number of data records that can be stored in the target directory. |
| Directory (pass) | Directory for archiving the data sets of good parts (for C:/TESTGOOD, only enter TESTGOOD). |
| Directory name (fail) | Directory for archiving the data sets of bad parts (for C:/TESTBAD, only enter TESTBAD) |
| File name | File name for images and protocol file; this name is extended automatically by the image number (e.g. TESTFILE). |
| Add expression | A dynamic portion (information such as date and time) is added to the file name.<br>See also table below |

The following table shows the expressions that can be added to the file name.

| Expression | Description | Example |
|---|---|---|
| TIME | HHhMMmSSsSSSms | 09h05m11s034ms |
| HOUR | hh | 09 |
| MIN | mm | 05 |
| SEC | ss | 11 |
| MSEC | sss | 034 |
| DATE | YYYY-MM-DD | 2011-09-21 |
| YEAR | YYYY | 2011 |
| 2YEAR | YY | 11 (for 2011) |
| MONTH | MM | 09 |
| DAY | DD | 21 |
| STRINGID | "Data" entry from extended trigger request "TRX" | Part 34 |
| COUNTER | Counter from statistic | 3824 |
| XXCOUNTER | Counter taken from statistics with a defined number of digits.<br>XX indicates the number of displayed digits and can accept values from 01 to 09<br>**NOTE:**<br>If the number of digits of the counter is too small, it is prefilled with "0".<br>If the number of digits of the counter is too large, digits are discarded. | 06COUNTER → 003824 |
| result | Overall result of job | Pass or Fail |
| SENSOR NAME | As specified in SensoFind | |
| JOB NAME | As specified in SensoConfig | |

## 9.4.8 Image transmission tab

In the Image transmission tab, the image transferal and / or the image recorder and the Ram disk can be activated.



**ATTENTION:**
When this icon appears on the life image, it is indicating the image visualization / image storage on the PC is running slower than the image processing on the VISOR®. Not all images taken by the VISOR® are displayed anymore. This may lead to image loss when using poor image archiving.
If the icon appears frequently, programs opened on the PC should be closed in the background to provide more PC performance.


*Fig. 276: Output, Image transmission tab*

**Parameter description:**

| Parameter | Function |
|---|---|
| Image recorder | Storage of max. 10 images in the sensor's internal ring buffer. Setting options: Off, Any, Pass, Fail |
| RAM disk | Storage of the last image in the internal RAM memory; this image can be fetched from an sFTP client.<br>Setting options: Off, Any, Pass, Fail.<br>The image is stored in the RAM of the VISOR®, under the name "image.bmp", in the /tmp/results/ directory.<br>Parameters for SFTP client: User: "user", Password: "user"<br>When switched on, the result data (all defined in "Output / Data Output", with separator ";") can be obtained in the same way via the "result.csv" file. |

**Different types of image archiving**

| Access | Description | Max. number of images | Image filter | Drawings |
|---|---|---|---|---|
| Image recorder in VISOR® (RAM) | In Run mode, images are stored on the VISOR®. The images can be transferred to a PC by SensoConfig or SensoView. | 10 | as predefined in the settings "Filter". | No |
| SensoView Archiving / SensoConfig saves image | Images transferred tSensoViewo can be stored on hard disk of PC. | unlimited (Limit is size of hard disk in PC) | as predefined in the settings "Filter". | Yes / No can be selected |
| Saving of filmstrips in SensoConfig | Current images from filmstrip can be saved as filmstrip (*.flm) or as bitmap (*.bmp) on hard disk of PC. | 50 | without filtering | No |

| Access | Description | Max. number of images | Image filter | Drawings |
|---|---|---|---|---|
| Last image in VISOR® | The last image will be stored on the RAM drive of the VISOR® and can be fetched from "directory /tamp/results" via FTP. | 1 | without filtering | No |
| Images archived via FTP, SFTP, or SMB | Images archived via FTP, SFTP, or SMB | unlimited (Limit is size of hard disk in PC) | selectable with / without filtering | No |
| Get Image Request | The last VISOR® image can be transmitted to a program on the PLC or PC with the "GetImage" command | unlimited (Limit is size of hard disk in PC) | as pre-defined in the settings "Filter". | No |

## 9.5 Setup Start sensor

This function sets the sensor to Run mode and executes the job.

### Starting execution of a job:

Click on the "Start Sensor" button.

The active (= marked in the drop-down list) job is transferred to the sensor, stored in the sensor's non-volatile memory, and started (Run mode). Dieser Stand wird zudem automatisch als Jobsatz-Backup abgespeichert (siehe Automatisches Job-Backup).

In the image window, the following are displayed: the found features, the test results for the first or selected detector in the drop-down list, and statistical parameters.

### Changing detector display:

To display the inspection results for another detector, mark it in the detector drop-down list (bottom left) or click on its graphical representation in the image window.

### Terminating job execution:

Click on the "Stop Sensor" button. You are now back in configuration mode and can edit your job.

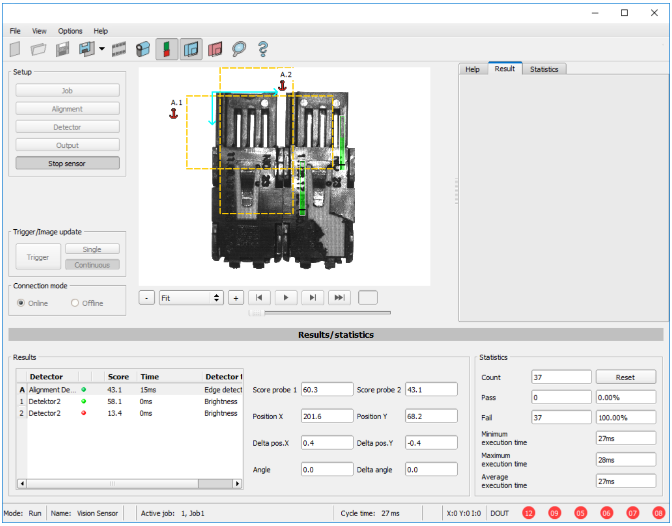



*Fig. 277: Start sensor*

## 9.6 Trigger / Image update

Select the Trigger mode you want in the job settings in the "Image acquisition" tab:

| Parameter | Function |
|---|---|
| Trigger | Operation with external trigger, or "Trigger" button on the SensoConfig surface |
| Free run | Operation with automatically running self-trigger; the sensor supplies images with the maximum possible frequency |

Select the format in which the sensor should deliver images by using the buttons in the Trigger / Image update update section:

| Parameter | Function |
|---|---|
| Single image | Capturing a single image; image capture occurs once when:<br>1. Trigger mode = Trigger: First external trigger signal or with the "Trigger" button on the SensoConfig interface<br>2. Trigger mode = Free run: First click on the "Single image" button on the SensoConfig interface (important e.g. in setup mode) |

| Parameter | Function |
|---|---|
| Continuous | Continuous supply of images; image capture occurs continuously when:<br>1. Trigger mode = Trigger: Each external trigger or with each click on the "Trigger" button on the SensoConfig interface<br>2. Trigger mode = Free run: Continuous through internal self-triggering with maximum frequency |

Changing the parameters exposure time, gain, lighting, or resolution in the job settings will automatically request a new image from the sensor.

To obtain a continuously updated live image without trigger, carry out the following settings:

- Set to free run under Job / Image acquisition
- Set to "continuous" under "Trigger / Image update"

## 9.7 Connection mode

There are two operating modes available for configuring and testing the sensor, which you can select in the "Connection mode" field.

- Online mode: Configuration with connected sensor.
- Offline mode: Simulation of a sensor with the help of images stored in filmstrips.

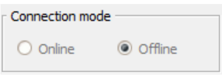



*Fig. 278: Connection mode*

When the sensor is connected, both modes are available; it is possible to switch between the two. If no sensor is available, it is only possible to work in Offline mode, i.e. with sensor simulation.

## 9.8 Displays in the image window

### 9.8.1 Image section and zoom

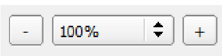



*Fig. 279: Zoom*

Use the buttons or the drop-down menu below the image window to select the desired image section.

## 9.8.2 Graphical display of results

In the View menu, you can activate or deactivate the following graphical representations:

- Bar graph result: Displays the inspection result as a bar graph
- Overlays: Displays search ranges, feature ranges, and position frames of detectors and Alignment
- Focusing aid: Displays image sharpness (see also Job settings)
- Enlarged display: Displays a separate, enlarged image window that can be scaled to any size via handles on the frame corners

The program SensoView offers a limited selection of these functions.

## 9.8.3 Controlling the image display

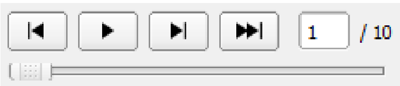



*Fig. 280: Image reproduction*

Use the buttons and the scroll bar below the image area to control the selection and playback of saved images. The frame counter displays the number of the current image and the number of images in the active filmstrip.

| Buttons | Function |
|---|---|
|  | Jump to previous image. |
|  | Starts / Stops the reproduction of the stored images. |
|  | Jump to next image. |
|  | Jump to last image. The statistics are reset and all images are evaluated. |

## 9.9 Open and save job or jobset (file)

Jobs can be loaded and stored individually or as a set of jobs as a job set. If several jobs are stored on the sensor, they form a job set which you can save as a single job as an XML file on your PC or an external storage medium.

**Saving a job / job set:**

1. Select "Save job as ..." from the File menu.
2. Select "Save job set under (Backup) ..." from the File menu.

**Opening a job / job set:**

1. Select "Load job ..." or "Load job set (Backup) ..." from the File menu.
2. Activate the button "Start Sensor" to transfer jobs to the sensor.

All the jobs stored on the sensor are deleted when a new job / job set is loaded!

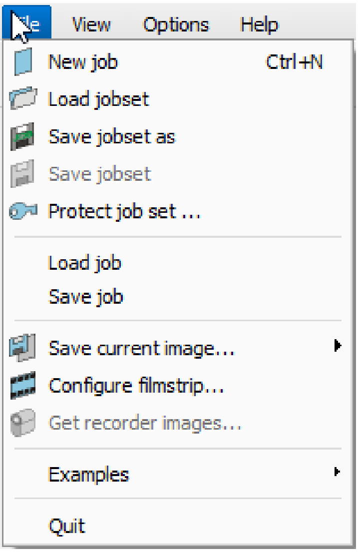



*Fig. 281: SensoConfig Load / save jobs*

**NOTE:**

Job files created with software version 2.x.x.x or later have a different format than job files created with older versions.

Job files created with version 1.x.x.x can be loaded in version 2.x.x.x. The conversion is performed automatically. It may be necessary to adjust the search ranges and fine tune the detector parameters.

Job files created with version 2.x.x.x cannot be used in version 1.x.x.x.

### Automatisches Job-Backup

Beim Klick auf "Start sensor" wird der aktuelle Stand des Jobsatzes auf den Sensor übertragen. Dieser Stand wird zudem automatisch als Jobsatz-Backup abgespeichert. Die Datei findet sich im entsprechenden Ordner des verwendeten Sensortyps (z. B. Allround / Robotic / ...) im Unterordner "Backup". Beim erneuten Starten des Sensors wird diese Datei überschrieben.

- Dateipfad: [[[Undefined variable visor_brand.Dateipfad auto Backup]]]
- Dateiname: "[[[Undefined variable visor_brand.Datei auto Backup]]]"

## 9.10 Protect jobset (file)

In the File menu of SensoConfig it is possible to protect the jobset with a password using the function "Protect job set...". Both the job set and all jobs are protected with a password. They can only be opened with SensoConfig, if the correct password has been entered. If the password is not entered correctly, the jobset cannot be displayed or changed. The VISOR® vision sensor or access to the vision sensor is not blocked, i.e. it operates normally in run mode.

**ATTENTION:**
There is **no** way to recover forgotten or lost passwords.
In case of forgotten or lost passwords, the entire job set must be recreated.

### Assign a password

1. Select "Protect job set ..."via:"SensoConfig/Menu/Protect job set ..."
2. Enter a password and provide additional information if desired.

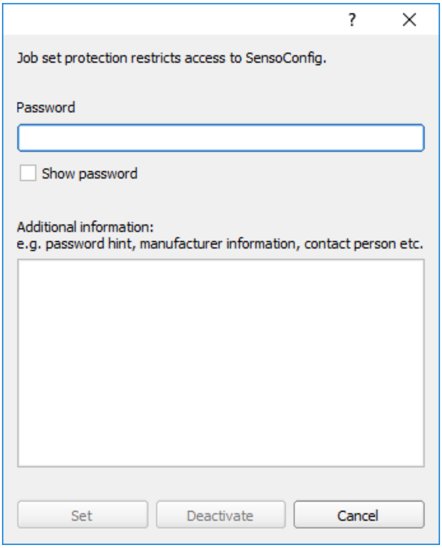



*Fig. 282: Enter a password*

**NOTE:**
The password must be between 1 and 100 characters long.

3. Confirm the entries with the button "Set". Another window to confirm the password opens.
4. Confirm the password by re-entering the password.

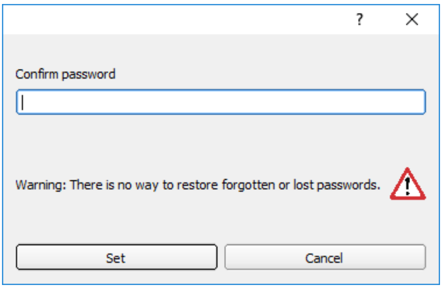



*Fig. 283: Confirm password*

5. Press "Set".
6. Save the protected job set
   a. ... on your VISOR®, by selecting the setup step "Start sensor"
   b. ... via File/Saving a job / job set: (Page 344).

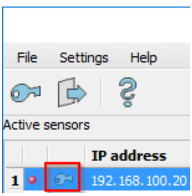
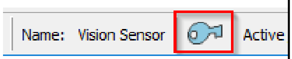
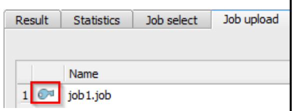
**NOTE:**
When saving the job or the job set, you can choose between the file types "With password protection (*.job)" and "Without password protection (*.job)".

- "With password protection (*.job)": The job / job set with the entered password for the job / job set is saved. The job / job set can only be opened by entering the correct password for the job / job set.
- "Without password protection (*.job)": The job / job set is saved without password protection. The job / job set can be opened and edited at any time without entering the password.

The following table explains how to open a protected / unprotected job in a protected / unprotected job set:

| | **Protected job set on the VISOR® vision sensor** | **Non-protected job set on the VISOR® vision sensor** |
|---|---|---|
| **Open a protected job** | Job set protection remains. To open, the password of the protected job must be entered, then the password of the active job set is accepted. | After opening the protected job and saving the job set or starting the sensor, the password protection is applied to the entire job set. |
| **Open an unprotected job** | Job set protection remains unchanged and is applied to the unprotected job when saving. | Job set remains unprotected. |

A protected job set is marked with a "key symbol". See also the following table:

| **SensoFind** | **SensoConfig / SensoView** | **SensoView** |
|---|---|---|
| <br><br>*Fig. 284: Protected job set, displayed in SensoFind* | <br><br>*Fig. 285: Protected job set, displayed in SensoConfig / SensoView* | <br><br>*Fig. 286: Protected job set, displayed in SensoView* |
| A VISOR® with a protected job set will be shown with a key symbol in the "Active sensors" list. | A protected job / job set is marked with a key symbol in the status bar. | A protected job / job set is marked with a key symbol in the "Upload" tab. |

**Change password**

1. Select "Protect job set ..."in:"SensoConfig/Menu/Protect job set ...".
2. Enter the existing old password and press the "Change" button.
3. Confirm the password by re-entering the password and press the button "Set".
4. Save the new password
   a. ... on your VISOR® vision sensor by selecting the "Start sensor" setup step.
   b. ... via File/Saving a job / job set: (Page 344).

**Deactivate password**

1. Select "Protect job set ..."via:"SensoConfig/Menu/Protect job set ..."
2. Enter the existing password and press the "Deactivate" button.
3. Save the job set
   a. ... on your VISOR® vision sensor by selecting the "Start sensor" setup step.
   b. ... via File/Saving a job / job set: (Page 344).

## 9.11 Filmstrips (file)

In the online mode configuration mode, images are continuously loaded from the sensor into the RAM of the PC. After switching from online to offline mode, you have a maximum of 30 images that you can save as a series of images in a filmstrip file. As an alternative or in addition to the images stored on the sensor, you can load image series or individual images that are stored on your PC or on an external storage medium and combine them to form new filmstrips.

If you select an image in the list, it will be displayed on the right in the preview window in the small format.

### 9.11.1 Storing images from the sensor as filmstrips:

1. First connect the PC to the sensor. Load the memory with images in Free run (connection mode = online).
2. Select radio button "Offline" in the field Connection mode.
3. Choose "Configure filmstrip" from the File menu or click the Filmstrip icon in the toolbar. In the drop-down list that opens, the images loaded by the sensor appear:

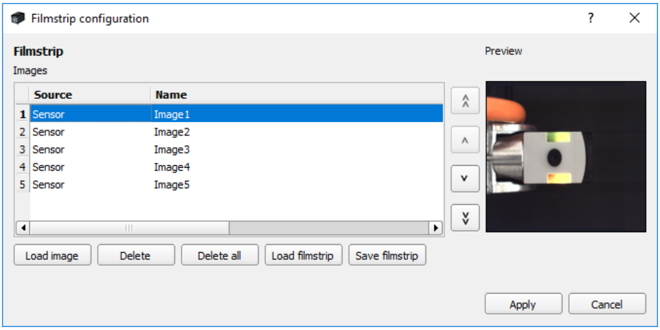



*Fig. 287: Filmstrip*

Now the images can be viewed, rearranged or individual images can be deleted or added. The maximum number of images in a filmstrip is 30.

4. Click on the button "Save filmstrip" under the drop-down list.

All images in the list are saved in the order shown in a filmstrip file (extension .flm) and are available for future simulations.

## 9.11.2 Loading filmstrips and individual images from the PC:

1. Select radio button "Offline" in the field "Connection mode".
2. Select "Configure Film Strip" from the File menu or click on the "Film Strip" icon in the toolbar.
3. Select a filmstrip file from the selection list and click on the button "Load filmstrip" or load individual images from your PC or an external storage medium with the button "Open image".

The loaded images are added to the drop-down list.

The Source column displays the type and location of the file: Filmstrip stored on the PC (film), frame (file) saved on the PC, image in the sensor memory (sensor). After switching from online to offline mode, all entries are from the sensor type.

## 9.11.3 Edit filmstrips:

You can create new movies from the frames in the drop-down list, regardless of their source.

The following functions are available for this purpose:

| Button | Function |
|---|---|
| "<", "<<", ">", ">>" | Change picture order: The selected picture is moved up or down by one place or to the end of the list. |
| Open image | Load further image |

| Button | Function |
|---|---|
| Delete, Delete all | Delete image from the list / Delete all images from the list. (The images on the PC are not deleted.) |
| Cancel | Quit the list without any modification |
| Apply | Load all images into the movie memory on the PC in the order shown. These are then available for display and evaluation in offline mode. |
| Open / save filmstrip | Load filmstrip from PC or save it there |

## 9.12 Image recorder

An image recorder is available in the programs SensoConfig and SensoView. When the recorder is activated, either all images or only fault images are continuously loaded into the internal memory of the sensor. This captures 10 images; the oldest images are each overwritten (ring buffer). The recorded images can then be retrieved and viewed on a PC, stored on the PC or on an external storage medium, and then be available for analysis or simulation purposes in offline mode.

In the program SensoView, you may need to enter a password (if enabled) to access the recorder images (for the user group "Worker", see User Management).

**Activate recorder:**

Activate the video recorder in the "Output" step in the "Image transmission" tab. In the drop-down list of the Recorder parameter, you can select whether all pictures, only the pictures of the good parts, or only the pictures of the bad parts are recorded.

**Selecting and recording images:**

Select "Get recorder images" from the File menu or click the "Rec. images" button (in SensoView).

An image window appears in which you can load, view, and save the images stored in the sensor on the PC:

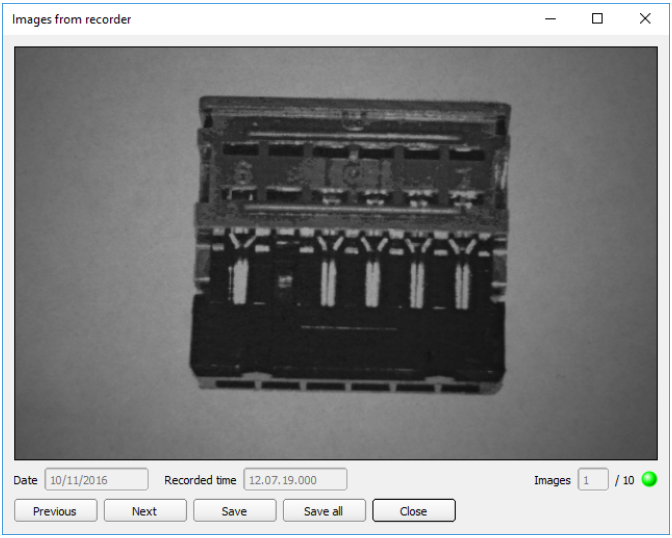



*Fig. 288: Image recorder*

| Parameter | Function |
|---|---|
| Date<br>Recording time | Date and time of the recording. |
| Images | The sequential number of the selected image and the total number of images (max. 10) are displayed in the counter below the image window. |
| Back | Displays the previous image |
| Next | Displays the next image |
| Save | Saves the image displayed on the PC or an external storage medium |
| Save all | Saves all images |

When saving, the images are saved in bitmap format (extension .bmp). The test result (OK or error) and the date associated with the respective image are saved in the file name (format YYMMTT_numbering no._pass/fail.bmp, e.g. 090225_123456_Pass.bmp). If you want to record detailed test results together with the images, use the "Archiving" function in SensoView. If you want to record only a single image with or without an overlay, you can use the "Save Current Image" function in the File menu instead of the recorder.

**NOTE:**
Loading the images from the sensor to the PC erases the data on the sensor. If the recorder window is closed without saving the images first, the images will be lost. In the event of a power failure, images will be lost from the buffer.

## 9.13 Examples (file)

In the menu "File / Examples", some predefined application examples can be loaded. A filmstrip is loaded together with a job file.

## 9.14 Search and feature ranges

You can define search and feature ranges in the configuration steps Alignment and Detectors. These are marked in the image window by different colored frames.

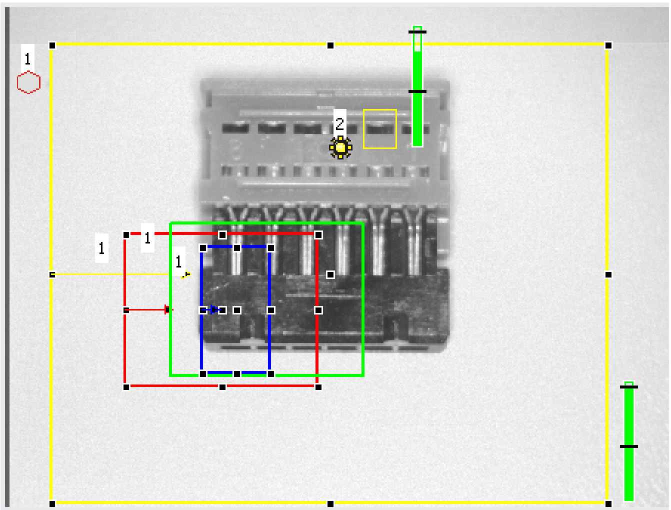


*Fig. 289: Search and feature ranges*

| Frame colour | Meaning |
|---|---|
| Yellow | Search range (ROI) |
| Red | Teach-in area |
| Green | Found features |
| Blue | Position control |
| Yellow dotted | Alignment |

## 9.14.1 Definition of search and feature ranges

When a new detector is created, a yellow frame is displayed which defines the detector's search range. The standard shape of the search range is a rectangle. Depending on the type of detector, the shape "Circle" or "Free shape" can also be selected.

The defined features (red frame) are found (green frame) provided its center is within the search range (yellow frame).

For the detectors Pattern matching, Contour and Contour 3D there is an additional feature area within the search area, which is represented by a red or green frame:

- Red frame = Teach feature
- Green frame = Feature found

If a Position control is defined, an additional blue frame appears (either rectangle, circle, or ellipse).

If an Alignment is defined, its frame is shown in dotted yellow lines.

The respective detector number is displayed at the upper left corner of the frame.

## 9.14.2 Adapting search and feature ranges

The ranges first displayed in standard size and position can be selected / marked in the image or in the detector list and changed in position and size. Eight handles on the frame enable you to adapt the format and size of the frame. Its position can be displaced by clicking anywhere inside the frame. The arrow pointing to the center can be used to change the rotational position of the frame.

The taught-in pattern or contour is displayed in its original size in the first tab in the lower right corner of the screen. Only the frames of the currently active detector selected in the image or in the detector list are displayed in thick line width and with touch points. All other frames that are not selected at this time are displayed with thin or dashed lines (Alignment).

**NOTE:**

- For optimum detection, features must be distinct and not contain any variable parts, e.g. shadows.
- Significant contours, edges, and contrast distinctions are advantageous.
- To minimize the evaluation time, the search range should only be as large as necessary.

#### Result bar

To the right of the search range, the degree of conformity of the sought-after feature with the feature found is displayed as a standing result bar with a set threshold value:

- Green bar = The sought-after feature was found and the preset threshold of minimum match was reached.
- Red bar = The object could not be found with the required degree of conformity.

### Overlays and display

In the "View" menu you can select which graphical representations are displayed.

"Current detector only": show only overlays of the detector currently being processed

"Failed detectors only": show only overlays of failed detectors

Under "View" / "Overlay settings", the overlays in the image (frames in yellow, red, etc.) can be switched on or off as required for each detector or category.

"Result Bar Graph": Show or hide result bar

## 9.15 Simulation mode: Simulation of jobs (offline mode)

You can also create and test your configuration without a connected sensor using saved filmstrips (= image series). A simulation may e.g. be sensible to prepare a configuration or to optimize an online configuration.

**NOTE:**

- In the delivery state of SensoConfig, there are some prepared filmstrips available.
- Additional image acquisition options: Image recorder (Page 360).

To start the simulation mode, select a sensor type in SensoFind from the list "Sensors for simulation mode". Via double click or click on "Config", SensoConfig opens in simulation mode for this sensor type.

## 9.16 Color models

There are so-called color models for the description of colors. The VISOR® Object color can work with various color models.

The following color models can be used:

Color model RGB

Color model HSV

Color model LAB

## 9.16.1 Color model RGB

An RGB color space is an additive color space that replicates color perceptions by additive mixing of three primary colors (red, green, and blue).

The RGB color space is described as a linear color space, as a cube with the three axis: red, green, and blue.

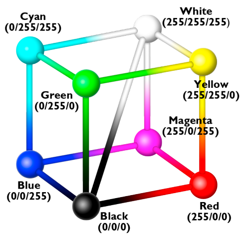



Fig. 290: Color model RGB

Red, green, blue, each 0-255
The RGB color space is used by both the image chip and the screen to define the colors. However, the image chip and the screen have different sensitivities within the individual color channels. Because of this, there must always be a compensation, so RGB is never equal to RGB.

### Linear RGB

RGB values are calculated as linear RGB values, as the sensor chip delivers linear RGB values. The advantage of the linear RGB values is that there is a linear relationship between physical influence and RGB values.

Example: Doubling the exposure time leads to a doubling of the RGB value under constant lighting conditions.

## 9.16.2 Color model HSV

In terms of color adjustment, the HSV color space is preferred over the alternatives RGB and Lab because it resembles human color perception.

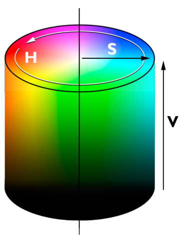



Fig. 291: Color model HSV

- H (hue) as a color angle on the color circle (e. g. 0° = red, 120° = green, 240° = blue)
- S (saturation) in percent (0 % = light gray, 50 % = low saturated color, 100 % = maximum saturated color)
- V (value) in percent (0 % = dark, 100 % = full brightness)

### 9.16.3 Color model LAB

LAB or L*a*b* color space is described by a three-dimensional coordinate system:

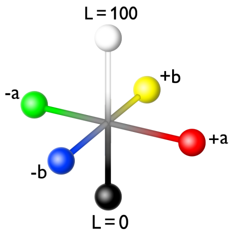



*Fig. 292: Color model LAB*

- An a*-axis describes the green and red components of a color; negative values stand for green and positive values stand for red. Range of values from -87 to +99.
- A b*-axis describes the blue and yellow components of a color; negative values stand for blue and positive values stand for yellow. Range of values from -108 to +95.
- An L*-axis describes the lightness (luminance) of the color with values from 0 to 100.

One of the most important features of the L*a*b* color model is its device independence. This means that the colors are defined independently of the way they are produced and reproduced. LAB values are calculated from linear RGB values. This is based on the standard illuminant D65 and 2° observer.

# 10 VISOR® Software – SensoView

This program is used to monitor / verify connected sensors and to analyze test results. No new settings can be made on the sensor.


- Image display (Page 357)
- Result tab (Page 362)
- Statistic tab (Page 364)
- Job tab (Page 364)
- Upload tab (Page 365)
- Freeze image (Page 358)
- Image recorder (Page 360)
- Archiving of test results and images (Page 359)


Beyond the mere display, it is only possible to switch between pre-existing jobs on the sensor, or predefined job sets can be uploaded from the PC / PLC to the sensor by the authorized worker. Thus, this display tool mainly serves to visualize images and results and to change jobs at e.g. part change on the machine.

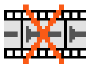
## 10.1 Image display

The graphic display of the image and the test results in the image window depends on the settings in the "Image transmission" tab in the job settings ("Image transmission" tab, Chapter "Image transfer parameters" in program SensoConfig):

- Image transfer active: The current image as well as the frames for the defined search, feature and position ranges, and the found characteristics are displayed.
- Image transferal inactive: Only the frames for the defined search, feature and position areas, and the found features are displayed (the current picture is not displayed).

To the right of the search area of the respective detector, the degree of agreement of the sought-after with the found feature is displayed as a standing result bar with a set threshold value:

- Green bar: The parameter searched for has been found and the preset threshold value for concordance has been reached.
- Red bar: The object could not be found with the required degree of conformity



**ATTENTION:**
When this icon appears on the life image, it is indicating the image visualization / image storage on the PC is running slower than the image processing on the VISOR®. Not all images taken by the VISOR® are displayed anymore. This may lead to image loss when using poor image archiving.
If the icon appears frequently, programs opened on the PC should be closed in the background to provide more PC performance.

In the mSensoViewenu / view, you can configure the graphical representation of the test results.

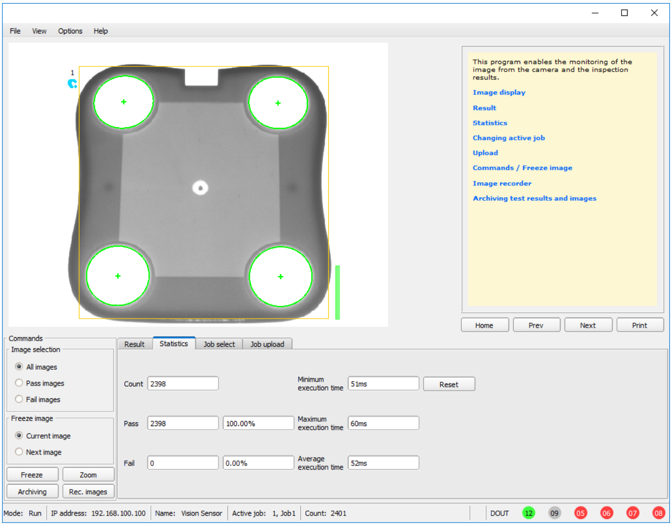



*Fig. 293: SensoView*

Except archiving, all functions of SensoView are also available in the SensoConfig program .

## 10.2 Commands

### 10.2.1 Freeze image

With the "Freeze image" button, you can request individual images of the desired type (Current image, Next image, Next defect image) and hold them in the image window for display.

The desired frame is displayed and the frame counter remains at the corresponding frame number.

Press "Continue" to end the frozen image state.

### 10.2.2 Zoom

With the button "Zoom", the image is opened in a new window with enlarged display.

### 10.2.3 Archiving of test results and images

You can archive images with and without markups and check results on your PC or on an external storage medium for analysis or simulation purposes (see Offline Mode).

The execution of these functions may require the entry of a password (user group worker, see User Administration).

**Configuring archiving:**

1. Select "Configure Archiving" from the File menu.
   A dialogue box appears with the following options:

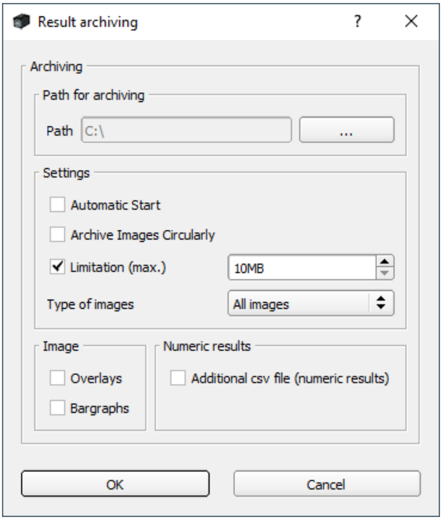



*Fig. 294: Configure archiving*

| Parameter | Function |
|---|---|
| Path for archiving | Directory in which the archived file(s) are stored. |
| Settings, Automatic start | Starts archiving automatically after starting SensoView. |

| Parameter | Function |
| --- | --- |
| Settings, Cyclic overwriting | Activates cyclic overwriting of oldest images if storage limit is reached. |
| Settings, Storage limit | Here, it is possible to limit the data volume. |
| Settings, Image type | In this drop-down list, you can specify which images (all images or only good or bad images) should be saved. |
| Overlays,<br>Bar graph result | Selection of the graphic representations to be archived in the image. |
| Numerical results | If "Log" is activated, numerical result data such as coordinate values, etc. are archived in an additional .csv file.<br>The "Legacy" / "Configured" setting determines the format of the .csv file.<br>With "Legacy" *1), the contents of the .csv file are specified; "Configured" is freely configurable via "Output / Data output".<br>*1) The storage mode "Legacy" is obsolete and only provided for reasons of backward compatibility. It will be omitted in one of the next versions. |

2. Select the desired options and confirm your choice with OK.

**Start / end archiving:**

Click on the "Archive images" button in the "Commands" window to start or end the archiving function in the above-mentioned settings. The status bar displays the name of the image file currently being saved. Archiving is carried out for as long as the button "Archive Images" is pressed.

## 10.2.4 Image recorder

An image recorder is available in the programs SensoConfig and SensoView. When the recorder is activated, either all images or only fault images are continuously loaded into the internal memory of the sensor. This captures 10 images; the oldest images are each overwritten (ring buffer). The recorded images can then be retrieved and viewed on a PC, stored on the PC or on an external storage medium, and then be available for analysis or simulation purposes in offline mode.

In the program SensoView, you may need to enter a password (if enabled) to access the recorder images (for the user group "Worker", see User Management).

**Activate recorder:**

Activate the video recorder in the "Output" step in the "Image transmission" tab. In the drop-down list of the Recorder parameter, you can select whether all pictures, only the pictures of the good parts, or only the pictures of the bad parts are recorded.

### Selecting and recording images:

Select "Get recorder images" from the File menu or click the "Rec. images" button (in SensoView).

An image window appears in which you can load, view, and save the images stored in the sensor on the PC:

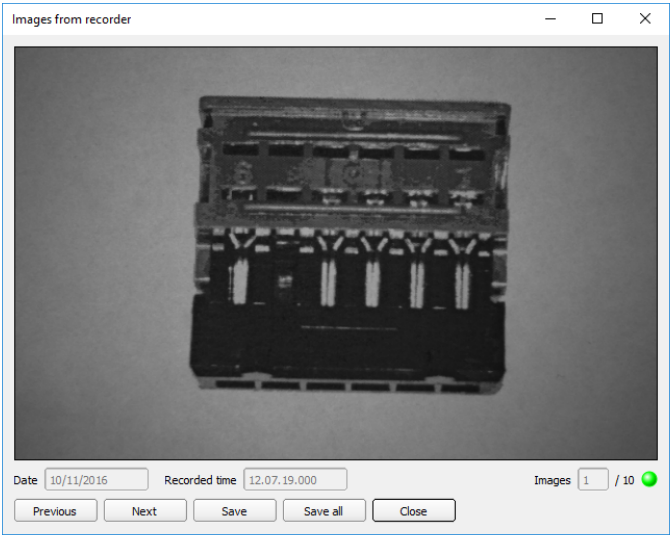



*Fig. 295: Image recorder*

| Parameter | Function |
|---|---|
| Date<br>Recording time | Date and time of the recording. |
| Images | The sequential number of the selected image and the total number of images (max. 10) are displayed in the counter below the image window. |
| Back | Displays the previous image |
| Next | Displays the next image |
| Save | Saves the image displayed on the PC or an external storage medium |
| Save all | Saves all images |

When saving, the images are saved in bitmap format (extension .bmp). The test result (OK or error) and the date associated with the respective image are saved in the file name (format YYMMTT_numbering no._pass/fail.bmp, e.g. 090225_123456_Pass.bmp). If you want to record detailed test results together with the images, use the "Archiving" function in SensoView. If you want to record only a single image with or without an overlay, you can use the "Save Current Image" function in the File menu instead of the recorder.

**NOTE:**
Loading the images from the sensor to the PC erases the data on the sensor. If the recorder window is closed without saving the images first, the images will be lost. In the event of a power failure, images will be lost from the buffer.

## 10.3 Result tab

This function executes the defined job on the PC and displays the "Result Statistic" window with detector list and evaluation results. The execution times are not updated in this mode because they are not available from the sensor.

In Run mode, the detailed test results of the detector marked in the drop-down list are displayed.

In the image window – if adjusted – the image, the search and feature ranges, and the result graphs are displayed.

The parameters displayed vary according to the type of detector selected:

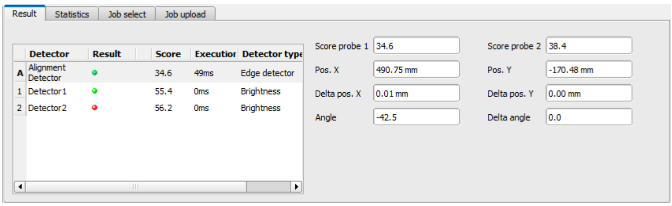



*Fig. 296: SensoView, Result*

| Displayed result parameters | For detector type | Function |
|---|---|---|
| Result | all | Part / feature recognized (detected = green, not detected = red) |
| Score value 1 ... n | all | Degree of conformity of pattern found with pattern taught |
| Execution time | all | Cycle time for an evaluation in ms |
| Distance | Caliper | Calculated distance |
| Position X 1 ... n,<br>Position Y 1 ... n | Pattern matching., Contour, Caliper, Contour 3D | Coordinates of feature found (center point) |

| Displayed result parameters | For detector type | Function |
|---|---|---|
| Delta X, Delta Y | Pattern matching, Contour, Contour 3D | Deviation of the finding coordinates from the teach-in position / by Alignment |
| Position control | Pattern matching, Contour | Position found within the defined position frame |
| angle | Pattern matching, Contour, Contour 3D | Orientation (absolute angle) of parameter found |
| Delta angle | Pattern matching, Contour, Contour 3D | Angle deviation between parameter taught and parameter found |
| Scaling | Contour | Scale of contour found in contrast to taught contour |
| Result index | Color List | Number of list entry |
| Color distance | Color List | Distance of measured color to taught color |
| Red (Color model RGB) | Color List, Color Value | Mean value red |
| Green (Color model RGB) | Color List, Color Value | Mean value green |
| Blue (Color model RGB) | Color List, Color Value | Mean value blue |
| Hue (Color model HSV) | Color List, Color Value | Hue of color |
| Saturation (Color model HSV) | Color List, Color Value | Saturation of color |
| Brightness (Color model HSV) | Color List, Color Value | Brightness of color |
| Luminance (Color model LAB) | Color List, Color Value | Luminance value of color |
| A (Color model LAB) | Color List, Color Value | A- Value of color |
| B (Color model LAB) | Color List, Color Value | B- Value of color |

To see the test results of another detector, mark it in the drop-down list.

You can archive inspection results and statistical evaluations, including the selected graphical representations, in the SensoView program.

## 10.4 Statistic tab

In the Run mode, the statistical data of the test process is displayed in the Statistic tab. The statistical data displayed is identical for all types of detectors:

| Parameter | Function |
|---|---|
| All evaluations | Total number of inspections |
| Good parts | Number of inspections with result "OK" |
| Bad parts | Number of inspections with result "Error" |
| Min./max./mean execution time | Min./max./mean execution time for evaluation in ms |

All statistical values can be reset to zero with the "Reset" button.

You can archive inspection results and statistical evaluations, including the selected graphical representations, in the SensoView program.

## 10.5 Job tab

In the Job tab, the jobs available in the sensor are displayed in the selection list. Here you can switch between different jobs stored in the sensor. The green arrow ( ) marks the active job.

Running functions that would cause the active sensor to be stopped (job change, job upload, and fetching recorder images), will also require for a password, if enabled in SensoFind, to be entered (Worker user group; please refer to User administration).

**Password levels**

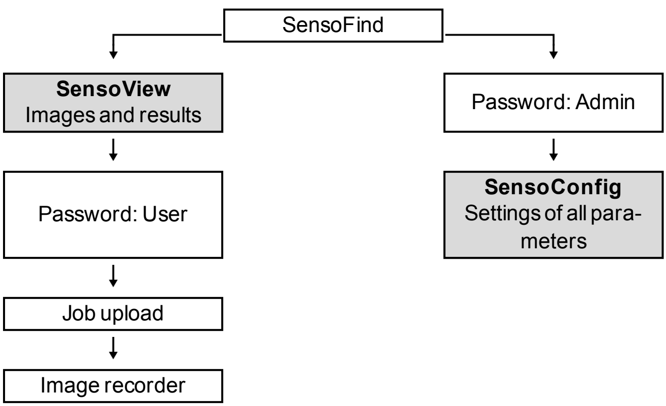



*Fig. 297: Password levels*

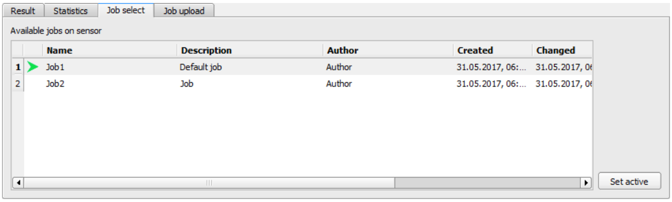



*Fig. 298: SensoView, Job switch*

### Procedure

Select a job from the list and activate it with the "Activate" button.

The previous job is deactivated; the selected job is now active.

**NOTE:**
When the job changes and the operating mode changes from Run to Config mode, the following special states of the outputs occur:

- The buffer of the delayed outputs is deleted from "Run" to "Config" when the job is changed and the operating mode is changed.
- Digital outputs: These are reset to the default settings (defaults) when the job is changed and the operating mode is changed from "Run" to "Config". The basic settings are defined by "Invert" in Output tab / Digital output tab. "Invert" inverts the basic setting of the digital output and, at the same time, the result.
- Ready and Valid: Ready and Valid signals when the job is changed and the operating mode changes from "Run" to "Config" that the sensor is not ready and the results are not valid (low level).

## 10.6 Upload tab

In the Upload tab, you can load new jobs or entire job sets from the PC into the sensor memory. The available jobs or job sets are displayed in the drop-down list.

Jobs and job sets can be created in the SensoConfig program and stored there under File / Save Job / Save job set as ...

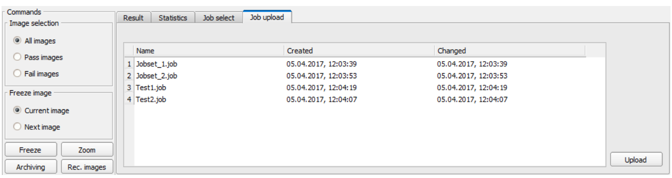



*Fig. 299: SensoView, Upload tab, Load job set*

**NOTE:**

- A job set consists of one or more jobs stored in the sensor or on the hard disk.
- Performing functions that cause the active sensor to stop may require the entry of a password (user group worker, see User Administration).
- Select a job or job set from the list and load it onto the sensor with the "Upload" button.
- This action deletes all jobs previously stored on the sensor!

## 10.7 VISOR® – SensoWeb

SensoWeb is used (as well as SensoView) to monitor / check connected sensors and to analyze test results. No new settings can be made on the sensor.

In contrast to the program SensoView, the display takes place in the browser (no software installation is required for display).

### Start SensoWeb

1. In SensoConfig, operating step Output, select Interfaces tab.
2. Activate the checkbox on the right side of the line SensoWeb.
3. In SensoConfig, start the sensor via button "Start sensor" .
4. Open the browser.
5. In the address bar of the browser, enter the IP address of the sensor (visible in SensoFind), in the format: "http://your sensor IP", e.g. "http://192.168.100.100" (Default).

**NOTE:**

- The following browsers are supported: Microsoft Internet Explorer® from IE10, Edge, Google Chrome® and Mozilla Firefox®.
- With http://192.168.100.100/zoom.html (or alternatively the IP address of the sensor), the enlarged view can be accessed directly.
- Only one browser connection is allowed per VISOR® vision sensor.

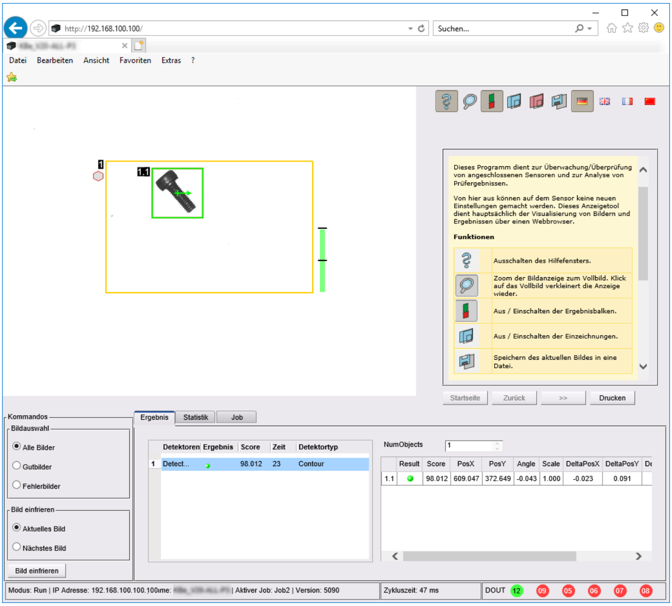
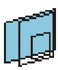
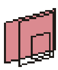



*Fig. 300: View SensoWeb in the Browser / Results*

### Buttons in the menu bar

| Symbol | Function |
|---|---|
|  | Switching off the help window. |
|  | Zoom the image display to full screen. Clicking on the full screen reduces the display again. |
|  | Switching the result bars off / on. |
|  | Switching the overlays off / on. |
|  | Overlay of failed detectors only. |
|  | Save the current image to a file. |

| Symbol | Function |
|---|---|
|  | Switches between languages |

**Functions of SensoWeb**

| Tab / Parameter | Function |
|---|---|
| Result tab | Display of the detector results of the sensor |
| Statistic tab | Overview of evaluations, good and bad parts as well as cycle and execution times |
| Job tab | Display of the jobs available on the sensor |
| Image selection | Selection of the images to be displayed: "All images" / only "Good images" / only "Error images |
| Freeze image | Option to "freeze" the image display. Selection of "Current image" / "Next image".<br>Only the image display is stopped. The execution of the sensor in the background continues. |
| Display in the status bar (below) | • Mode, IP address, name, active jobs and version of the vision sensor<br>• Cycle time<br>• DOUT: Status of the sensor outputs |

# 11 Communication

**NOTE:**
For more information on the topic of communication, you can download the VISOR® Communications manual from the Download area of the SensoPart website (www.sensopart.com). The VISOR® Communications Manual is also part of the VISOR® Communications manual installation and can be found in the \Documentation subfolder.

## 11.1 Network connection

### 11.1.1 Integrating the VISOR® into the network / gateway

SensoFind/Active sensors will show a list with all the VISOR® vision sensors that are found on the same network segment on the PC on which is running SensoFind. To update the list, press the "Find" button, e.g. for sensors that were only activated after viewing SensoFind.

For sensors which are installed in the network but are located in a different network segment via a gateway, please enter the corresponding sensor IP address under "Add active sensor" and press the button "Add". The corresponding sensor will now also appear in the "Active sensors" list, and you will be able to access it and work with it.

## 11.1.2 Network connection: Direct connection

Establishing a direct Ethernet connection between the VISOR® vision sensor and the PC

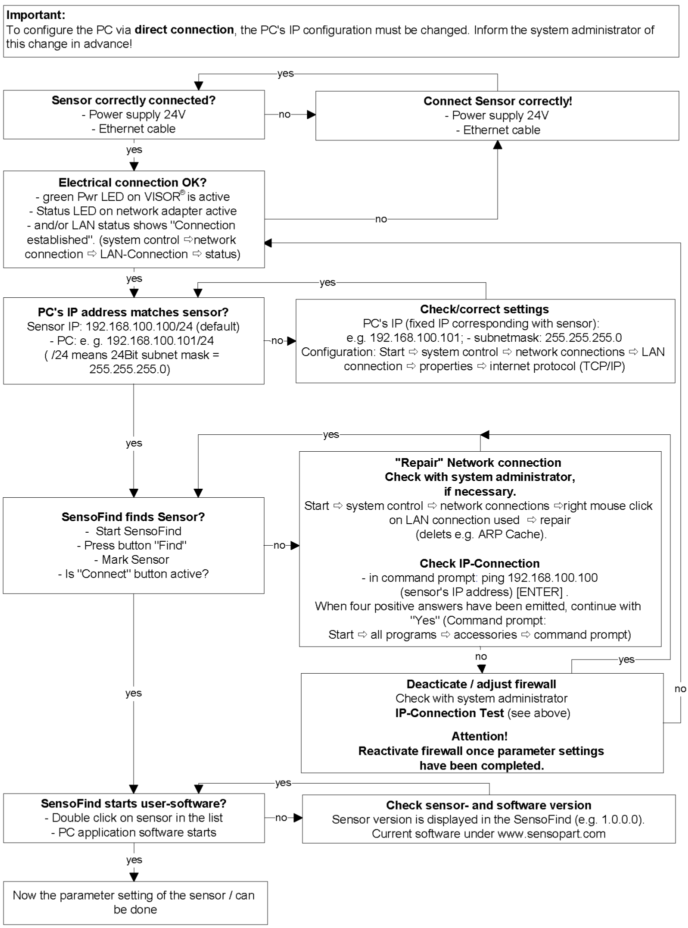



*Fig. 301: Direct connection sensor / PC, procedure and troubleshooting*

### 11.1.3 Network connection: Connection via network

Establishing an Ethernet connection between the VISOR® vision sensor and the PC through a network.

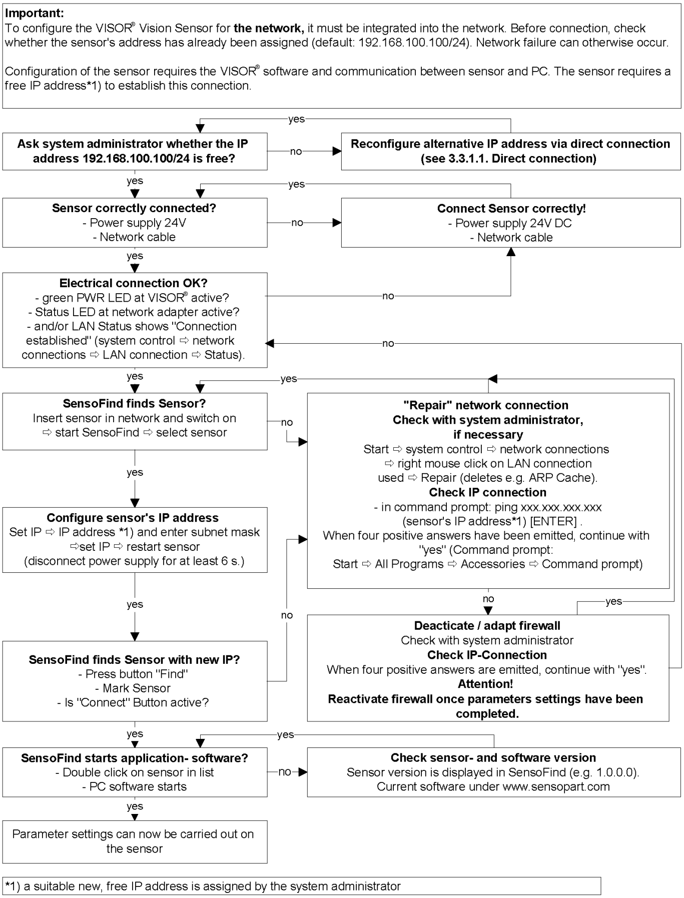



*Fig. 302: Connection via network sensor / PC, procedure and troubleshooting*

### 11.1.4 Used Ethernet ports

If you are integrating the VISOR® into a network, make sure that an admin opens the following ports if necessary. This is only the case if these ports were previously explicitly blocked in the company network or by a firewall installed on the PC.

The following ports are used for communications between the VISOR® software (PC) and the VISOR®:

- Port 2000, TCP
- Port 2001, UDP Broadcast (to find sensors via SensoFind)
- Port 2002, TCP
- Port 2003, TCP
- Port 2004, TCP

The following ports are used for communications between the PLC (PLC or control PC) and VISOR® vision sensor:

Process interfaces:

- Ethernet
  - Port 2005, TCP (Implicit results, i.e. user-configured result data)
  - Port 2006, TCP (Explicit requests, e.g. trigger or job switch)
- EtherNet/IP:
  - Port 2222, UDP
  - Port 44818, TCP
- PROFINET:
  - Port 161, UDP
  - Port 34962, UDP
  - Port 34963, UDP
  - Port 34964, UDP
- Service:
  - Port 22, TCP
  - Port 1998, TCP
- SensoWeb:
  - Port 80

**NOTE:**
If Ports 2005 or 2006 are changed in the configuration software, they must also be changed accordingly in the firewall by an administrator.

### 11.1.5 Access to VISOR® through network

Exemplary values for IP, etc.

Access to VISOR® 1 from PC 1, if on the same subnet

- Via SensoFind (/find)

Access to VISOR® 2 from PC 1, if on a different subnet

Only if:

- Gateway is set correctly in Sensor 2 (here to 192.168.30.1) - and
- in SensoFind via Add IP, the sensor IP of Sensor 2 is set correctly
  > after this, VISOR® 2 will also appear in the "Active sensors" list in SensoFind!

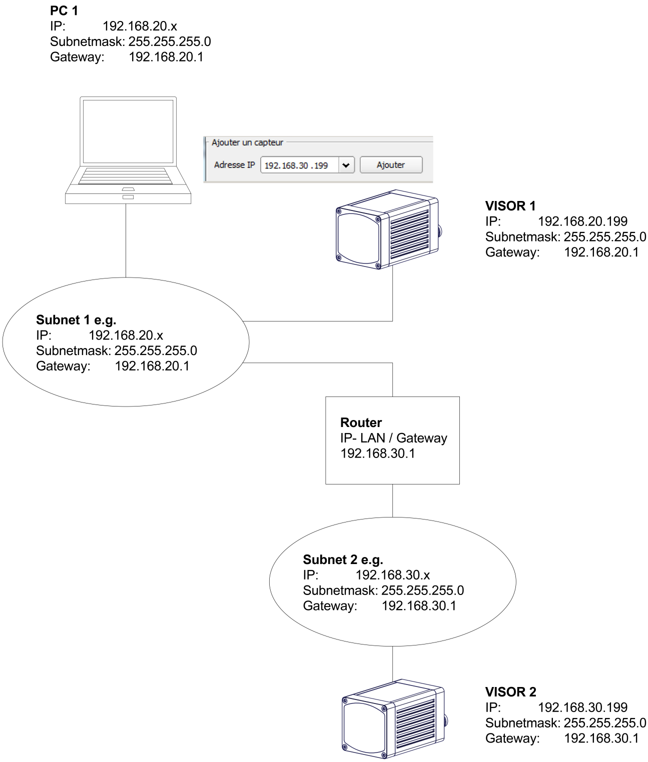



*Fig. 303: Access to VISOR® through network, same or other subnet*

### 11.1.6 Access to VISOR® through the Internet / World Wide Web

Exemplary values for IP, etc.

Access from PC 1 (company network 1), through the Word Wide Web, to company network 2 to VISOR® 1.

1. On PC 1 (company network 1SensoFind) enter and add the IP WAN of Router 2 (company network 2) under "Add active sensor" in (here in this example: 62.75.148.101)
2. On router 2, open the ports that the sensor will be using (please refer to section: Used Ethernet ports).See Chapter:

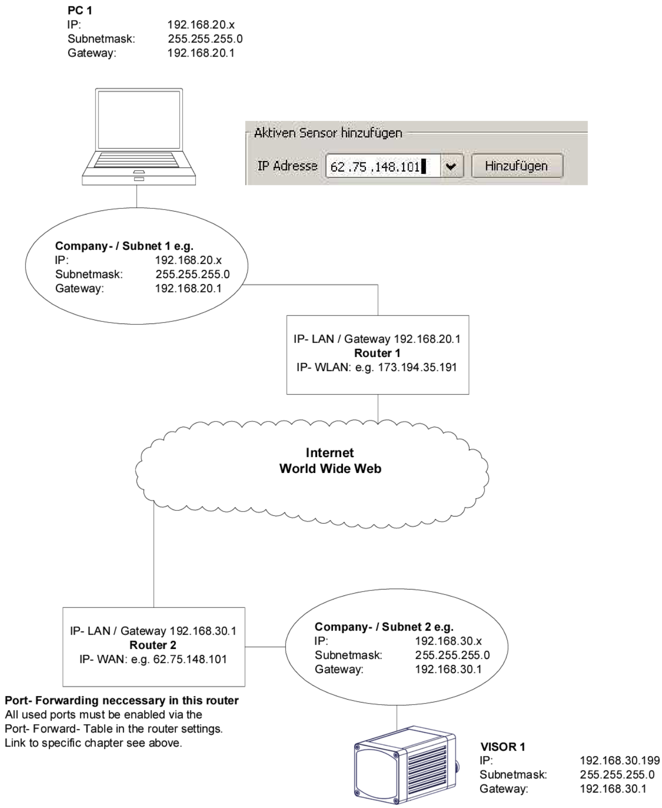



*Fig. 304: Access to VISOR® through the Internet / World Wide Web*

## 11.2 Job change

## 11.2.1 Job change with digital inputs

To switch between several jobs which are already stored on the sensor, the following options are available via digital inputs:

See also Chapter I/O mapping tab (Page 319), timing diagrams and comments

### 11.2.1.1 Job 1 or Job 2

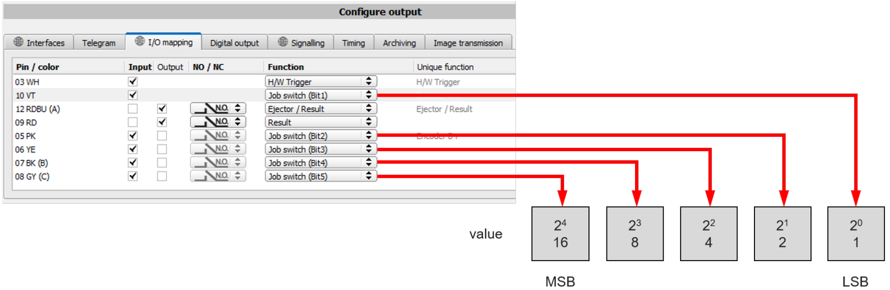
To switch between Job 1 and Job 2, an arbitrary value can be defined in SensoConfig/Output/I/O Mapping with the function "Job 1 or 2". After applying the according level at this input, Job 1 or Job 2 will then be executed (Low = Job 1, High = Job 2). See also Chapter I/O mapping tab (Page 319) / Function of inputs et seqq.

### 11.2.1.2 Job 1 … 255 via a binary input bit pattern

To switch between up to 255 jobs via a binary input combination of up to 8 inputs, all required inputs are assigned under SensoConfig / Output /I/O Mapping with the corresponding function "Job change bit x". The corresponding binary input patterns, as shown in the image below, then directly switch to the corresponding job when it is created. Please refer to the following section as well: I/O mapping tab (Page 319) / Input functions.

**NOTE:**

- The job change will start immediately after the input combination changes.
- The display of the active job in the status bar changes with the first, following trigger.
- The mapping of the I/O's is not fixed. It depending on the settings in SensoConfig/Output/I/O Mapping.
- The level change of the associated inputs must be made simultaneously (within a maximum of 10 ms, all levels must be stable).




| e.g. Job No. | Input 5 | Input 4 | Input 3 | Input 2 | Input 1 |
|---|---|---|---|---|---|
| 1 | false | false | false | false | true |
| 2 | false | false | false | true | false |
| 6 | false | false | true | true | false |
| 13 | false | true | true | false | true |
| 31 | true | true | true | true | true |
| No change | false | false | false | false | false |

*Fig. 305: Job change binary (in this example with 5 inputs and therefore up to 31 jobs)*

## 11.2.2 Job change Ethernet

For more information, please refer to the VISOR® Communications manual.

## 11.2.3 Job change with SensoView

In the applicatioSensoViewn , a job switch can be made, or completely new job sets can be uploaded to the sensor. See also Chapter Job tab (Page 364)

In the tab "SensoView/Job", all jobs stored on the sensor are displayed. If there is more than one job in the sensor memory, one of them can be marked in the list and activated with "Activate".

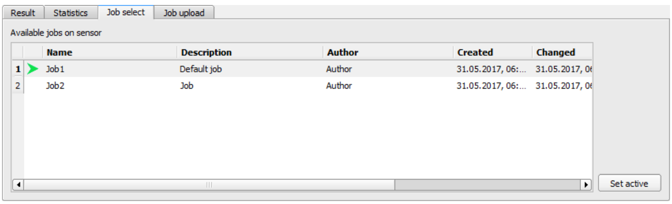



*Fig. 306: SensoView, job change*

In the tab "SensoView/Upload", all available job sets on the PC are displayed. These can be marked in the list and then uploaded to the sensor with "Upload".

**ATTENTION:**
By uploading a new job set, all jobs in the sensor memory are deleted.

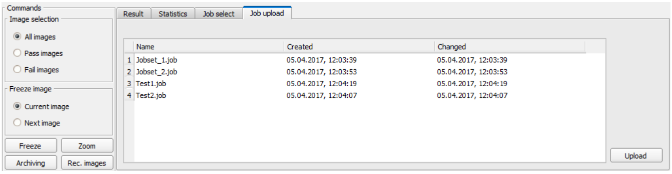



*Fig. 307: SensoView, Upload job*

## 11.3 PC archiving (SensoView)

Via SensoFind, images and numerical data (in .csv format) can be stored in a folder on the PC.

Configuration (directory, etc. ...) of the archiving is done via SensoView in the menu "File/Configure Archiving". This function is available on PC only.

**Step 1:**

Start SensoView by clicking on the button "View" in SensoFind.

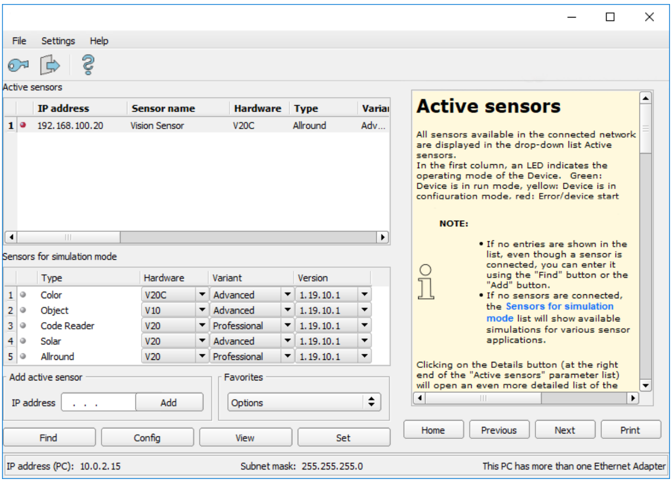



*Fig. 308: SensoFind*

SensoView is opened.

The conditions for correct image display are:

- Free run is set or
- at least one trigger occurs
- Image transferal is active at: SensoConfig/Job/Image Transferal

**Step 2**

Select "Configure Archiving" under SensoView/File.

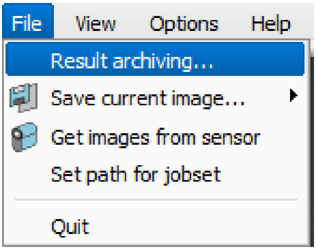



*Fig. 309: SensoView, Archiving*

The following dialog for setting the parameters for archiving is then displayed.

| Parameter | Function |
|---|---|
| Path for archiving | Directory in which the archived file(s) are stored. |
| Settings, Automatic start | Starts archiving automatically after starting SensoView. |
| Settings, Cyclical Overwriting | Activates cyclic overwriting of oldest images if storage limit is reached. |
| Settings, Storage limit | Here, it is possible to limit the data volume. |
| Settings, Image type | In this drop-down list, it can be specified which images (all image or only good or bad images) should be stored. |
| Overlays,<br>Bar graph result | The image data can be stored in various ways.<br>By activating "Overlays", the frames of detectors and Alignment are also saved.<br>By activating "Result Bar Graph", the result bars of detectors and Alignment are also saved.<br>If none of the options are activated here, the image data are saved in a raw state. |
| Numerical results | If "Log" is activated, numerical result data such as coordinate values, etc. are archived in an additional .csv file.<br>The Legacy / Configured determines the format of the CSV file. For "Legacy", the content of the .csv file is predefined; for "Configured", this can be freely configured via "Output / Data Output". |

Select the desired options and confirm your choice with OK.

**Start / end archiving:**

Click on the button "Archive Images" in the window "Commands" to start or end the archiving function. The status bar displays the name of the image file currently being saved. Archiving is carried out for as long as the button "Archive Images" is pressed.

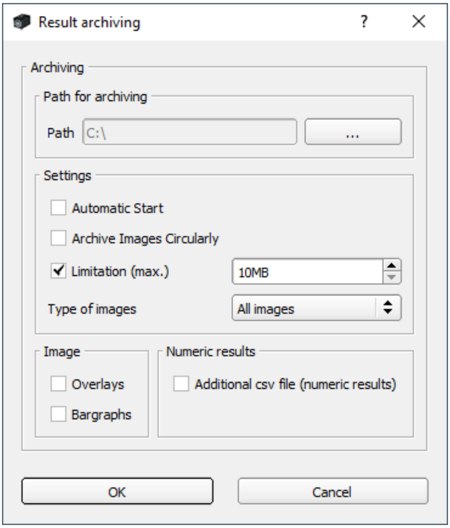



*Fig. 310: SensoView, Configure archiving*

## 11.4 Archiving via ftp or smb

Here, images and numeric data (in CSV format) can be archived by the sensor via ftp / smb.

This archiving is configured under "Output/Archiving".

For this type of archiving:

a. **for ftp**: The sensor is an "ftp client" and "writes" the data to an "ftp server" available in the network.
With Job Start, the sensor connects to the ftp server.

b. **for smb**: The sensor "writes" its data directly into a network-shared directory. With Job Start, the sensor connects with this directory.

When using this type of image and result data archiving in normal use cases, neither the SensoFind nor the SensoConfig PC application is active. Instead, only an FTP or SMB server configured to communicate with the VISOR® will be active.

### 11.4.1 Example: Archiving via ftp

In the example here, an exemplary FTP communication was established with the freely available FTP server software "Quick'n Easy FTP Server" and image and result data were saved on the hard disk of the PC.

The account wizard was used to create a user account named "VISOR®_FTP" on the FTP server. A password and a path for data storage are specified, and upload and download are enabled.

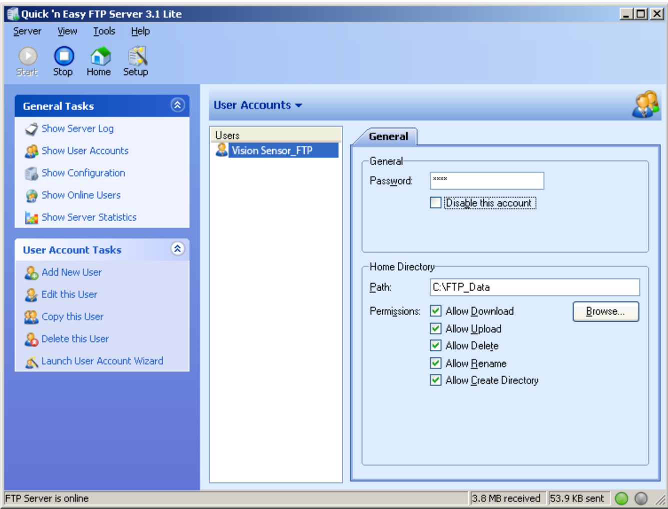



*Fig. 311: FTP Server*

In SensoConfig, in Output/Archiving, the appropriate settings for the FTP server still need to be configured on the VISOR®. In addition:

- Archive type = FTP
- IP address = IP of the PC on which the FTP server is running (can be seen in SensoFind in the status bar, bottom left)
- User name = Name of the user account in the FTP server
- Password = Password assigned in the FTP account (optional)

With this, the matching settings necessary for the FTP are made.

Here, additional settings, e.g. data name, max. number of files, and storage method (e.g. "cyclic") can / should be made.

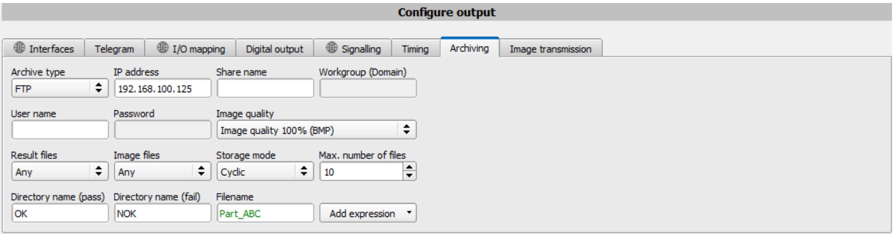



*Fig. 312: FTP server, settings in SensoConfig*

Once these settings have been configured and transferred to the VISOR® using "Start sensor," the image and result data will be saved on the PC in the specified directory without the SensoFind, SensoConfig or SensoView applications needing to be active.

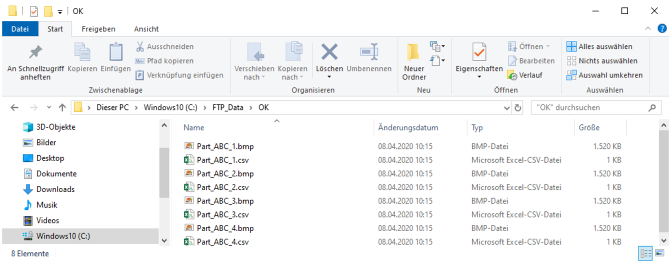



*Fig. 313: Transferring files with FTP*

Archiving via smb takes place analogously via an smb server, which must be set accordingly.

## 11.4.2 Example: Archiving via SMB

For data and / or image archiving via SMB (server message block), a folder must be shared from the PC.

The following example shows some exemplary settings for setting up data archiving via SMB.

#### 11.4.2.1 Settings for SMB on PC: Create folder and share it

1. By right clicking on the folder (here "Test_SMB"), select the menu item "properties".

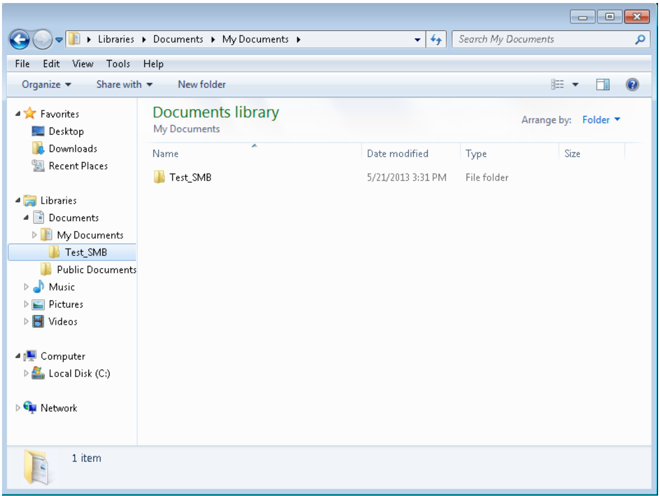



*Fig. 314: Create folder to be described, here for example: "Test_SMB".*

2. In the following dialog, "Properties of Test_", open the tab "Sharing" and click on "Advanced Sharing".

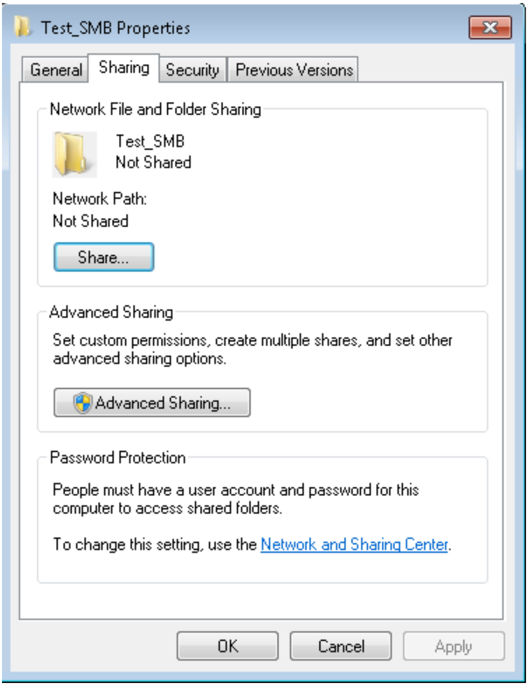



*Fig. 315: Folder sharing > Advanced sharing*

3. In the dialog "Advanced Sharing", activate "Share this folder". Here, the name of the folder "Test_SMB" is suggested as a "share name". Here, any other name can be set. In this example the suggested folder name is used.
   Important: You will later on need to enter this share name in the VISOR® SMB interface exactly the way it is entered here!

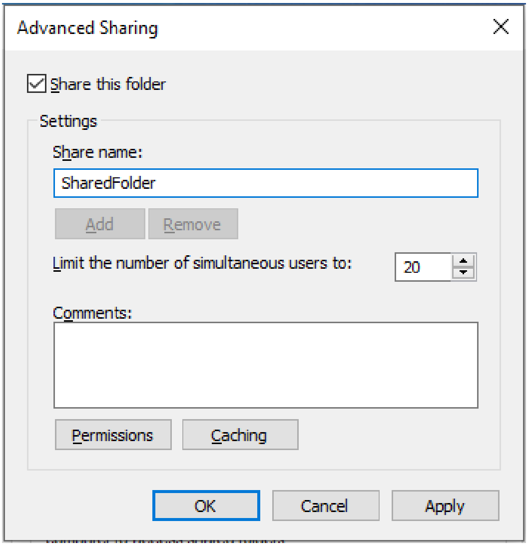



*Fig. 316: Set share name*

4. By clicking on "Permissions", the following dialog appears.

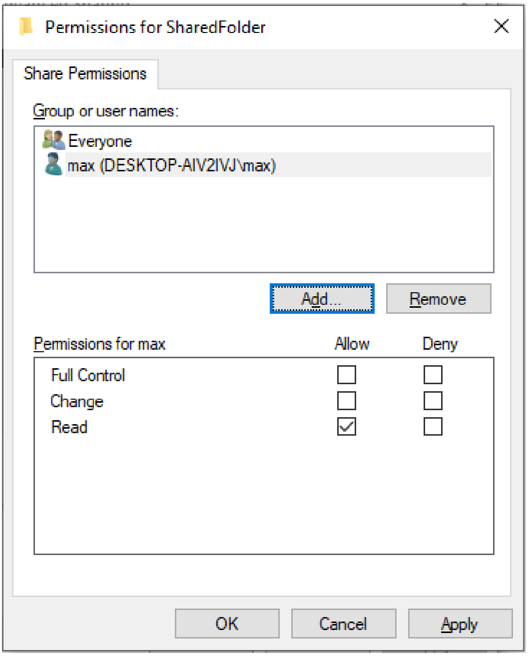



*Fig. 317: Set permissions*

5. In the window "Test_SMB Permissions", select a user (here "fsc") (for which the user name and password are known). You will need to enter the username and password in the VISOR® SMB interface later on.
6. Activate "Full control" and close the dialog with "Apply" and "OK".
7. Then also close the dialog "Advanced Sharing" and "Test_SMB Properties" with "Apply" and "OK".
8. Access for the user selected here has now been set up on the PC. The corresponding settings in the VISOR® "SensoConfig" interface can be configured now.

### 11.4.2.2 SMB setup

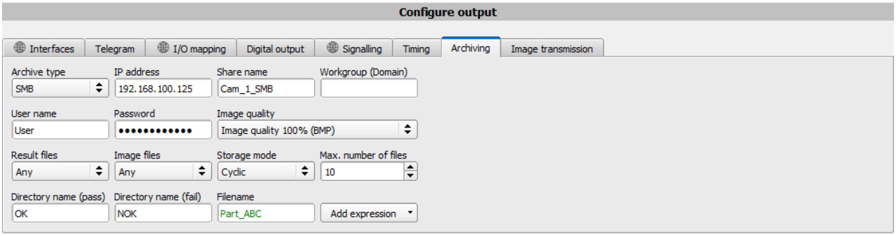



*Fig. 318: Settings in VISOR® SMB Interface*

After starting SensoConfig, select "SMB" under Output/Archiving/Archiving type.

Make the following entries:

- IP address: IP address of the PC (this can be found with the command "ipconfig" under Start/Run/cmd, see following screenshot). In this example: 192.168.60.14

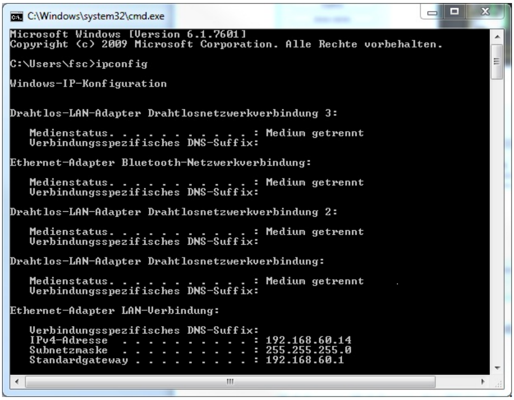



*Fig. 319: IP address of the PC via Start/Run/cmd/ipconfig*

- Share name: Here, enter share name as previously specified on the PC in the dialog "Advanced Sharing".
- Workgroup (domain): Optional! Enter name of workgroup.
- User name and password: Depending on the selection made in the dialog "Test_SMB Permissions", the following must be performed in the case of:

1. User group "Everyone": Keep user name and password empty
2. Enter corresponding user name with password (here in this example for "fsc")

- Directory Good parts, Directory Bad parts: Here is a name for the folder in which the data and images to be archived should be stored in the case of a good or bad part. These folders are created below the folder to be described and shared (here: "Test_SMB").
- File name: Enter any file name here for the output file.
- Result files: If the output for the result data is activated, all data specified under "Output / Data Output" are protocolled in a .csv file. A file is created for each evaluation (trigger). The files are numbered consecutively.
- Image files: Archiving of images as .bmp: None, All, Only good parts, Only bad parts
- Storage mode: Limited: Once the maximum number of files is reached, the transferal is terminated. Unlimited: Files are written until the target drive is full. Cyclic: After reaching the maximum number of files, the oldest file is overwritten by the newest.
- Max. number of files: Maximum number of file sets which are allowed to be stored in the target directory.

### 11.4.2.3 Archiving via SMB, Output data

After starting of the sensor, images are archived in the shared directory in the corresponding subfolder, and the data are archived as a .csv file, which has been specified under SensoConfig/Output/Data Output.

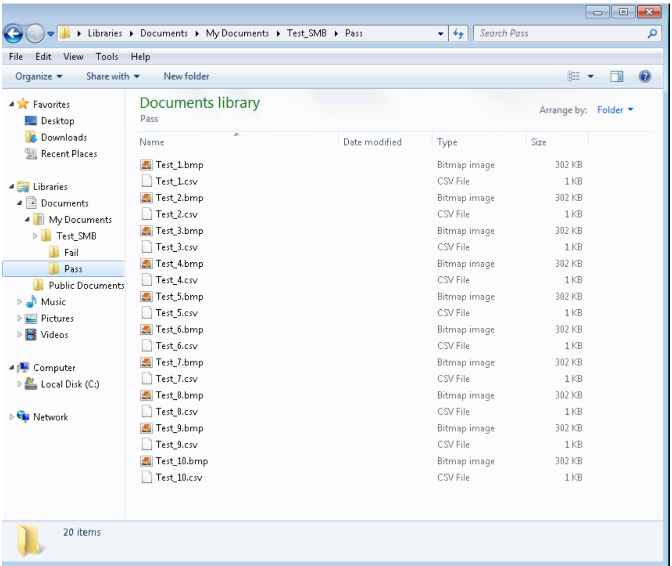



*Fig. 320: Successfully performed archiving via SMB*

## 11.5 SensoRescue

The utility "SensoRescue" is used to reset VISOR® vision sensors that can no longer be found with SensoFind to a state from which they can be addressed and parametrized again by SensoFind and SensoConfig.

1. Start SensoRescue (leave field "Mac address of Sensor" empty)
2. Restart VISOR®, Power off / on or SensoFind/File (VISOR® must be connected to the same network as the PC via Ethernet connection)
3. The "Received Data" field below will show the settings for the VISOR® vision sensor.

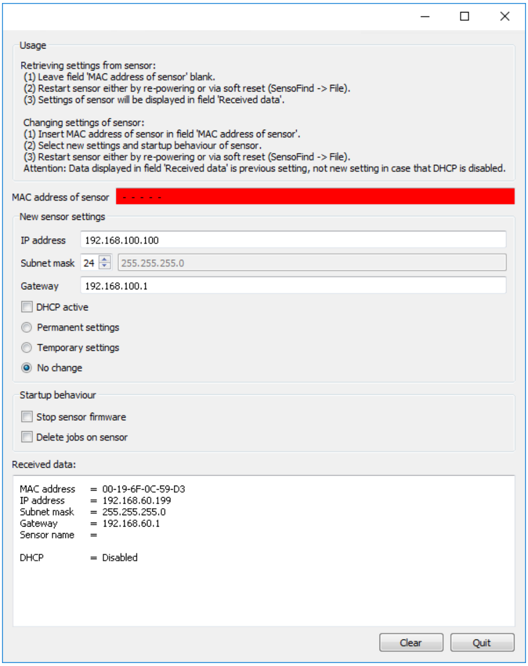



*Fig. 321: SensoRescue /1*

4. Now the Mac address shown below can be entered into the field "Mac address of Sensor"
5. You can enter all the network settings, e.g., IP address, subnet mask, etc. that you want the VISOR® to have after the next restart (power off / on) underneath.
VISOR® neu starten

**NOTE:**
Die im unteren Feld angezeigten Daten werden bei einem Neustart nicht aktualisiert.

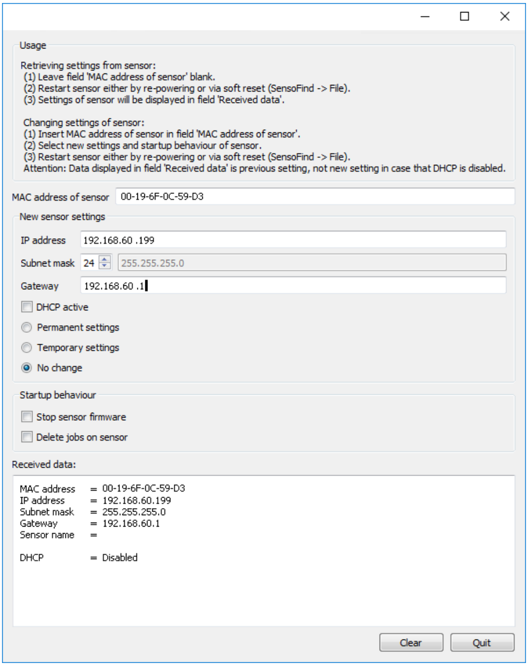



*Fig. 322: SensoRescue / 2*

# 12 Accessories

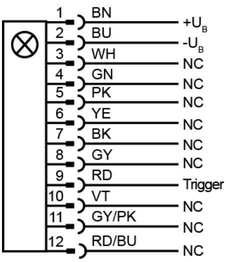
### External illumination

There is an extensive range of accessories available for the VISOR®. It includes a variety of external illuminations, which can be used additionally or instead of the internal illumination.

For more information on vision accessories, please refer to: http://www.-sensopart.com/en/download.

Both types, LF 45 xxxLFR 115 and xxx, can be connected directly to the sensor.

### Connection




### Connection Ring light with sensor

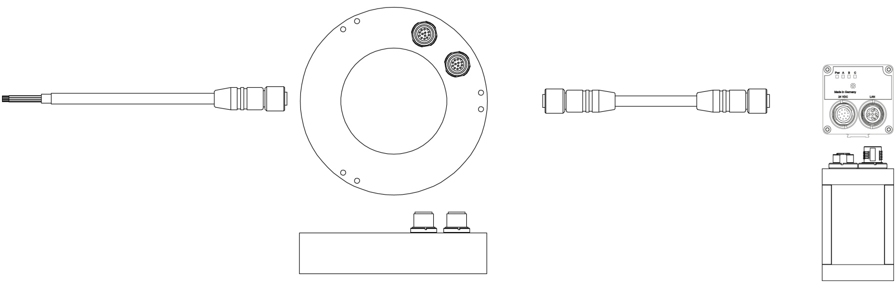
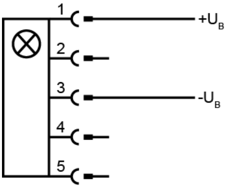


*Fig. 323: Connection of external illumination LF 45LFR 115 xxx and xxx*

All other models are connected to the VISOR® as follows.

## Connection




## Connection Ring light - connection adapter - sensor

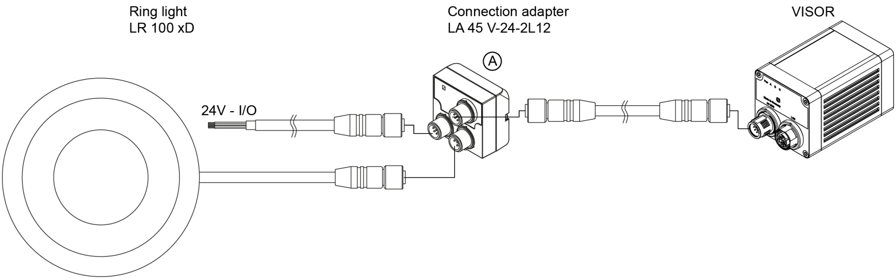



*Fig. 324: Connection of external illumination, all types except LF 45 xxx and LFR 115 xxx*

# 13 Technical data

| Electrical data | |
|---|---|
| Operating Voltage $U_B$ | 24 VDC (18 V – 30 V) |
| Reverse polarity protection | Yes |
| Short circuit protection | Yes |
| Residual ripple | < 5 Vss, level 3<br>EN 61000-4-17 |
| Boot delay | < 13 s |
| Current consumption (without I/O) | ≤ 300 mA |
| Input/output polarity | PNP / NPN |
| Switching threshold for all inputs, incl. encoder | High > $V_B$ - 1 V,<br>Low < 3 V |
| Input resistance | > 20 kOhm |
| Encoder input | 40 kHz |
| Max. output current per output | 50 mA, Ejector (Pin 12 / RDBU) 100 mA |
| Total current (all active outputs) | Max. 200 mA |
| Inductive load | Typical: Relay: 17 K / 2 H (50 mA outputs),<br>Pneumatic valve: 1.4 K / 190 mH (100 mA output) |
| Capacitive load | 900 nF for ejector (pin 12 / RDBU); otherwise 500 nF |
| $t_{rise}$ steepness of switching outputs | If 50 mA or 4.7 kohm pull-up / pull-down<br>PNP: < 300 ns<br>NPN: < 4 us |
| $t_{fall}$ steepness of switching outputs | If 50 mA or 4.7 kohm pull-up / pull-down<br>NPN: < 200 us<br>PNP: < 400 us |
| Interfaces | 100 Mbit LAN, PROFINET, EtherNet/IP, SensoWeb |

| Optical data | V10 / V10C | V20 / V20C | V50 / V50C |
|---|---|---|---|
| Number of pixels | 800 (H) x 600 (V) | 1440 (H) x 1080 (V) | 2560 (H) x 1936 (V) |

| Optical data | V10 / V10C | V20 / V20C | V50 / V50C |
|---|---|---|---|
| Field of view size | 1 / 3.6" | 1 / 2.9" | 1 / 1.8" |
| Pixel size | 4.8 µm x 4.8 µm | 3.45 µm x 3.45 µm | 2,8 µm x 2,8 µm |
| Technology | CMOS Mono / Color | | |
| Light type | Red / white / infrared LED | | |
| Target laser | Red, laser class 1 | | |
| Integrated lens, Focal length | 5.2 (W)<br>9.6 (M)<br>20 (N) | 6.5 (W)<br>12 (M)<br>20 (N) | |
| Focal point (working distance) | Motorized | | |
| Max. number of adjustment cycles per hour | 60 | | |

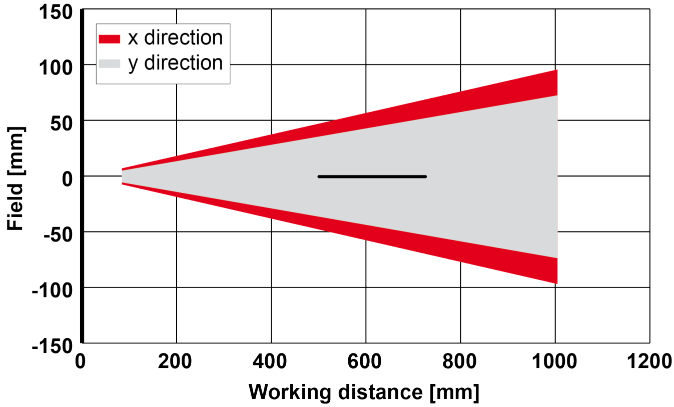
| Mechanical data | |
|---|---|
| Weight | Approx. 200 g |
| Ambient air temperature: operation | 0 °C to 50 °C (80% humidity, non-condensing) 1) |
| Ambient air temperature: storage | -20 °C ... 60 °C (80% humidity, non-condensing) |
| Protection class | IP67<br>EN 60529 |
| Housing material | Die-cast aluminum,<br>RoHS-compliant |
| 1) If you use the spark protection guard, the maximum operating temperature is lowered to 45 °C. | |

| Tests | |
|---|---|
| Vibration resistance | EN 60068-2-6 |
| Shock resistance | EN 60068-2-27 |
| LABS-free | Yes |
| EMC | EN 61000-6-2 / EN 55011 |

| Typical cycle time | |
|---|---|
| Monochrome detectors | Typ. 20 ms Pattern matching<br>Typ. 30 ms Contour<br>Typ. 300 ms Contour 3D<br>Typ. 8 ms Caliper<br>Typ. 30 ms BLOB<br>Typ. 2 ms Brightness<br>Typ. 2 ms Contrast<br>Typ. 2 ms Gray<br>Typ. 30 ms Barcode<br>Typ. 40 ms Datacode<br>Typ. 15 ms pro Zeichen OCR |
| Color Detectors | Typ. 2 ms Color Value<br>Typ. 30 ms Color Area<br>Typ. 2 ms Color List |

# 14 Field of view and depth of field

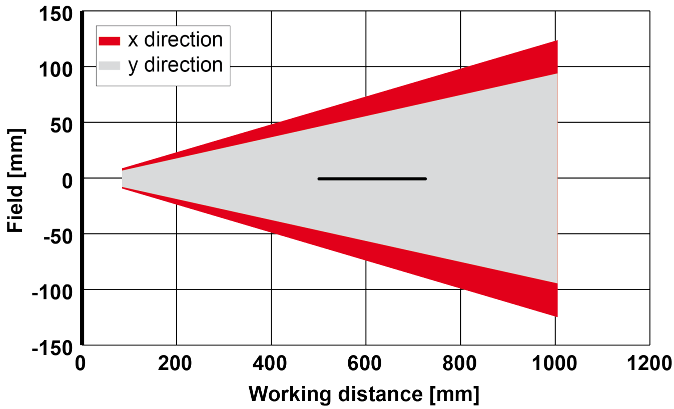
### V10 Narrow objective, internal




- - - - - increased depth of field ——— normal depth of field

*Fig. 325: Field of view and depth of field Narrow objective, internal*

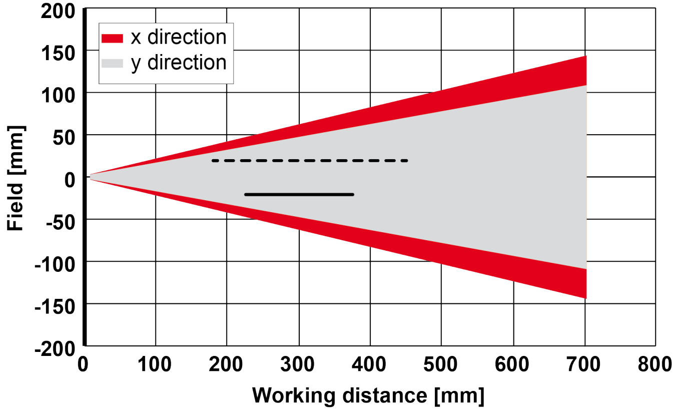
### V20 Narrow objective, internal




- - - - - increased depth of field ——— normal depth of field

*Fig. 326: Field of view and depth of field Narrow objective, internal*

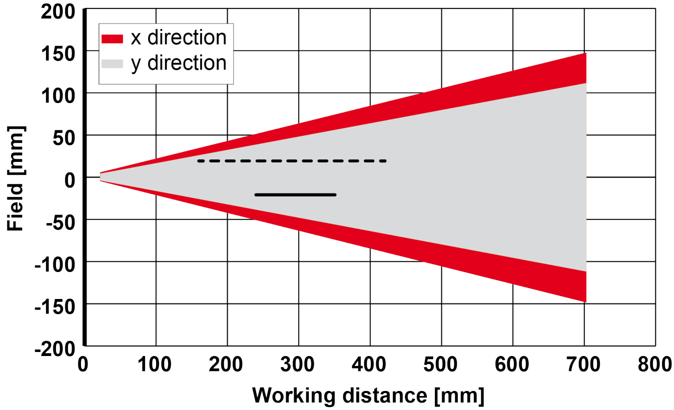
**V10 Medium objective, internal**




- - - - - increased depth of field ——— normal depth of field

*Fig. 327: Field of view and depth of field Medium objective, internal*

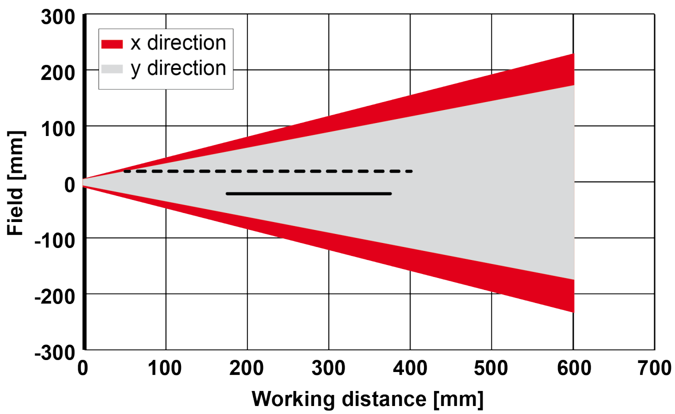
**V20 Medium objective, internal**




- - - - - increased depth of field ——— normal depth of field

*Fig. 328: Field of view and depth of field Medium objective, internal*

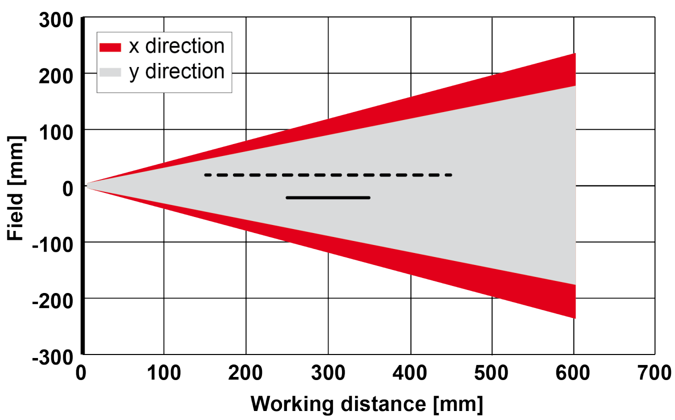
**V10 Wide objective, internal**




- - - - - increased depth of field ——— normal depth of field

*Fig. 329: Field of view and depth of field Wide objective, internal*

**V20 Wide objective, internal**




- - - - - increased depth of field ——— normal depth of field

*Fig. 330: Field of view and depth of field Wide objective, internal*

# 15 Sensor types

## 15.1 Allround

### V10 Allround

| Advanced | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination White | | | |
| 631-91004 | V10-ALL-A3-W-W-M2-L | Wide | normal |
| 631-91005 | V10-ALL-A3-W-M-M2-L | Medium | normal |
| 631-91006 | V10-ALL-A3-W-N-M2-L | Narrow | normal |
| Internal illumination Red | | | |
| 631-91007 | V10-ALL-A3-R-W-M2-L | Wide | normal |
| 631-91008 | V10-ALL-A3-R-M-M2-L | Medium | normal |
| 631-91009 | V10-ALL-A3-R-N-M2-L | Narrow | normal |
| Internal illumination Infrared | | | |
| 631-91010 | V10-ALL-A3-I-W-M2-L | Wide | normal |
| 631-91029 | V10-ALL-A3-I-M-M2-L | Medium | normal |
| 631-91030 | V10-ALL-A3-I-N-M2-L | Narrow | normal |
| C-Mount | | | |
| 631-91003 | V10-ALL-A3-C-2*) | C-Mount | |

### V10C Allround Color

| Advanced | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination White | | | |
| 631-91041 | V10C-ALL-A3-W-W-M2-L | Wide | normal |
| 631-91038 | V10C-ALL-A3-W-M-M2-L | Medium | normal |
| 631-91039 | V10C-ALL-A3-W-N-M2-L | Narrow | normal |
| C-Mount | | | |
| 631-91036 | V10C-ALL-A3-C-2*) | | |

**V20 Allround**

| **Professional** | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination White | | | |
| 632-91012 | V20-ALL-P3-W-W-M2-L | Wide | normal |
| 632-91000 | V20-ALL-P3-W-M-M2-L | Medium | normal |
| 632-91011 | V20-ALL-P3-W-N-M2-L | Narrow | normal |
| Internal illumination Red | | | |
| 632-91010 | V20-ALL-P3-R-W-M2-L | Wide | normal |
| 632-91008 | V20-ALL-P3-R-M-M2-L | Medium | normal |
| 632-91009 | V20-ALL-P3-R-N-M2-L | Narrow | normal |
| Internal illumination Infrared | | | |
| 632-91007 | V20-ALL-P3-I-W-M2-L | Wide | normal |
| 632-91005 | V20-ALL-P3-I-M-M2-L | Medium | normal |
| 632-91006 | V20-ALL-P3-I-N-M2-L | Narrow | normal |
| C-Mount | | | |
| 632-91004 | V20-ALL-P3-C-2*) | C-Mount | |

| **Advanced** | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination White | | | |
| 632-91024 | V20-ALL-A3-W-W-M2-L | Wide | normal |
| 632-91025 | V20-ALL-A3-W-M-M2-L | Medium | normal |
| 632-91026 | V20-ALL-A3-W-N-M2-L | Narrow | normal |
| Internal illumination Red | | | |
| 632-91021 | V20-ALL-A3-R-W-M2-L | Wide | normal |
| 632-91022 | V20-ALL-A3-R-M-M2-L | Medium | normal |
| 632-91023 | V20-ALL-A3-R-N-M2-L | Narrow | normal |
| Internal illumination Infrared | | | |
| 632-91018 | V20-ALL-A3-I-W-M2-L | Wide | normal |
| 632-91019 | V20-ALL-A3-I-M-M2-L | Medium | normal |
| 632-91020 | V20-ALL-A3-I-N-M2-L | Narrow | normal |

| Advanced | | | |
|---|---|---|---|
| C-Mount | | | |
| 632-91017 | V20-ALL-A3-C-2*) | C-Mount | |

**V20C Allround Color**

| Professional | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination White | | | |
| 632-91016 | V20C-ALL-P3-W-W-M2-L | Wide | normal |
| 632-91014 | V20C-ALL-P3-W-M-M2-L | Medium | normal |
| 632-91015 | V20C-ALL-P3-W-N-M2-L | Narrow | normal |
| C-Mount | | | |
| 632-91013 | V20C-ALL-P3-C-2*) | C-Mount | |

| Advanced | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination White | | | |
| 632-91028 | V20C-ALL-A3-W-W-M2-L | Wide | normal |
| 632-91029 | V20C-ALL-A3-W-M-M2-L | Medium | normal |
| 632-91030 | V20C-ALL-A3-W-N-M2-L | Narrow | normal |
| C-Mount | | | |
| 632-91027 | V20C-ALL-A3-C-2*) | C-Mount | |

**V50 Allround**

| Professional | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| C-Mount | | | |
| 635-91006 | V50-ALL-P3-C-2*) | C-Mount | |

**V50C Allround Color**

| Professional | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| C-Mount | | | |
| 635-91009 | V50C-ALL-P3-C-2*) | C-Mount | |

*)

**NOTE:**
For longer operating distances (from approx. 200 mm), external illumination may be necessary.
External IR illumination is only possible with IR types or C-Mount sensors.

## 15.2 Object

### V10 Object

| Advanced | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination White | | | |
| 631-91014 | V10-OB-A3-W-W-M2-L | Wide | normal |
| 631-91015 | V10-OB-A3-W-M-M2-L | Medium | normal |
| 631-91016 | V10-OB-A3-W-N-M2-L | Narrow | normal |
| 631-91017 | V10-OB-A3-W-WD-M2-L | Wide | Enhanced |
| 631-91018 | V10-OB-A3-W-MD-M2-L | Medium | Enhanced |
| Internal illumination Red | | | |
| 631-91019 | V10-OB-A3-R-W-M2-L | Wide | normal |
| 631-91020 | V10-OB-A3-R-M-M2-L | Medium | normal |
| 631-91021 | V10-OB-A3-R-N-M2-L | Narrow | normal |
| 631-91022 | V10-OB-A3-R-WD-M2-L | Wide | Enhanced |
| 631-91023 | V10-OB-A3-R-MD-M2-L | Medium | Enhanced |
| Internal illumination Infrared | | | |
| 631-91024 | V10-OB-A3-I-W-M2-L | Wide | normal |
| 631-91025 | V10-OB-A3-I-M-M2-L | Medium | normal |
| 631-91026 | V10-OB-A3-I-N-M2-L | Narrow | normal |
| 631-91027 | V10-OB-A3-I-WD-M2-L | Wide | Enhanced |
| 631-91012 | V10-OB-A3-I-MD-M2-L | Medium | Enhanced |
| C-Mount | | | |
| 631-91001 | V10-OB-A3-C-2*) | C-Mount | |

| Standard | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination White | | | |
| 631-91043 | V10-OB-S3-W-W-M2 | Wide | normal |
| 631-91044 | V10-OB-S3-W-M-M2 | Medium | normal |
| Internal illumination Red | | | |
| 631-91045 | V10-OB-S3-R-W-M2 | Wide | normal |

| Standard | | | |
|---|---|---|---|
| 631-91046 | V10-OB-S3-R-M-M2 | Medium | normal |
| Internal illumination Infrared | | | |
| 631-91047 | V10-OB-S3-I-W-M2 | Wide | normal |
| 631-91048 | V10-OB-S3-I-M-M2 | Medium | normal |

**V10C Object**

| Advanced | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination White | | | |
| 631-91013 | V10C-OB-A3-W-W-M2-L | Wide | normal |
| 631-91011 | V10C-OB-A3-W-M-M2-L | Medium | normal |
| 631-91002 | V10C-OB-A3-W-N-M2-L | Narrow | normal |
| C-Mount | | | |
| 631-91042 | V10C-OB-A3-C-2*) | C-Mount | |

| Standard | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination White | | | |
| 631-91049 | V10C-OB-S3-W-W-M2 | Wide | normal |
| 631-91050 | V10C-OB-S3-W-M-M2 | Medium | normal |

**V20 Object**

| Advanced | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination White | | | |
| 632-91031 | V20-OB-A3-W-W-M2-L | Wide | normal |
| 632-91032 | V20-OB-A3-W-M-M2-L | Medium | normal |
| 632-91033 | V20-OB-A3-R-W-M2-L | Narrow | normal |
| Internal illumination Red | | | |
| 632-91034 | V20-OB-A3-R-W-M2-L | Wide | normal |
| 632-91035 | V20-OB-A3-R-M-M2-L | Medium | normal |
| 632-91036 | V20-OB-A3-R-N-M2-L | Narrow | normal |

| Advanced | | | |
| --- | --- | --- | --- |
| Internal illumination Infrared | | | |
| 632-91037 | V20-OB-A3-I-W-M2-L | Wide | normal |
| 632-91038 | V20-OB-A3-I-M-M2-L | Medium | normal |
| 632-91039 | V20-OB-A3-I-N-M2-L | Narrow | normal |
| C-Mount | | | |
| 632-91040 | V20-OB-A3-C-2 | C-Mount | |

**V20C Object**

| Advanced | | | |
| --- | --- | --- | --- |
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination White | | | |
| 632-91041 | V20C-OB-A3-W-W-M2-L | Wide | normal |
| 632-91042 | V20C-OB-A3-W-M-M2-L | Medium | normal |
| 632-91043 | V20C-OB-A3-W-N-M2-L | Narrow | normal |
| C-Mount | | | |
| 632-91044 | V20C-OB-A3-C-2 | C-Mount | |

**V50 Object**

| Advanced | | | |
| --- | --- | --- | --- |
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| C-Mount | | | |
| 635-91016 | V50-OB-A3-C-2*) | C-Mount | |

**V50C Object**

| Advanced | | | |
| --- | --- | --- | --- |
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| C-Mount | | | |
| 635-91019 | V50C-OB-A3-C-2*) | C-Mount | |

*)

**NOTE:**
For longer operating distances (from approx. 200 mm), external illumination may be necessary.
External IR illumination is only possible with IR types or C-Mount sensors.

## 15.3 Code reader

### V10 Code reader

| Advanced | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination White | | | |
| 631-91051 | V10-CR-A3-W-W-M2-L | Wide | normal |
| 631-91052 | V10-CR-A3-W-M-M2-L | Medium | normal |
| 631-91053 | V10-CR-A3-W-N-M2-L | Narrow | normal |
| 631-91035 | V10-CR-A3-W-WD-M2-L | Wide | Enhanced |
| 631-91034 | V10-CR-A3-W-MD-M2-L | Medium | Enhanced |
| Internal illumination Red | | | |
| 631-91054 | V10-CR-A3-R-W-M2-L | Wide | normal |
| 631-91055 | V10-CR-A3-R-M-M2-L | Medium | normal |
| 631-91056 | V10-CR-A3-R-N-M2-L | Narrow | normal |
| 631-91033 | V10-CR-A3-R-WD-M2-L | Wide | Enhanced |
| 631-91032 | V10-CR-A3-R-MD-M2-L | Medium | Enhanced |
| Internal illumination Infrared | | | |
| 631-91057 | V10-CR-A3-I-W-M2-L | Wide | normal |
| 631-91058 | V10-CR-A3-I-M-M2-L | Medium | normal |
| 631-91059 | V10-CR-A3-I-N-M2-L | Narrow | normal |
| 631-91031 | V10-CR-A3-I-WD-M2-L | Wide | Enhanced |
| 631-91028 | V10-CR-A3-I-MD-M2-L | Medium | Enhanced |
| C-Mount | | | |
| 631-91060 | V10-CR-A3-C-2*) | C-Mount | |

| Standard | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination White | | | |
| 631-91061 | V10-CR-S3-W-W-M2 | Wide | normal |
| 631-91062 | V10-CR-S3-W-M-M2 | Medium | normal |
| 631-91063 | V10-CR-S3-W-N-M2 | Narrow | normal |

| Standard | | | |
|---|---|---|---|
| 631-91000 | V10-CR-S3-W-WD-M2 | Wide | Enhanced |
| 631-91037 | V10-CR-S3-W-MD-M2 | Medium | Enhanced |
| Internal illumination Red | | | |
| 631-91064 | V10-CR-S3-R-W-M2 | Wide | normal |
| 631-91065 | V10-CR-S3-R-M-M2 | Medium | normal |
| 631-91068 | V10-CR-S3-R-N-M2 | Narrow | normal |
| 631-91066 | V10-CR-S3-R-WD-M2 | Wide | Enhanced |
| 631-91067 | V10-CR-S3-R-MD-M2 | Medium | Enhanced |
| Internal illumination Infrared | | | |
| 631-91069 | V10-CR-S3-I-W-M2 | Wide | normal |
| 631-91070 | V10-CR-S3-I-M-M2 | Medium | normal |
| 631-91072 | V10-CR-S3-I-N-M2 | Narrow | normal |
| 631-91040 | V10-CR-S3-I-WD-M2 | Wide | Enhanced |
| 631-91071 | V10-CR-S3-I-MD-M2 | Medium | Enhanced |

**V20 Code reader**

| Advanced | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination White | | | |
| 632-91045 | V20-CR-A3-W-W-M2-L | Wide | normal |
| 632-91046 | V20-CR-A3-W-M-M2-L | Medium | normal |
| 632-91047 | V20-CR-A3-W-N-M2-L | Narrow | normal |
| Internal illumination Red | | | |
| 632-91048 | V20-CR-A3-R-W-M2-L | Wide | normal |
| 632-91049 | V20-CR-A3-R-M-M2-L | Medium | normal |
| 632-91050 | V20-CR-A3-R-N-M2-L | Narrow | normal |
| 632-91078 | V20-CR-A3-R-MD-M2-L | Medium | Enhanced |
| Internal illumination Infrared | | | |
| 632-91051 | V20-CR-A3-I-W-M2-L | Wide | normal |
| 632-91052 | V20-CR-A3-I-M-M2-L | Medium | normal |
| 632-91053 | V20-CR-A3-I-N-M2-L | Narrow | normal |

| Advanced | | | |
|---|---|---|---|
| C-Mount | | | |
| 632-91054 | V20-CR-A3-C-2*) | C-Mount | |

| Standard | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination White | | | |
| 632-91055 | V20-CR-S3-W-W-M2-L | Wide | normal |
| 632-91056 | V20-CR-S3-W-M-M2-L | Medium | normal |
| 632-91057 | V20-CR-S3-W-N-M2-L | Narrow | normal |
| Internal illumination Red | | | |
| 632-91058 | V20-CR-S3-R-W-M2-L | Wide | normal |
| 632-91059 | V20-CR-S3-R-M-M2-L | Medium | normal |
| 632-91060 | V20-CR-S3-R-N-M2-L | Narrow | normal |
| Internal illumination Infrared | | | |
| 632-91061 | V20-CR-S3-I-W-M2-L | Wide | normal |
| 632-91062 | V20-CR-S3-I-M-M2-L | Medium | normal |
| 632-91063 | V20-CR-S3-I-N-M2-L | Narrow | normal |

**V50 Code reader**

| Advanced | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| C-Mount | | | |
| 635-91033 | V50-CR-A3-C-2*) | C-Mount | |

| Professional | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| C-Mount | | | |
| 635-91026 | V50-CR-P3-C-2*) | C-Mount | |

*)

**NOTE:**
For longer operating distances (from approx. 200 mm), external illumination may be necessary.
External IR illumination is only possible with IR types or C-Mount sensors.

## 15.4 Robotic

### V10 Robotic

| Advanced | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination WhiteInternal illumination | | | |
| 631-91073 | V10-RO-A3-W-W-M2-L | Wide | normal |
| 631-91074 | V10-RO-A3-W-M-M2-L | Medium | normal |
| 631-91075 | V10-RO-A3-W-N-M2-L | Narrow | normal |
| Internal illumination RedInternal illumination | | | |
| 631-91076 | V10-RO-A3-R-W-M2-L | Wide | normal |
| 631-91077 | V10-RO-A3-R-M-M2-L | Medium | normal |
| 631-91078 | V10-RO-A3-R-N-M2-L | Narrow | normal |
| Internal illumination InfraredInternal illumination | | | |
| 631-91079 | V10-RO-A3-I-W-M2-L | Wide | normal |
| 631-91080 | V10-RO-A3-I-M-M2-L | Medium | normal |
| 631-91081 | V10-RO-A3-I-N-M2-L | Narrow | normal |
| C-Mount | | | |
| 631-91082 | V10-RO-A3-C-2*) | C-Mount | |

### V20 Robotic

| Advanced | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination WhiteInternal illumination | | | |
| 632-91064 | V20-RO-A3-W-W-M2-L | Wide | normal |
| 632-91065 | V20-RO-A3-W-M-M2-L | Medium | normal |
| 632-91066 | V20-RO-A3-W-N-M2-L | Narrow | normal |
| Internal illumination RedInternal illumination | | | |
| 632-91067 | V20-RO-A3-R-W-M2-L | Wide | normal |
| 632-91068 | V20-RO-A3-R-M-M2-L | Medium | normal |
| 632-91069 | V20-RO-A3-R-N-M2-L | Narrow | normal |

| Advanced | | | |
|---|---|---|---|
| Internal illumination InfraredInternal illumination | | | |
| 632-91070 | V20-RO-A3-I-W-M2-L | Wide | normal |
| 632-91071 | V20-RO-A3-I-M-M2-L | Medium | normal |
| 632-91072 | V20-RO-A3-I-N-M2-L | Narrow | normal |
| C-Mount | | | |
| 632-91073 | V20-RO-A3-C-2*) | C-Mount | |

**V20C Robotic**

| Advanced | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination WhiteInternal illumination | | | |
| 632-91074 | V20C-RO-A3-W-W-M2-L | Wide | normal |
| 632-91075 | V20C-RO-A3-W-M-M2-L | Medium | normal |
| 632-91076 | V20C-RO-A3-W-N-M2-L | Narrow | normal |
| C-Mount | | | |
| 632-91077 | V20C-RO-A3-C-2*) | C-Mount | |

**V50 Robotic**

| Professional | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| C-Mount | | | |
| 635-91040 | V50-RO-P3-C-2*) | C-Mount | |

**V50C Robotic**

| Professional | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| C-Mount | | | |
| 635-91043 | V50C-RO-P3-C-2*) | C-Mount | |

*)

**NOTE:**
For longer operating distances (from approx. 200 mm), external illumination may be necessary.
External IR illumination is only possible with IR types or C-Mount sensors.

## 15.5 Solar

### V20 Solar

| Advanced | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| Internal illumination Red | | | |
| 632-91157 | V20-ALL-A3-R-W-M2-L | Wide | Normal |

### V50 Solar

| Professional | | | |
|---|---|---|---|
| **Part no.** | **Type designation** | **Focal length** | **Depth of focus** |
| C-Mount | | | |
| 635-91044 | V50-SO-A3-C-2*) | C-Mount | |

*)

**NOTE:**
For longer operating distances (from approx. 200 mm), external illumination may be necessary.
External IR illumination is only possible with IR types or C-Mount sensors.

# 16 Maintenance

## 16.1 Maintenance

The following maintenance work must be performed for the vision sensor at regular intervals:

- Clean the vision sensor
- Check all connectors and fittings

## 16.2 Cleaning

The vision sensor's housing must be cleaned with a clean, dry cloth.

If the sensor's front is soiled, it must be cleaned with a soft cloth and a bit of plastic cleaner if necessary

**ATTENTION:**
Please note that improperly cleaning the front can damage it:

- Never use aggressive detergents such as solvents or benzine.
- Do not use any sharp objects; do not scratch the front.

## 16.3 Repairs

The vision sensor should be repaired exclusively by the manufacturer. The manufacturer's warranty will be void if you open, alter, or otherwise modify the product.

# 17 Disposal

This device must be disposed of in accordance with all applicable national environmental regulations and waste disposal regulations.
Since it is e-waste, it is strictly prohibited to dispose of it in household waste.

# We look ahead
Yesterday, today and in the future

SENSOPART

Germany
SensoPart Industriesensorik GmbH
79288 Gottenheim
Tel.: +49 7665 94769-0
info@sensopart.de

France
SensoPart France SARL
F-77420 Champs sur Marne
Tel.: +33 1 64 73 00 61
info@sensopart.fr

Great Britain
SensoPart UK Ltd.
Melton Mowbray, Leicestershire, LE13 0PB
Tel.: +44 1664 561539
uk@sensopart.com

USA
SensoPart Inc.
Perrysburg, OH 43551
Tel.: +1 866 282 7610
usa@sensopart.com

China
SensoPart (Shanghai)
Shanghai, 201803
Tel.: +86 216 901 7660
china@sensopart.cn



EN
068-14797-02

SENSOPART

# VISOR®

## Communications manual

Software version 2.2

## Copyright (English)

No part of this document may be reproduced, published, or stored in databases or information retrieval systems in any form – even in part – nor may illustrations, drawings, or the layout be copied without prior written permission from SensoPart Industriesensorik GmbH.

We accept no responsibility for printing errors or mistakes which occurred in drafting these document. Subject to delivery and technical alterations.

First publication 01 / 2019

SensoPart Industriesensorik GmbH
Nägelseestr. 16
79288 Gottenheim

c UL US LISTED
IND. CONT. EQ.
77HA
For use in class 2 circuits

For use in NFPA 79 Applications only

CE

## Table of contents

# 1 Information on this document

## 1.1 Explanation of symbols

### Warnings

**CAUTION / WARNING / DANGER**
This symbol is used to indicate a potentially hazardous situation that, if not avoided, could result in death or serious injury.

**WARNING**
This symbol is used to indicate potentially hazardous situations arising from laser beams.

**ATTENTION:**
This symbol is used to indicate text that must be observed without fail. Failure to do so may result in bodily injury or property damage.

**NOTE:**
This symbol is used to highlight useful tips and recommendations, as well as information intended to help ensure efficient operation.

### Detectors

- Pattern matching
- Contour
- Contrast
- Brightness
- Gray
- Caliper
- BLOB
- Contour 3D
- Barcode
- Datacode
- OCR
- Color Value
- Color List
- Color Area
- Result processing

### Alignment

Alignment
Includes the position detectors: Contour matching, Pattern matching, and Edge detector

## 1.2 Additional documents

The following documents for the VISOR® vision sensor are available for download in the Download area of the SensoPart website.

- VISOR® User Manual
- VISOR® Communications manual
- VISOR® Operating manual

Furthermore, these documents are part of the software installation and can be found in the subfolder "\Documentation\", as well as via the Windows Start menu.

## 1.3 Document version

This manual describes the VISOR® software version 2.2.

Documents for the previous software versions (< 2.2) can be found in the download area of the SensoPart homepage (www.sensopart.com).

# 2 Network connection

## 2.1 Integrating the VISOR® into the network / gateway

SensoFind/Active sensors will show a list with all the VISOR® vision sensors that are found on the same network segment on the PC on which is running SensoFind. To update the list, press the "Find" button, e.g. for sensors that were only activated after viewing SensoFind.

For sensors which are installed in the network but are located in a different network segment via a gateway, please enter the corresponding sensor IP address under "Add active sensor" and press the button "Add". The corresponding sensor will now also appear in the "Active sensors" list, and you will be able to access it and work with it.

## 2.2 Network connection: Direct connection

Establishing a direct Ethernet connection between the VISOR® vision sensor and the PC

**Important:**
To configure the PC via **direct connection**, the PC's IP configuration must be changed. Inform the system administrator of this change in advance!

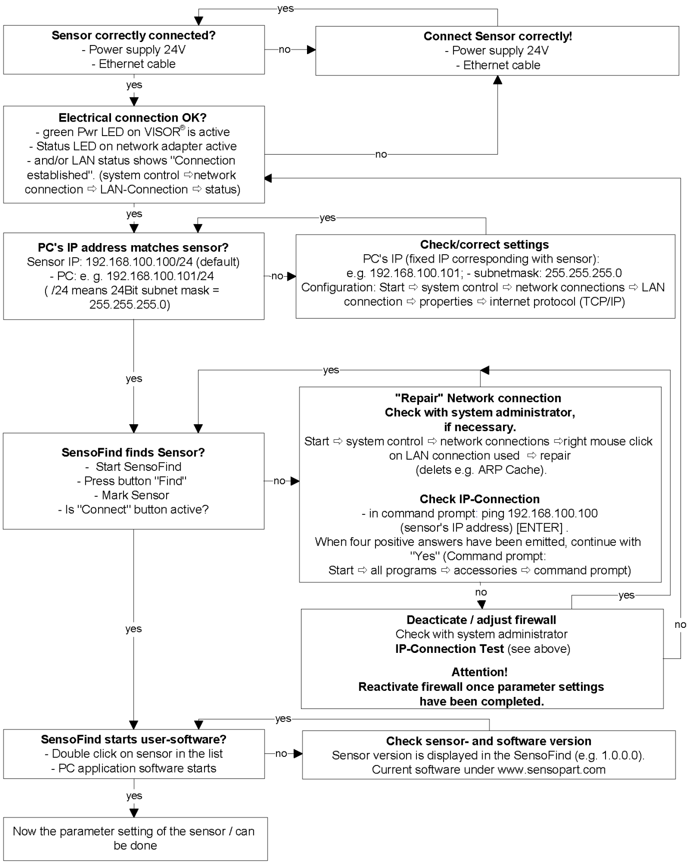



*Fig. 1: Direct connection sensor / PC, procedure and troubleshooting*

## 2.3 Network connection: Connection via network

Establishing an Ethernet connection between the VISOR® vision sensor and the PC through a network.

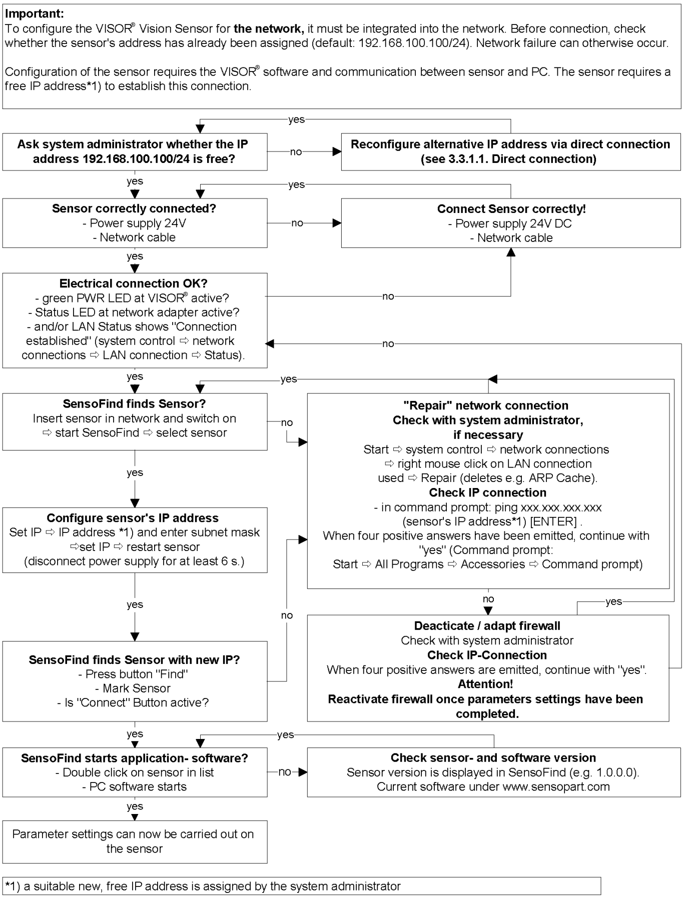



*Fig. 2: Connection via network sensor / PC, procedure and troubleshooting*

## 2.4 Used Ethernet ports

If you are integrating the VISOR® into a network, make sure that an admin opens the following ports if necessary. This is only the case if these ports were previously explicitly blocked in the company network or by a firewall installed on the PC.

The following ports are used for communications between the VISOR® software (PC) and the VISOR®:

- Port 2000, TCP
- Port 2001, UDP Broadcast (to find sensors via SensoFind)
- Port 2002, TCP
- Port 2003, TCP
- Port 2004, TCP

The following ports are used for communications between the PLC (PLC or control PC) and VISOR® vision sensor:

Process interfaces:

- Ethernet
  - Port 2005, TCP (Implicit results, i.e. user-configured result data)
  - Port 2006, TCP (Explicit requests, e.g. trigger or job switch)
- EtherNet/IP:
  - Port 2222, UDP
  - Port 44818, TCP
- PROFINET:
  - Port 161, UDP
  - Port 34962, UDP
  - Port 34963, UDP
  - Port 34964, UDP
- Service:
  - Port 22, TCP
  - Port 1998, TCP
- SensoWeb:
  - Port 80

**NOTE:**
If Ports 2005 or 2006 are changed in the configuration software, they must also be changed accordingly in the firewall by an administrator.

## 2.5 Access to VISOR® through network

Exemplary values for IP, etc.

Access to VISOR® 1 from PC 1, if on the same subnet

- Via SensoFind (/find)

Access to VISOR® 2 from PC 1, if on a different subnet

Only if:

- Gateway is set correctly in Sensor 2 (here to 192.168.30.1) - and
- in SensoFind via Add IP, the sensor IP of Sensor 2 is set correctly
  > after this, VISOR® 2 will also appear in the "Active sensors" list in SensoFind!

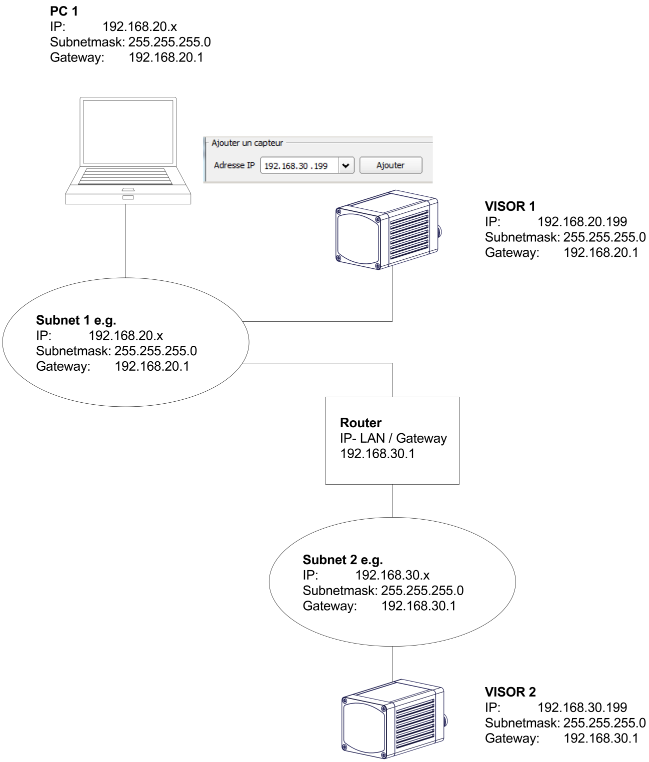



*Fig. 3: Access to VISOR® through network, same or other subnet*

## 2.6 Access to VISOR® through the Internet / World Wide Web

Exemplary values for IP, etc.

Access from PC 1 (company network 1), through the Word Wide Web, to company network 2 to VISOR® 1.

1. On PC 1 (company network 1SensoFind) enter and add the IP WAN of Router 2 (company network 2) under "Add active sensor" in (here in this example: 62.75.148.101)
2. On router 2, open the ports that the sensor will be using (please refer to section: Used Ethernet ports).See Chapter:

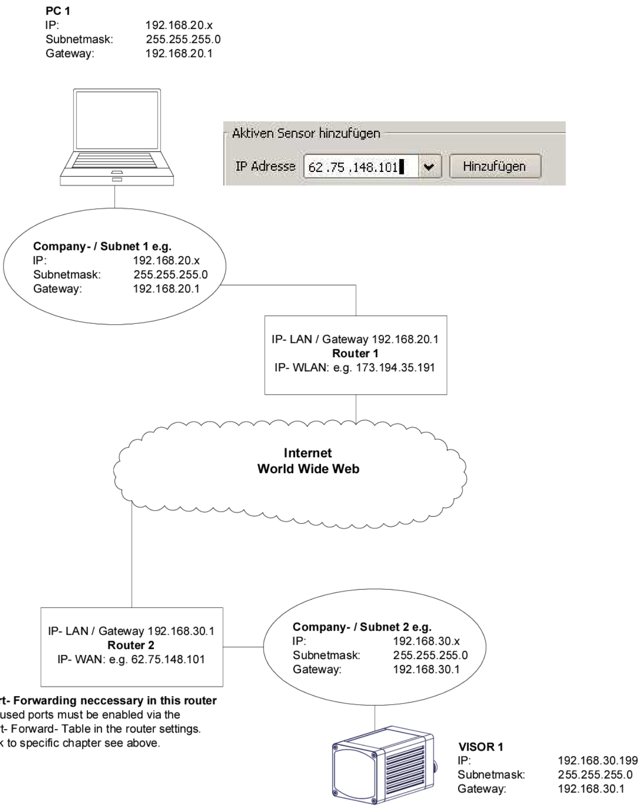



*Fig. 4: Access to VISOR® through the Internet / World Wide Web*

## 2.7 Electrical connection of VISOR® in the network

The VISOR® vision sensor is connected to the network through a switch.

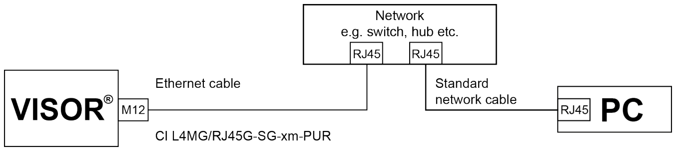


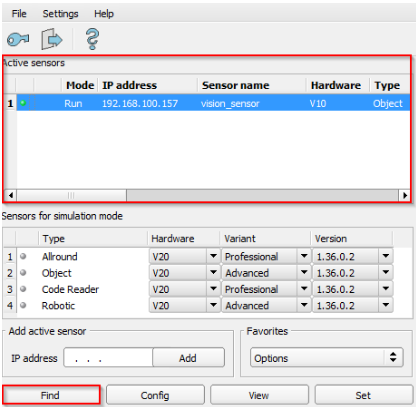

*Fig. 5: Electrical connection of VISOR® in the network*

# 3 Configuration VISOR® vision sensor

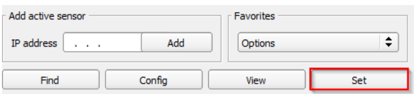
In order to configure the vision sensor, follow the steps below.

**Settings in SensoFind**

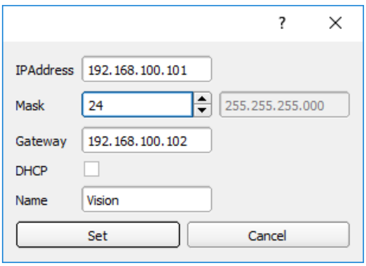

1. Start the VISOR® software. SensoFind is opened.



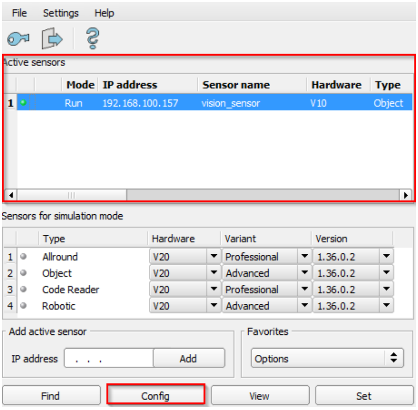

2. Click on the "Find" button. The vision sensor will be listed in the "Active sensors" window.



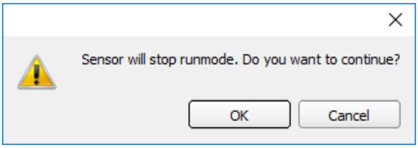

3. Click on the "Set" button. The dialog box for configuring the IP address and the sensor name will appear.




4. Assign an IP address and a name to the sensor.
5. Click on the "Set" button. The IP address and the name have now been updated.

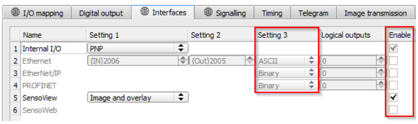



6. Open SensoConfig by selecting the sensor you want and then clicking on the "Config" button.



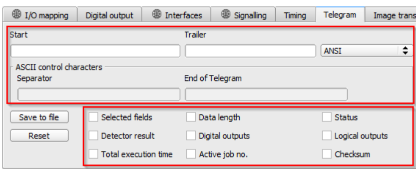

7. Confirm the following dialog box with "OK" to stop SensoFind and start the configuration in SensoConfig.

## Select an interface in SensoConfig

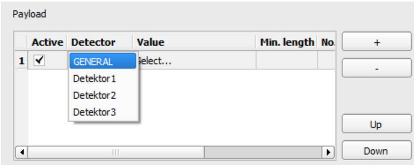



1. Use the "Output" setup step to open the "Interfaces" tab.
2. Enable the interface by enabling the corresponding checkbox in the "Enabled" column.
3. In the "Setting 3" column, select the format for the data output.

## Defining telegrams / data output in SensoConfig



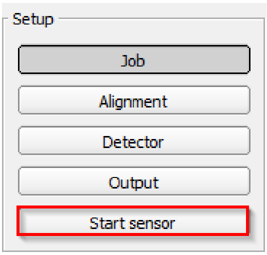

1. Use the "Output" setup step to open the "Telegram" tab.
2. Set the control characters you want for the data output.
3. Select the Checkboxes you want.




4. Configure the data you want to be output. Use the "+" button to generate new entry. What the buttons do:
   - "+": Insert new entry
   - "-": Delete marked entry
   - "Up", "Down": Displace marked entry
5. Select the detector you want in the "Detector" column.
6. Select the detector value you want in the "Value" column so that this value will be output through the enabled interface.

Additional information:: Data output (ASCII / binary)

**Start sensor**




1. Click on the "Start sensor" setup step. The data will be transferred to the vision sensor and the vision sensor will be started.

**NOTE:**
Detector must be generated.

# 4 Ethernet TCP/IP, port 2005 / 2006

Numerical data, which has been configured under Output/Telegram, can be output in a separate ASCII/BINARY format.

The sensor here is the (socket) "server", and provides the data via a "server socket" interface. This is mainly a "programming interface".

To read / process the data, a "socket client" (PC, PLC, etc.) must establish a (socket) connection (active) to the sensor, and then receives the data.

**Handling, Settings**

## 4.1 Example: Data output from VISOR® to PC / PLC

**Step 1:**

After the job with all necessary detectors, Alignment, etc. is set, the Ethernet interface for data output is activated and, if necessary, parameterized.

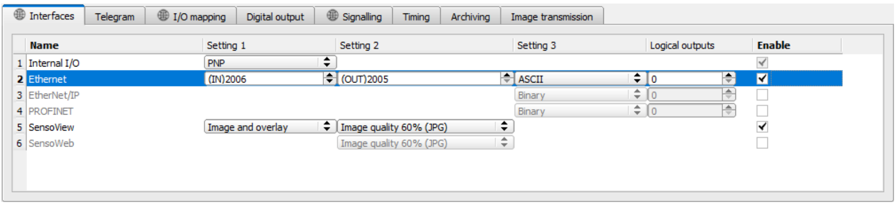



*Fig. 6: Data output, Ethernet*

In the example, the Ethernet interface is activated in the parameter field in the tab "Interfaces" by marking the checkbox "Enable". The default settings for input port (IN) = 2006 and output port (OUT) = 2005 are adopted in this way. Any other settings can be made here to adapt the data output to your network environment. If necessary, contact your network administrator.

**Step 2:**

The "Telegram" tab configures the payload to be output via Ethernet Port 2005.

In this example, it is the:

- Start "010"
- Overall result of Detector 1
- Trailer "xxx"

"ASCII" is defined as a data format, which facilitates the traceability of this example. The function with other data or in binary is analogous to settings made here by way of example.

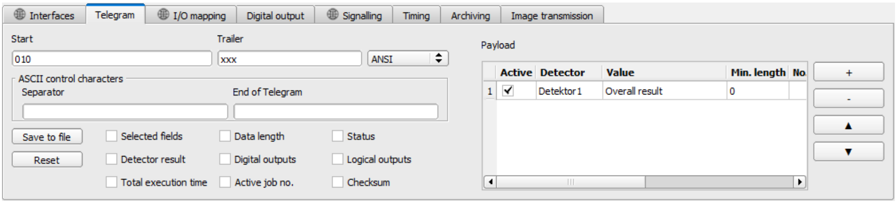



*Fig. 7: Telegram, configure output data*

**Step 3:**

After opening the Hercules Ethernet tool, you will need to open the "TCP-Client" tab to communicate with the VISOR® socket server via Ethernet.

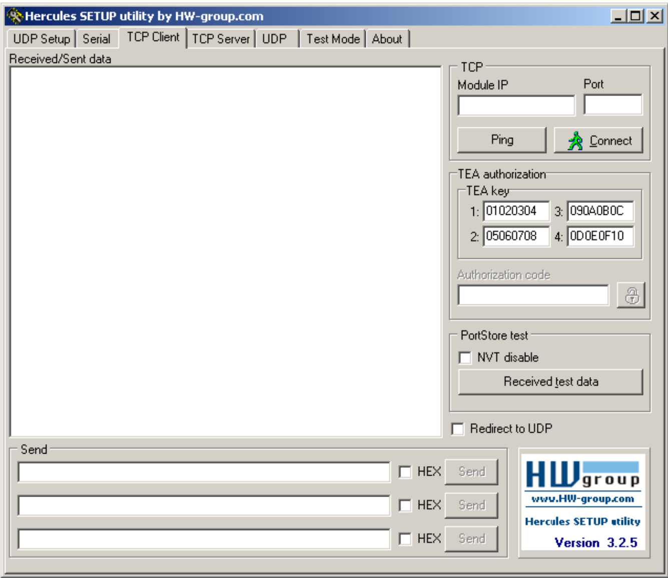



*Fig. 8: Data output, Ethernet Tool / 1*

You will need to enter the IP address of the VISOR® and the correct port in order to receive data.

The IP address of the VISOR® can be found in SensoFind. See the first line in the window "Active sensors" = 192.168.60.199

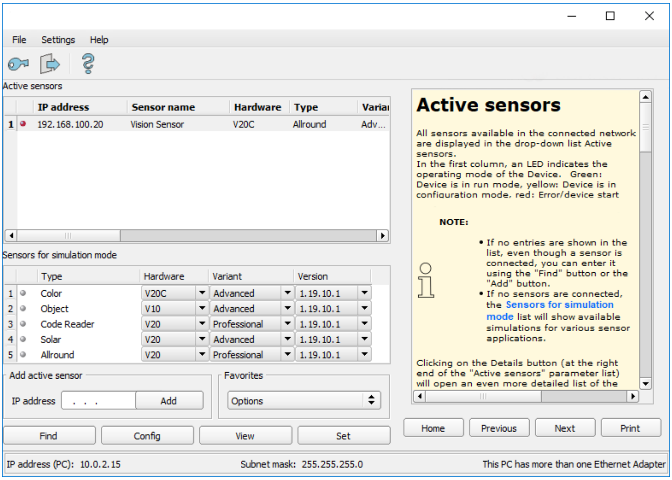



*Fig. 9: SensoFind, IP address ...*

The port number for the output port was adopted under Step 1 with Port 2005.

**Step 4:**

Therefore, the following settings are made in Hercules: Module IP = 192.168.60.199, Port = 2005. All other settings remain in the default values. Clicking on the "Connect" button will establish a connection to the VISOR® and the connection will be shown in green letters in the main window.

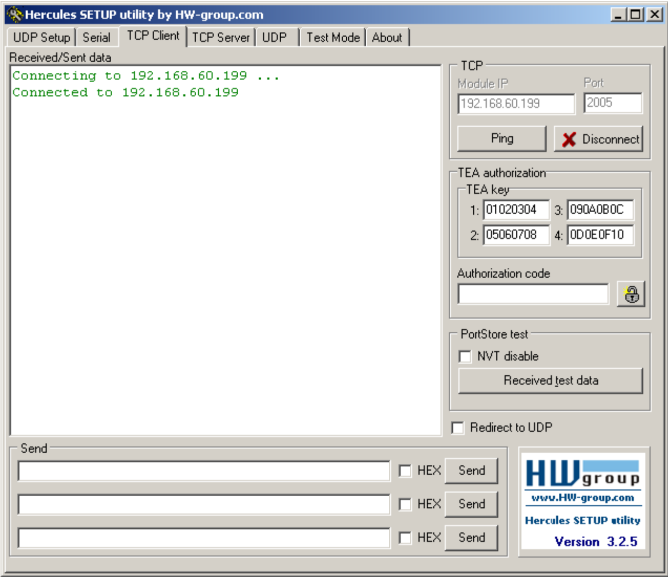



*Fig. 10: Data output, Ethernet Tool / 2*

**Step 5:**

You will now need to start the VISOR® from the PC application with "Start sensor" (later during operation, the VISOR® will run normally after being turned on and will transmit data if configured). In this example, Trigger mode = continuous is set, i.e. evaluations are made continuously and data is sent. These are only visible in the main window of Hercules.

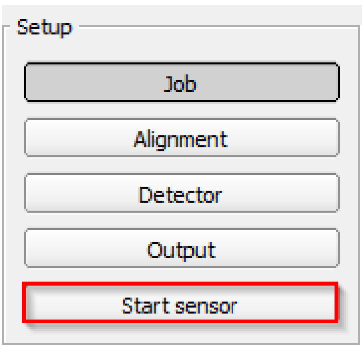



*Fig. 11: Start sensor*

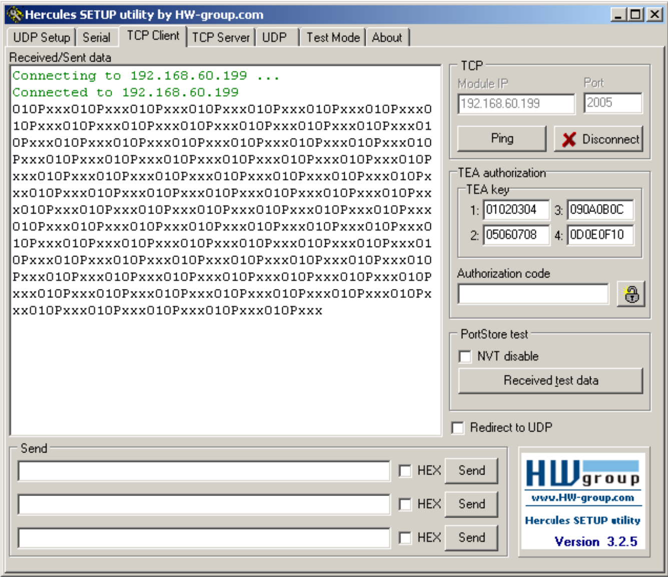



*Fig. 12: Data output, Ethernet, Tool / 3*

The data visible here are set under "Telegram":

- Start "010"
- Overall result of Detector 1 (here, a "P" for positive, since test condition: brightness fulfilled)
- Trailer "xxx"

## 4.2 Example: Commands (requests) from PC / PLC VISOR®

### With acknowledgement / data output from VISOR®

### Step 1

For better clarity, the triggered operation is switched to here for Example 2. This can be done as follows: In SensoConfig under Job/Image Acquisition/Trigger mode = Set "Trigger". All other settings from Ethernet example 1 in the VISOR® remain unchanged.

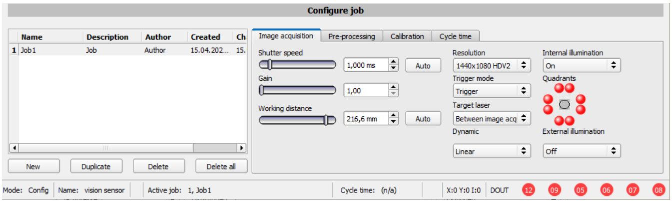



*Fig. 13: Data output, Ethernet, Trigger*

**Step 2**

In order to transmit commands to the VISOR®, the Hercules application needs to be opened again. This time with port 2006 as the VISOR® input port through which it can receive commands. All telegrams (commands and response strings) to and from the VISOR® are described in section Overview telegrams (Page 87).

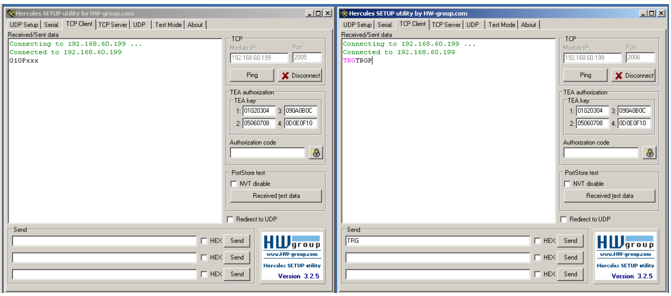



*Fig. 14: Data output, Ethernet Tool / 4*

In the right window, the "TRG" command (for Trigger; see first line on the bottom for command) was sent from port 2006 to the VISOR® by clicking on the corresponding "Send" button. This command is shown in the main window in red letters when being sent. The VISOR® responds to port 2006 with an acknowledge to the "TRG" command and, in this case, "P" for a positive detector 1 result (black letters in right pane).

In the left window, the VISOR® uses output port 2005 to send the "010Pxxx" value defined in Data output the same way as in the Ethernet 1 example.

## 4.3 Example: Job change from PC / PLC to VISOR®

With acknowledgement / data output from VISOR®

**Function of both Ethernet ports for in- and output:**

*A: Port 2005, only one direction: Sensor >> PC, all payload, defined under "Data output"

*B: Port 2006, both directions: Sensor <> PC, commands to VISOR® with acknowledge, + all response data to commands (no payloads)

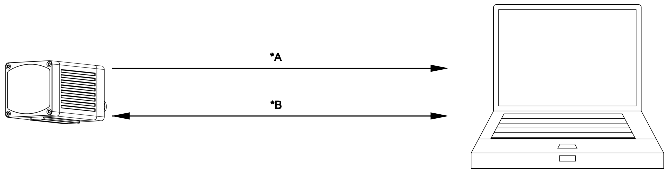



*Fig. 15: Ethernet ports*

**Step 1**

For better clarity, it is switched here to the triggered mode. This can be done as follows: In SensoConfig under Job/Image Acquisition/Trigger mode = Set "Trigger". All other settings from Ethernet example 1 in the VISOR® remain unchanged. All data output definitions are made here in "ASCII" for better traceability of the examples.

For this example, at least two jobs must be created on the VISOR® vision sensor. To create a new job based on an existing job, you can use the "Duplicate" function. Adjust the following parameters to easily check the job change. Later you can freely define the output.

For this example, Job 1 was defined with the data output:

- Start: "010" and
- Trailer: "xxx"

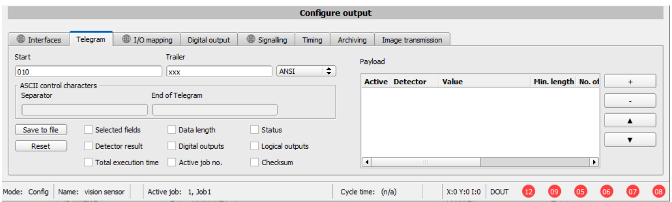



*Fig. 16: Data output, Ethernet, Job switch Job 1*

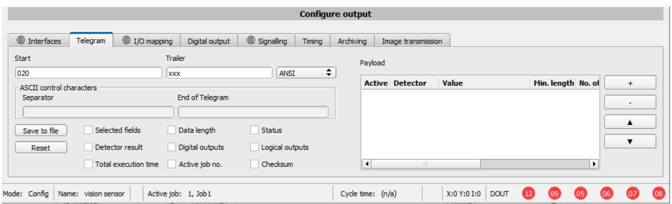



*Fig. 17: Data output, Ethernet, Job switch, Job 2*

**Step 2**

Here, the application Hercules was opened twice. Once with port 2005 (receiving of results as defined in "Data output") and port 2006 (commands + acknowledge) as VISOR® input port through which it can receive commands.

All telegrams (commands and response strings) to and from the VISOR® are described in section Overview telegrams (Page 87).

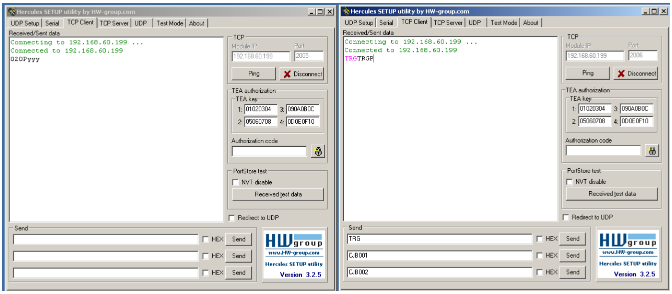



*Fig. 18: Data output, Ethernet, Job switch, Tool / 1*

In the right window (Port 2006), the command TRG (Trigger, see "Send" below, first line) was issued. This is displayed in the main window in red letters with "TRG". The VISOR® responds immediately with the "TRGP" acknowledge (repetition of "TRG" command and "P" for positive, in black letters in the right pane)

In the left window (Port2005), the VISOR® on which Job 2 is currently active sends the corresponding result string, which is defined in Data output in Job 2 with "020Pyyy".

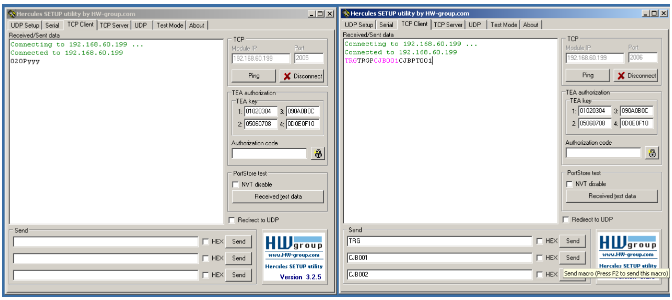



*Fig. 19: Data output, Ethernet, Job switch, Tool / 2*

Nun wurde im rechten Fenster (Port2006) das Kommando CJB001 (Change Job 001, 001 = Job Nr. 1, siehe unten bei „Send“, zweite Zeile) abgesetzt. This is displayed in the main window in red letters with "CJB001". The VISOR® responds immediately with the "CJBPT001" acknowledge (repetition of "CJB" command, "P" for positive, "T" = Triggered, 001 job number to which the change was made)

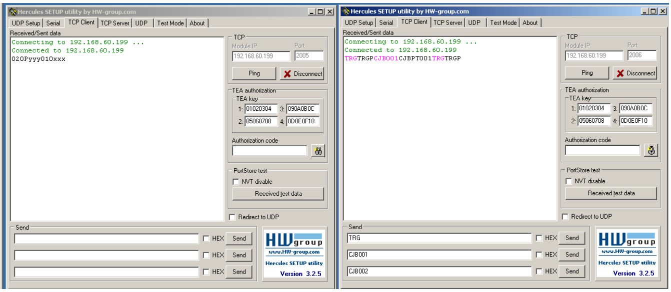



*Fig. 20: Data output, Ethernet, Job switch, Tool / 3*

After the next Trigger command TRG (see "Send" below, third line) is displayed again in the main window in red letters. The VISOR® immediately responds again with the "TRGP" acknowledge (repetition of "TRG" command and "P" for positive)

In the left window (Port2005), after the job has changed to Job 1, the VISOR® sends the corresponding result string, which is defined in Data output in Job 1 with "010xxx"!

## 4.4 Example Beckhoff CX 1020

The connection to a Beckhoff CX 1020 and the corresponding configuration is described in the Beckhoff Operating instructions in:

Start menu/SensoPart/VISOR vision sensor/Tools/SPS PLC/...

## 4.5 Example Siemens S7

The connection to a Siemens S7 PLC and the corresponding configuration is described in the Siemens S7 Operating instructions in:

Start menu/SensoPart/VISOR vision sensor/Tools/SPS PLC/...

# 5 Service / Visualization

There is a service port (Ethernet TCP/IP port 1998) available for the VISOR® vision sensor. This port will be available regardless of how you configure the various steps.

## 5.1 Backup creation

The following telegrams can be used for automatic backups and restores

- **Read job set** (ASCII)
  The "Set job set" telegram can be used to change the VISOR® vision sensor's job set. The job set file must first be loaded onto the VISOR®.
- **Save job set** (ASCII)
  The "Save job set" telegram can be used to read the VISOR® vision sensor's job set.

## 5.2 Visualization

The VISOR® vision sensor provides all data for the visualization of the applications via the service port.

Additional information: Service (available only on port 1998 and in ASCII format) (Page 161)

# 6 VISOR® telegrams for PROFINET and EtherNet/IP

## 6.1 Module 1: "Control" (From PLC to VISOR®)

Name in PLC "CTRL (3 bytes)"

| Byte Offset | Bit Adr.. | Name | Data type | Meaning |
|---|---|---|---|---|
| 0 | 0 | Reset error | 1 bit | Reset Error clears the 4 bit error code in the "Status" module.<br>Rising edge (False → True) clears error code. |
| | 1 | HW trigger Disable | 1 bit | This bit is used to disable the trigger. Valid for Trigger mode Trigger and Free run.<br>• False (0): Trigger activated.<br>• True (1): Trigger disabled.<br>If the digital input "Trigger enable" is used, both conditions (digital input "Hardware Trigger" and "Trigger Disable Bit") must be set to "Enable" to accept a trigger. |
| | 2 | Trigger | 1 bit | Rising edge (false → true): Trigger is executed immediately.<br>If the trigger could not be executed, the Trigger acknowledge Bit stays false and "Error status" module has the error code "1: Failure trigger request".<br>See also Timing diagram, Chapter Case: Trigger not possible (not ready) |
| | 3 | Change job | 1 bit | Rising edge (false → true): Switch to job with number "Job number" from Control module.<br>When executing this request, delays may occur.<br>After a successful job change, the "Job number" byte in the "Status" module shows the same value as in the Control module.<br>If the job change could not be executed due to error (due to an error, e.g. wrong job number), the "Error status" module has the error code "2: Failure change job" (and Ready stays false!). See also Timing diagram, Chapter Case: Job change not possible (e.g. wrong job number) |

| Byte Offset | Bit Adr.. | Name | Data type | Meaning |
|---|---|---|---|---|
| | 4 | Switch-to-Run | 1 bit | Rising edge (false → true) "Switch to Run" is executed. Success or failure of Switch to Run request is shown in the "Error status" module (error code "3: Failure Switch to run request") and bit "Operation Mode". See also Timing diagram, Chapter Case: Switch to run not possible |
| | 5-7 | Reserve | | |
| 1 | | Reserve | 1 byte | |
| 2 | | Job number | U8 | Job number to be switched to, on the rising edge of the change job bit.<br>Binary value 1-255 for "Job number change".<br>0 stands for "No switching", even if the Change Job Bit changes. |

Timing diagrams for VISOR® communication (Page 41)

## 6.2 Module 2: “Status” (from VISOR® to PLC)

Name in PLC "STAT (6 bytes)"

| Byte Offset | Bit Addr. | Name | Data type | Description |
|---|---|---|---|---|
| 0 | 0 | Ready | 1 bit | VISOR® ready for next evaluation. Ready=1.<br>**Attention:**<br>The Ready bit is exclusively reserved for indicating the readiness of the VISOR® vision sensor for the next evaluation. It is not suitable for indicating that an evaluation has been completed or the results of an evaluation are available! |
| | 1 | Reserve | 1 bit | |
| | 2 | Trigger acknowledge | 1 bit | Acknowledge (confirmation) for successful trigger request (via Trigger Bit in Control module).<br>Acknowledge is deleted as a response to the deletion of the trigger bit.<br>If the trigger could not be executed, the Trigger Acknowledge Bit stays false. |
| | 3 | Change Job acknowledge | 1 bit | Acknowledge (confirmation) for the Change Job Request (via Change Job Bit in Control module) – independent of its success. Acknowledge is deleted as soon as the Change Job Request Bit has been deleted.<br>Success or failure of Change Job Request is shown in the bitfield "Error" (error code "2: Failure change job") and in the byte "Job number" in the Status module. If there are delays in executing the job change, this acknowledge bit can also be set with a delay. |

| Byte Offset | Bit Addr. | Name | Data type | Description |
|---|---|---|---|---|
| | 4 | Switch to run acknowledge | 1 bit | Acknowledge (confirmation) for the Switch to Run Request (via Switch to Run Request Bit in the Control module). Acknowledge is deleted as soon as the Request Bit is deleted. Success or failure of Switch to Run Request is shown in the bitfield "Error" (error code "3: Failure Switch to run request") and bit "Operation Mode".<br>Acknowledge is set after SensoConfig is closed and the job has been loaded from the flash or if an error has occurred. |
| | 5-7 | Reserve | | |
| 1 | | Reserve | 1 byte | |
| 2 | 0 | Digital Results | 1 bit | 12 RDBU |
| | 1 | | 1 bit | 09 RD |
| | 2 | | 1 bit | 05 PK |
| | 3 | | 1 bit | 06 YE |
| | 4 | | 1 bit | 07 BK |
| | 5 | | 1 bit | 08 GY |
| | 6 | Reserve | 1 bit | This byte is filled with the results of the digital switching outputs. The bit position is fixed. The value of the output is defined in the tab: Output/Digital output, Column: "Logical expression" in SensoConfig. If not selected as result output pin, or if no valid logical expression is assigned, the value is = 0. |
| | 7 | Reserve | 1 bit | |
| 3 | | Job number | U8 | Number of current job: Job number 1-255 |
| 4 | | Image ID | U8 | Image ID (0 - 255) is incremented by 1 with each job execution, independent of the trigger source. |

| Byte Offset | Bit Addr. | Name | Data type | Description |
|---|---|---|---|---|
| 5 | 0-3 | Error | 4 bit | 4 bit error code (decimal). Used to indicate errors in requests via the control module or VISOR® system errors. The error code can be reset by "Reset error" or is overwritten by the next error. In case of an archiving error (8), you can continue without a "Reset error".<br>0: No error<br>1: Error: Trigger request error (sensor not Ready)<br>2: Error: Change job<br>3: Error: Switch-to-Run<br>5: Error: Interface not active in job<br>7: Focus lock time<br>8: Error: Archiving<br>15: System error |
| | 4 | Trigger Mode | 1 bit | 1 = Free run<br>0 = Trigger |
| | 5 | Reserve | 1 bit | |
| | 6 | Operation mode | 1 bit | 1 = Run<br>0 = Config |
| | 7 | Reserve | 1 bit | |

## 6.3 Module 3: "Data" (from VISOR® to PLC)

Name in PLC "DATA (2 + 8 / 16 / … / 192 / 252 Bytes)"

| Byte Offset | Bit Addr. | Name | Data type | Description |
|---|---|---|---|---|
| 0 | | Image ID | U8 | Image ID (0 - 255) is incremented by 1 with each job execution, independent of the trigger source. |
| 1 | 0 | Result data overrun | 1 bit | Result data has been truncated.<br>1: Data overrun = truncated<br>0: No overrun |
| | 1 - 7 | Reserve | 7 Bit | |
| 2 | | Result data | Byte array | Data as defined in SensoConfig in "Output/Data Output/Detector-Specific payload".<br>When using PROFINET "binary" must be enabled in the Interfaces tab. |

## 6.4 Module 4: "Request" (From PLC to VISOR®)

Name in PLC "REQU (4 + 8 / 16 / … / 192 / 250 Bytes)"

| Byte Offset | Bit Addr. | Name | Data type | Meaning |
|---|---|---|---|---|
| 0 | 1 | Key | 1 byte | Request key (Request counter) |
| 1 | 1 | Reserve | 1 byte | Reserve |
| 2 | 1 | Reserve | 1 byte | Reserve |
| 3 | 1 | Reserve | 1 byte | Reserve |
| 4 | | Request Data | Byte array | Additional information: Overview telegrams (Page 87) |

## 6.5 Module 5: "Response" (from PLC to VISOR®)

Name in PLC "RESP (4 + 8 / 16 / … / 192 / 250 Bytes)"

| Byte Offset | Bit Addr. | Name | Data type | Description |
|---|---|---|---|---|
| 0 | | Key | U8 | Response key = mirrored from request |
| 1 | 0 | Result Data overrun | 1 bit | Response data has been truncated |
| | 1-7 | Reserve | 7 Bit | |
| 2 | | Reserve | 1 byte | |
| 3 | | Reserve | 1 byte | |
| 4 | | Result Data | Byte array | Additional information: Overview telegrams (Page 87) |

## 6.6 Start / end criteria for each telegram

| Telegram ("Control" module) | Start condition ("Status" module) | Acceptance confirmation ("Status" module) | Execution confirmation ("Status" module) |
|---|---|---|---|
| Trigger | Ready = True | Trigger acknowledge = True | Image ID changed |
| Change job | / | Change Job acknowledge = True | Job number changes |
| Switch-to-Run | Operation Mode = False | Switch-to-Run acknowledge = True | Operation Mode = True |

# 7 Timing diagrams for VISOR® communication

### Case: Trigger ok

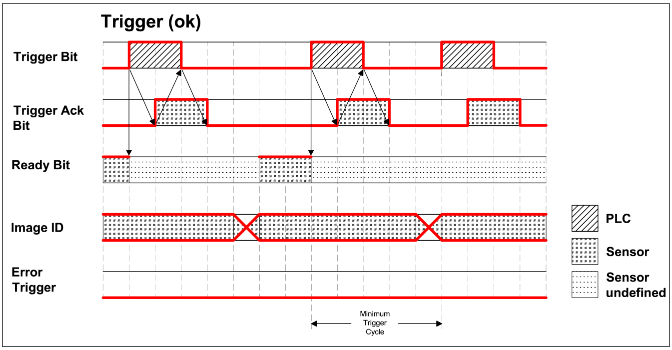



*Fig. 21: Timing Trigger ok*

### Case: Trigger not possible (not ready)

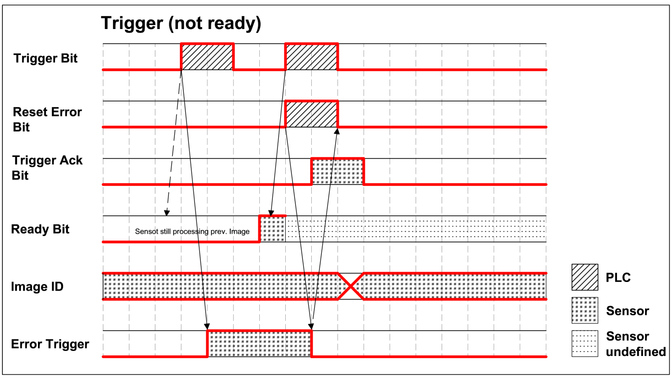



*Fig. 22: Timing Trigger not ready*

### Case: Job change ok

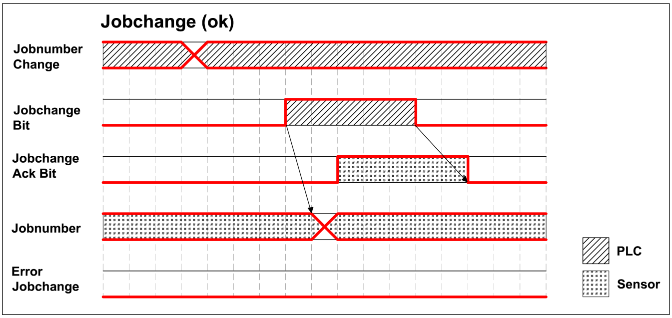



*Fig. 23: Timing Job change ok*

### Case: Job change delayed

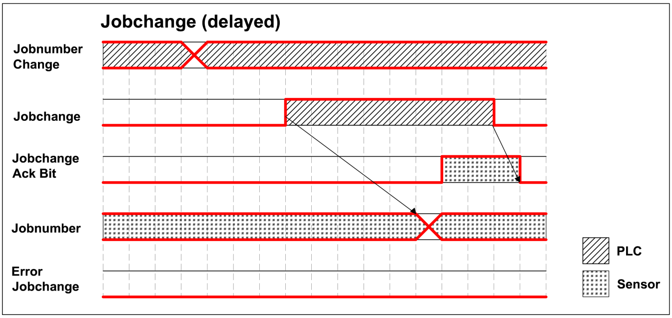



*Fig. 24: Timing Job change delayed*

### Case: Job change not possible (e.g. wrong job number)

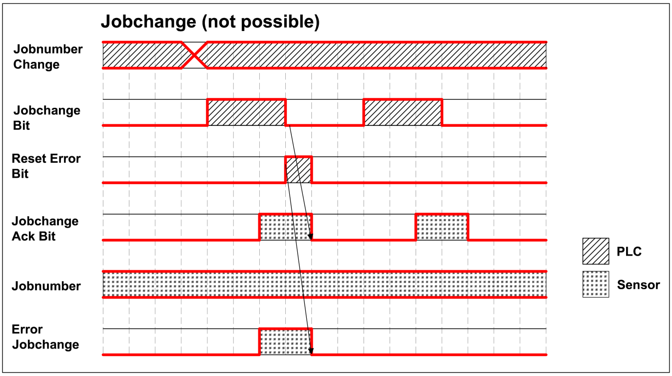



*Fig. 25: Timing Job change not possible*

### Case: Switch to run ok

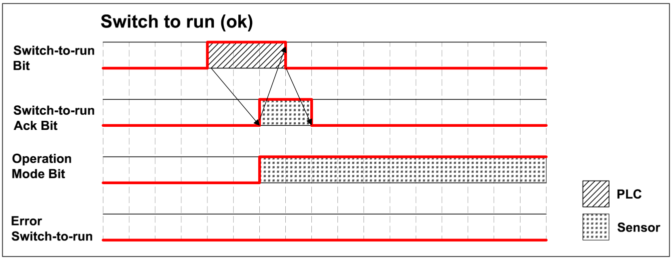



*Fig. 26: Timing Switch to run ok*

### Case: Switch to run not possible

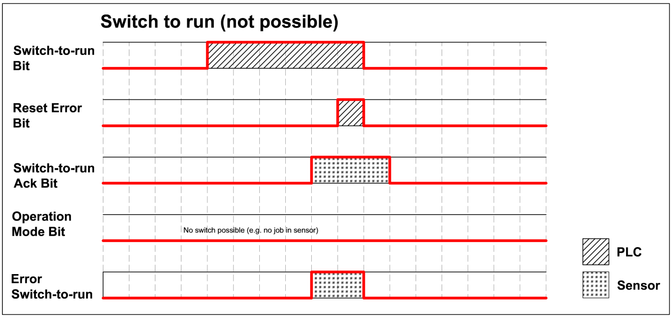



*Fig. 27: Switch to run not possible*

# 8 Request sequences

### Important recommendations for PLC programmers

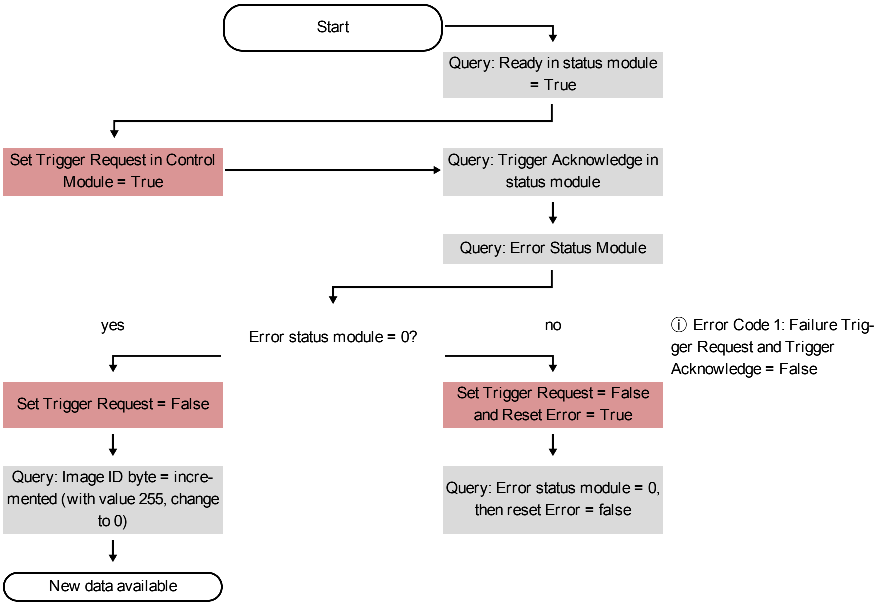
1. Follow the sequence of requests
2. Wait for complete execution of an action before sending the next one. Complete execution takes place when the image ID changes in the trigger request, or the corresponding acknowledge bit is set for the other requests.

   **NOTE:**
   The complete execution of an action cannot be recognized as safe due to the low/high change of READY, since due to possibly long cycle times between PLC and VISOR® (e.g. 32ms), READY may never become low.
3. READY should always be high before a trigger request is sent

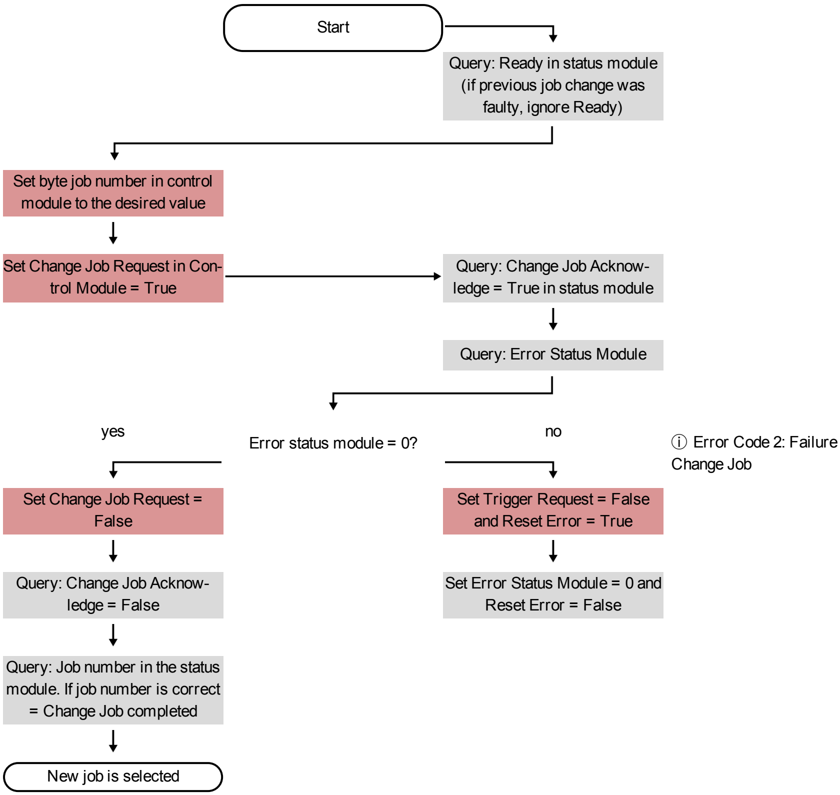
### Accepting / discarding of requests of the control module

1. Request is accepted with an increasing acknowledge bit
2. Request is discarded if the error bit is set.
3. Request is discarded without an error bit and acknowledge bit if the sensor is still processing the previous request and no acknowledgment has yet been set for it. (i.e. not following the recommended handshake)

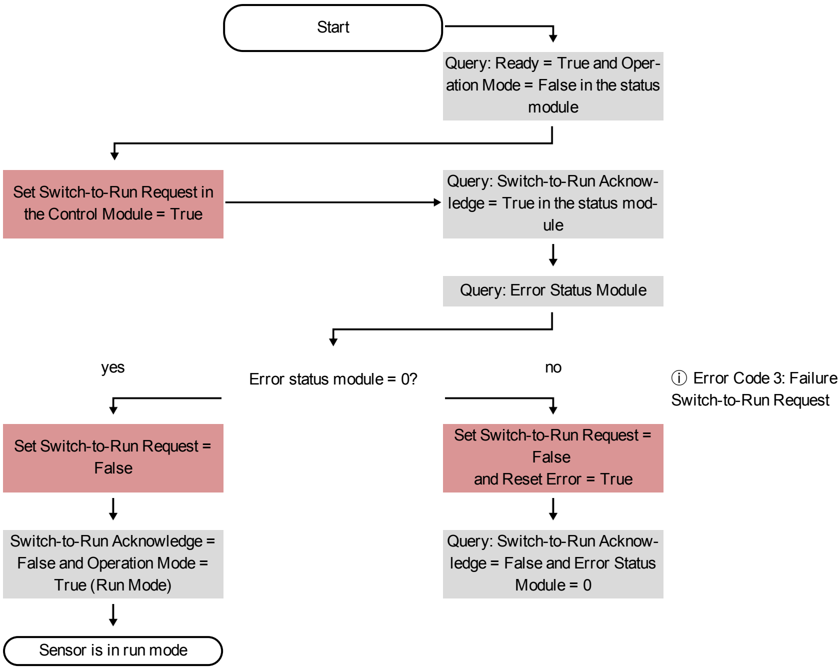
## 8.1 Trigger Request Sequence



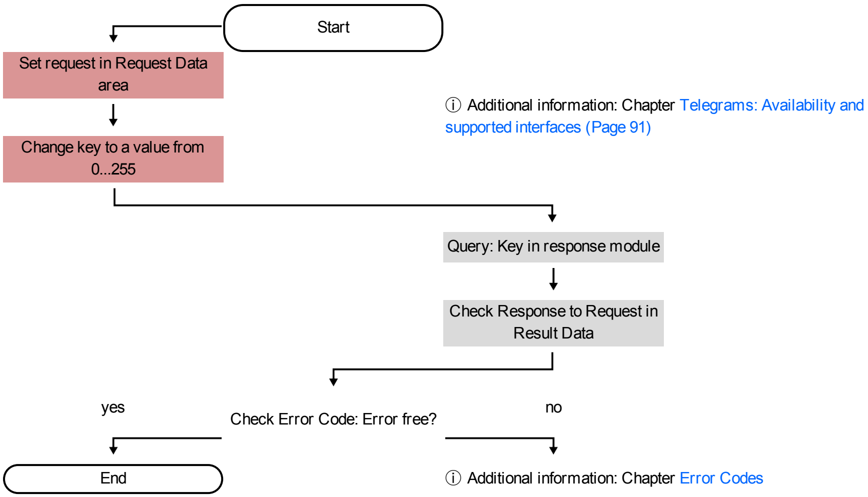
## 8.2 Change job request sequence



## 8.3 Switch to Run sequence



## 8.4 Sequence for requests via request/response module




**Additional information:**

Telegrams: Availability and supported interfaces (Page 91)

Error codes (Page 94)

**Error Reset (depicted in the use case "Job change not possible")**

1. Reset with "Reset Error Bit"
2. Error bits are overwritten by new error bits.

# 9 PROFINET

This section explains how to operate the VISOR® vision sensor with PROFINET.

## 9.1 Siemens S7-1200 TIA 12 configuration example

This description shows all PLC screenshots in English; switch the TIA software to English if necessary.

### 9.1.1 Create new project

New project with: Project / Create new project

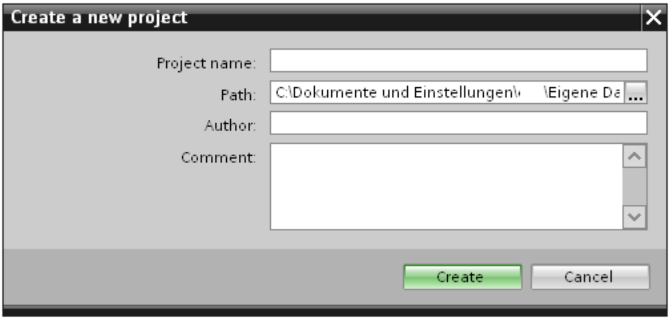



*Fig. 28: PROFINET Create new project*

### 9.1.2 Selecting the GSD file

First a PROFINET PLC must be added to the project.

In order to be able to use the PROFINET functions of the VISOR® vision sensor, the latest version of the corresponding VISOR® GSD file must be installed. This is done at: Options/Install general station description file. The EDS file can be found in the installation path for the VISOR® in: ...\SensoPart\VISOR vision sensor\Tools\PROFINET and is also available for download at www.-sensopart.com.

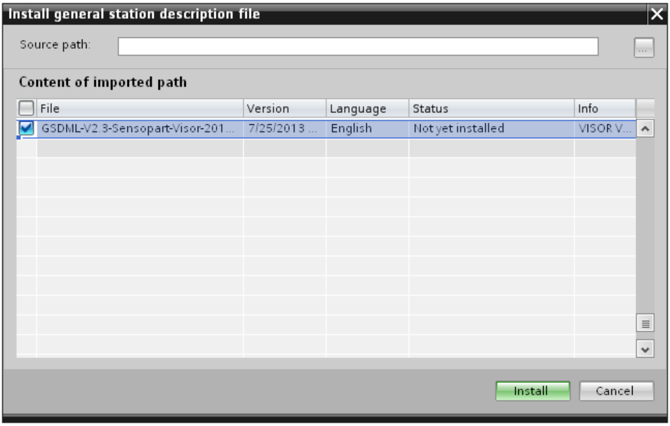



*Fig. 29: Selecting and installing the GSD file*

## 9.1.3 Adding the VISOR® vision sensor to the project

The VISOR® modules are added in the hardware catalog: Other field devices/PROFINET IO/-sensors/SensoPart Industriesensorik GmbH.

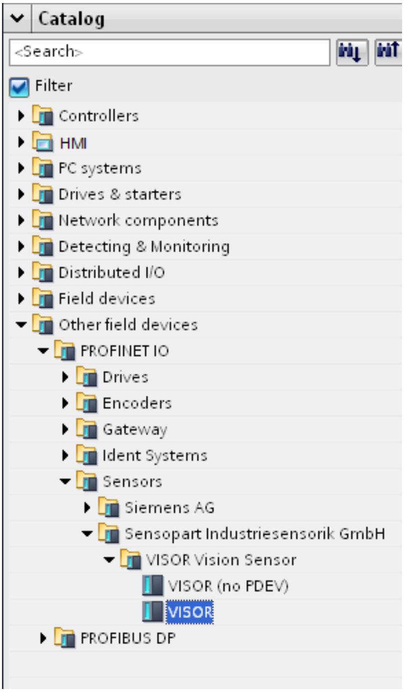



*Fig. 30: Adding the VISOR® to the project*

## Connecting the VISOR® to the PLC

You can now drag a VISOR® module from the catalog and drop it in the Network View. The VISOR® is connected to the PLC via PROFINET (Network View tab).

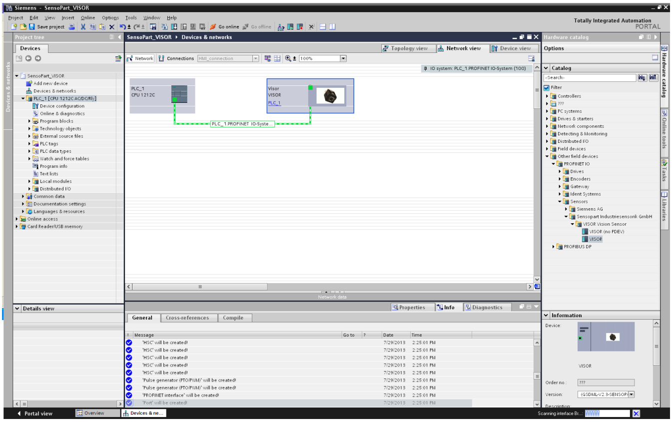


*Fig. 31: Connecting the VISOR® to the PLC*

## Definition of I/O data

In the tab "Device view", the modules CTRL (Control) and STAT (Status) are active by default. As an option, the module DATA (Data module) can be added with a certain use size.

In this example: 2 bytes + 16 bytes of payload (1 byte: Image ID, 1 byte: Result data overrun (see Module 3: "Data" (from VISOR® to PLC) (Page 36)), + 16 bytes of data). If the data are longer than the defined range, these are truncated (in this case: Result data overrun = 1); if it's shorter, the rest of the 16 bytes are filled with 00h.

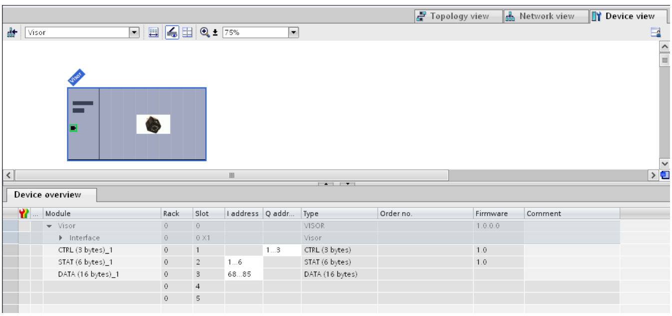



*Fig. 32: Enter I/O data*

# Configuring the VISOR® IP address

## Option 1: In the project

The IP address for the VISOR® can be assigned through the project in the PLC. Select option "Set IP address in the project" and enter IP address. The address from the "IP address" field will be written to the VISOR®. The IP addresses of the PLC and the VISOR® must be different from each other but correspond to each other, i.e., fall within the same address space.

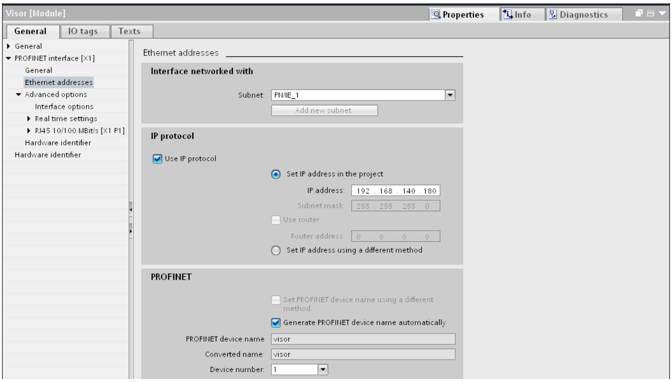



*Fig. 33: Configuring the VISOR® IP address in the project*

The VISOR® can also be used without a running PROFINET, and accordingly can be configured via SensoFind. If the IP address of the VISOR® does not match the one in the TIA project, the PLC will configure the IP address instead. In this case, the original configuration in the VISOR® will be overwritten with 0.0.0.0. This means that the IP address is set correctly but the IP configuration is deleted (this is important for a restart, possibly without a connected PLC).

## Option 2: In SensoFind

The IP address of the VISOR® can also be configured via SensoFind. Select option "Set IP address using a different method" in the PLC / TIA interface. Configure the IP address via SensoFind (See Chapter: Settings in SensoFind (Page 15)).

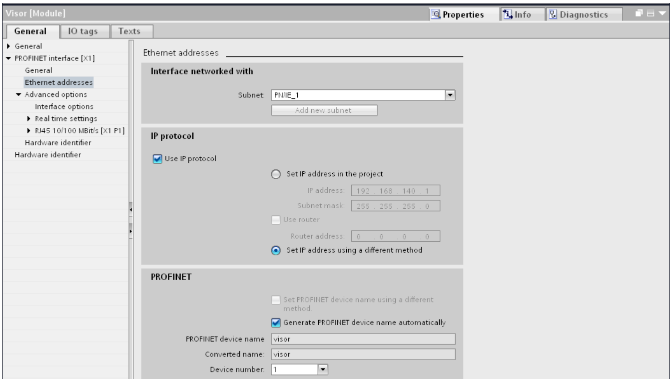



*Fig. 34: Configure the IP address for the VISOR® in SensoFind; the corresponding settings can be found in the PLC/TIA interface*

## Set the name with TIA interface

There are two ways to configure the name for the VISOR® from the TIA Portal.

### Generate name automatically

The PROFINET name for the VISOR® can be generated automatically in the PLC. Option: "Generate PROFINET device name automatically" takes the name from the project.

### Set name manually

If the option "Set PROFINET device name using a different method" is selected any name can be set.

Information: In the field "Converted name", a different name than entered is displayed, which is then also used. Da im PROFINET nicht alle Zeichen genutzt werden können ist eventuell eine Konvertierung notwendig und wird hier automatisch gemacht (Namen müssen DNS kompatibel sein, s. dazu auch Kap. Settings in SensoFind (Page 15)).

If the VISOR®'s name is configured using the TIA Portal, it must be written to the sensor with the "PROFINET device name" tool (as described in section Writing a name to VISOR® (Page 57)).

The PROFINET name in the project and in the VISOR® must match.

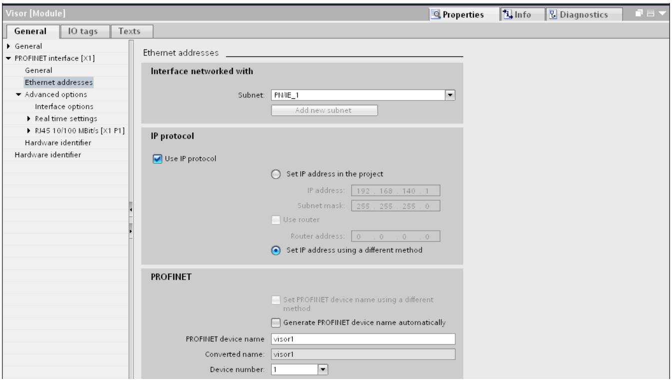



*Fig. 35: Set name in project*

## 9.1.4 Writing a name to VISOR®

In order to be able to establish communications, the PROFINET name must be written to the VISOR® in case it needs to be updated.

This is done with the tool: Online/Assign PROFINET device name. Select the corresponding device (VISOR®) and apply the name with "Assign name."

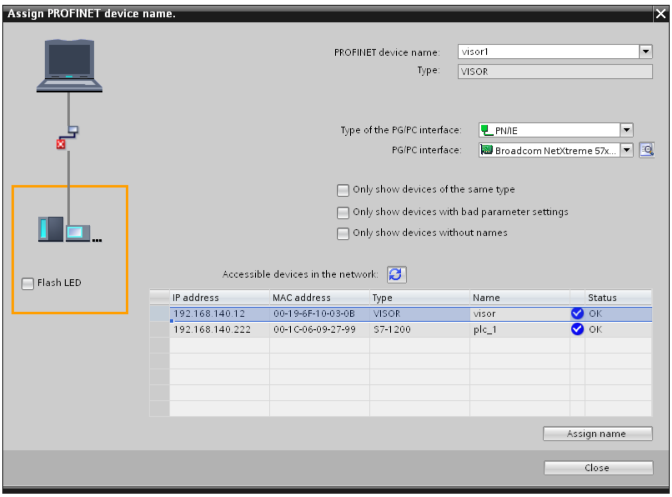



*Fig. 36: Writing a name to VISOR®*

## 9.1.5 Load the project onto the PLC

To finish the configuration and save changes of the project: 1. translate and 2. transfer / write to the PLC


*Fig. 37: Translate project and write to PLC*

## 9.1.6 Mapping of output data

The VISOR® vision sensor's output data can be mapped to the data in the PROFINET log as follows:

**Step 1) The start address for an input variable can be taken from "Device Overview".**

| | i | Name | Address | Display format | Monitor value | Modify value |
|---|---|---|---|---|---|---|
| 1 | | | %IB68 | Hex | 16#00 | |
| 2 | | | %IB69 | Hex | 16#00 | |
| 3 | | "Data1" | %IB70 | Hex | | |
| 4 | | "Data2" | %IB71 | Hex | | |
| 5 | | "Data3" | %IB72 | Hex | | |
| 6 | | "Data4" | %IB73 | Hex | | |
| 7 | | "Data5" | %IB74 | Hex | | |
| 8 | | "Data6" | %IB75 | Hex | | |
| 9 | | "Data7" | %IB76 | Hex | | |
| 10 | | "Data8" | %IB77 | Hex | | |
| 11 | | "Data9" | %IB78 | Hex | | |
| 12 | | "Data10" | %IB79 | Hex | | |
| 13 | | "Data11" | %IB80 | Hex | | |
| 14 | | "Data12" | %IB81 | Hex | | |
| 15 | | "Data13" | %IB82 | Hex | | |
| 16 | | "Data14" | %IB83 | Hex | | |
| 17 | | "Data15" | %IB84 | Hex | | |
| 18 | | "Data16" | %IB85 | Hex | | |
| 19 | | | <Add new> | | | |

*Fig. 38: Table of variables*

**Step 2) Creating a tag table in the PLC**

Device overview

| Module | Rack | Slot | I address | Q address | Type | Order no. | Firmware | Comment |
|---|---|---|---|---|---|---|---|---|
| Visor | 0 | 0 | | | VISOR | | | |
| Interface | 0 | 0 X1 | | | Visor | | | |
| CTRL (3 bytes)_1 | 0 | 1 | | 1...3 | CTRL (3 bytes) | | | |
| STAT (6 bytes)_1 | 0 | 2 | 1...6 | | STAT (6 bytes) | | | |
| DATA (2 + 16 bytes)_1 | 0 | 3 | 68...85 | | DATA (2 + 16 bytes) | | | |
| REQU (4 + 16 bytes)_1 | 0 | 4 | | 64...83 | REQU (4 + 16 bytes) | | | |
| RESP (4 + 16 bytes)_1 | 0 | 5 | 86...105 | | RESP (4 + 16 bytes) | | | |

*Fig. 39: Device overview*

**Step 3) Creating the configuration in SensoFind and saving the configured log as a CSV file.**

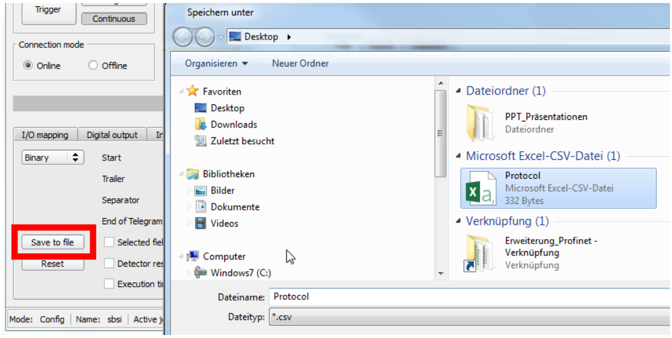
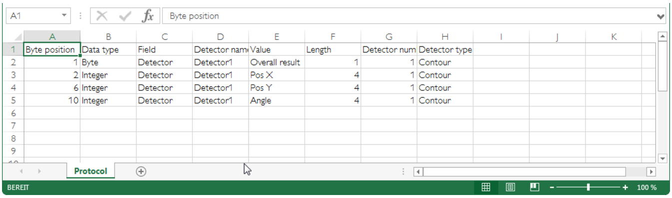



*Fig. 40: Output format saved as CSV file*

### Step 4) Opening the file with the text program

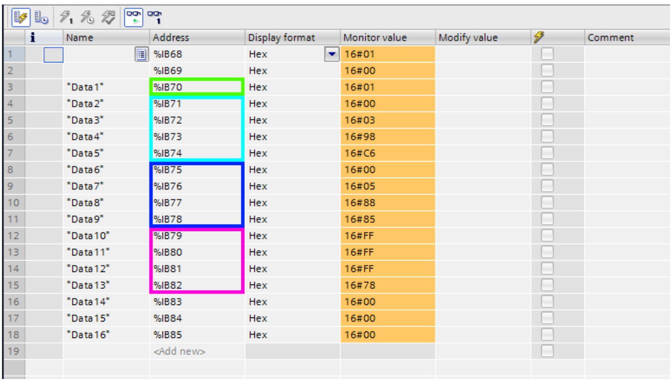


| Byte position | Data type | Field | Detector nam | Value | Length | Detector num | Detector type |
|---|---|---|---|---|---|---|---|
| 1 | Byte | Detector | Detector1 | Overall result | 1 | 1 | Contour |
| 2 | Integer | Detector | Detector1 | Pos X | 4 | 1 | Contour |
| 6 | Integer | Detector | Detector1 | Pos Y | 4 | 1 | Contour |
| 10 | Integer | Detector | Detector1 | Angle | 4 | 1 | Contour |

*Fig. 41: Output protocol in Excel representation*

For a description of the format of the PROFINET Data module, please refer to Module 3: "Data" (from VISOR® to PLC) (Page 36)

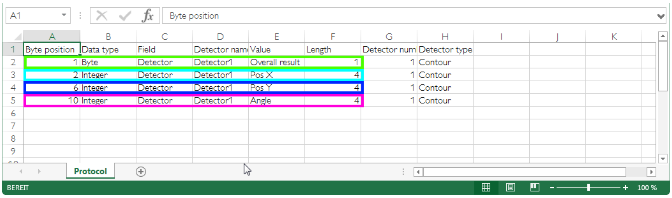
### Step 5) The result is the following assignment between the input data of the PLC



| | i | Name | Address | Display format | Monitor value | Modify value | | Comment |
|---|---|---|---|---|---|---|---|---|
| 1 | | | %IB68 | Hex | 16#01 | | | |
| 2 | | | %IB69 | Hex | 16#00 | | | |
| 3 | | "Data1" | %IB70 | Hex | 16#01 | | | |
| 4 | | "Data2" | %IB71 | Hex | 16#00 | | | |
| 5 | | "Data3" | %IB72 | Hex | 16#03 | | | |
| 6 | | "Data4" | %IB73 | Hex | 16#98 | | | |
| 7 | | "Data5" | %IB74 | Hex | 16#C6 | | | |
| 8 | | "Data6" | %IB75 | Hex | 16#00 | | | |
| 9 | | "Data7" | %IB76 | Hex | 16#05 | | | |
| 10 | | "Data8" | %IB77 | Hex | 16#88 | | | |
| 11 | | "Data9" | %IB78 | Hex | 16#85 | | | |
| 12 | | "Data10" | %IB79 | Hex | 16#FF | | | |
| 13 | | "Data11" | %IB80 | Hex | 16#FF | | | |
| 14 | | "Data12" | %IB81 | Hex | 16#FF | | | |
| 15 | | "Data13" | %IB82 | Hex | 16#78 | | | |
| 16 | | "Data14" | %IB83 | Hex | 16#00 | | | |
| 17 | | "Data15" | %IB84 | Hex | 16#00 | | | |
| 18 | | "Data16" | %IB85 | Hex | 16#00 | | | |
| 19 | | | <Add new> | | | | | |

*Fig. 42: Input data PLC*

... and the configured protocol:

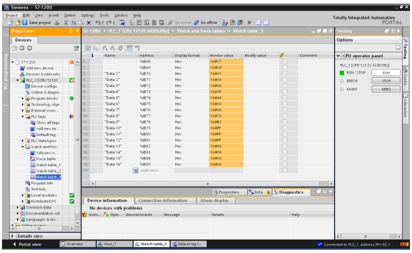


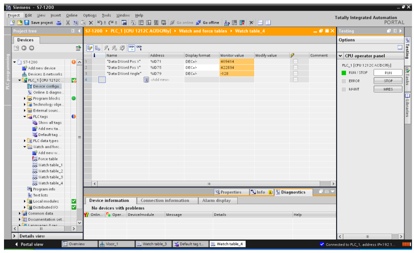
| Byte position | Data type | Field | Detector name | Value | Length | Detector num | Detector type |
|---|---|---|---|---|---|---|---|
| 1 | Byte | Detector | Detector1 | Overall result | 1 | 1 | Contour |
| 2 | Integer | Detector | Detector1 | Pos X | 4 | 1 | Contour |
| 6 | Integer | Detector | Detector1 | Pos Y | 4 | 1 | Contour |
| 10 | Integer | Detector | Detector1 | Angle | 4 | 1 | Contour |

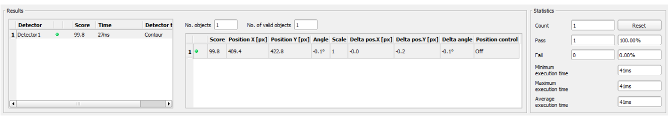
*Fig. 43: In the vision sensor configured protocol*

### Conversion of binary values

All detector-specific payloads with decimal places will be transmitted as integers multiplied by 1000, and accordingly must be divided by 1000 after the data is received. The values are transferred in the format "Big-endian". The length is based on the value, e.g., score 32 bits (DWord).







## 9.2 PLC example programs

The following PLC example programs show some basic functions.

## PLC example 1: Trigger when VISOR® Ready

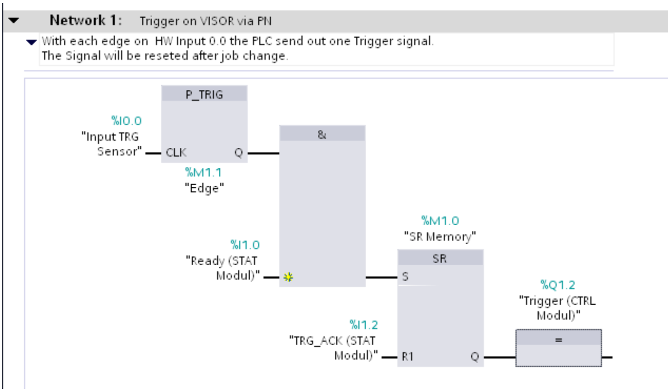



*Fig. 44: Trigger when VISOR® Ready, (without error handling)*

## PLC example 2: Send job number to VISOR®

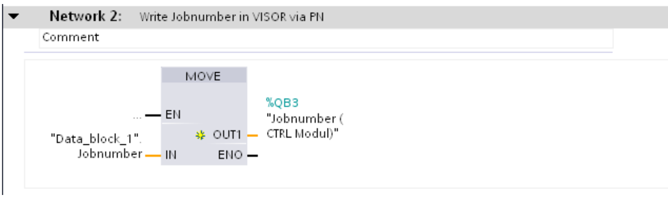



*Fig. 45: Send job number*

### PLC example 2.1: Job change when VISOR® Ready

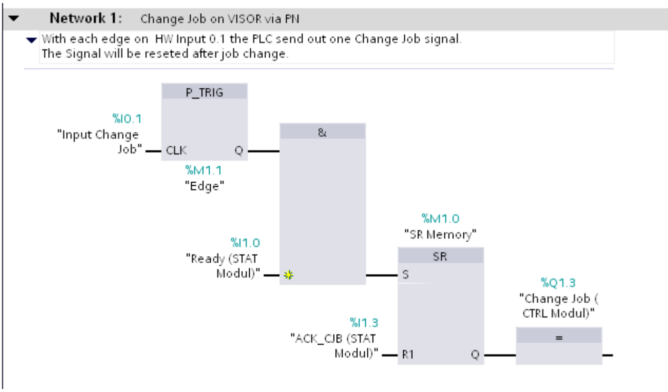



*Fig. 46: Job change when VISOR® Ready, (without error handling)*

### PLC example 3: Switch to Run when VISOR® in configuration mode

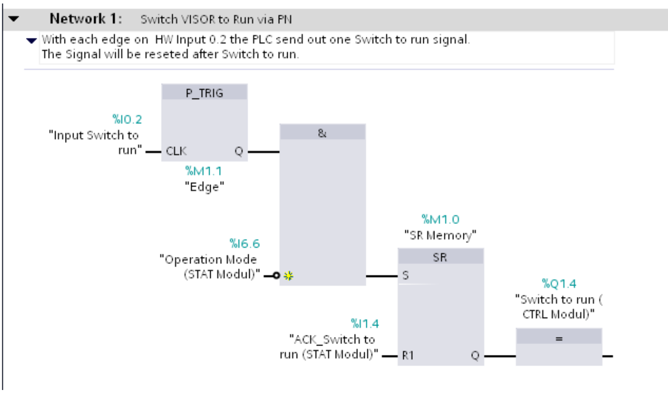



*Fig. 47: Switch to Run when VISOR® in configuration mode (without error handling)*

### PLC example 4: Data transfer, data block on PLC, creating tags

Variable "Data Array" (type: Array of Byte) Length (34 bytes) = Payload (32) + 2 bytes (header)

(Module "Data" with 32 bytes: User data + 1 byte: Image ID + 1 byte: Result data overrun = 34 bytes)

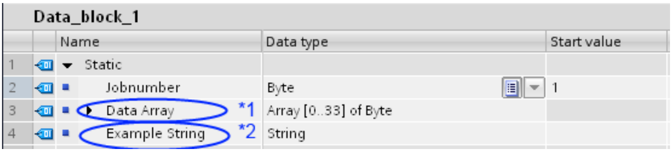



*Fig. 48: Data block for data transfer*

## PLC example 4.1: Data transfer

Data transfer from input memory to data block with function DRPD_DAT. Access to diagnosis address via "PLC_Tags". Conversion of data of the read codes into a string with variable data length.

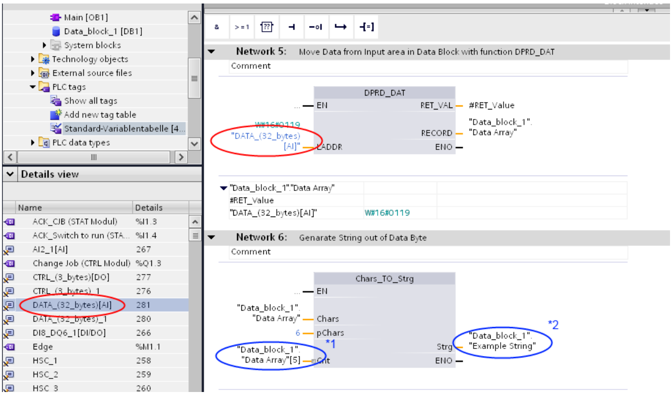



*Fig. 49: Data transfer*

## PLC example 4.2, VISOR® telegram settings

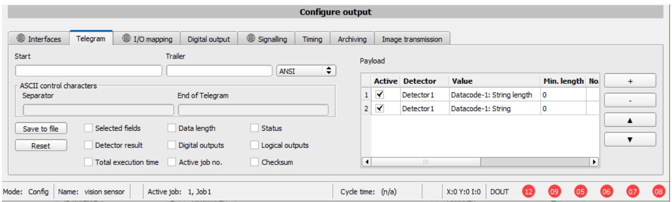



*Fig. 50: Settings for sample telegram in VISOR®*

# 10 EtherNet/IP

This section explains how to operate the VISOR® vision sensor with EtherNet/IP.

## 10.1 Rockwell CompactLogix™ configuration example

Following is a description of the PLC settings required for data transfers between the VISOR® vision sensor and the PLC via EtherNet/IP (using Rockwell CompactLogix™ as an example).

### Rockwell Studio 5000

This description shows all PLC screenshots (Studio 5000, version 30 under Windows 7) in English language. Switch Rockwell software to English if necessary.

1. Create a new project: "Create" / "New Project

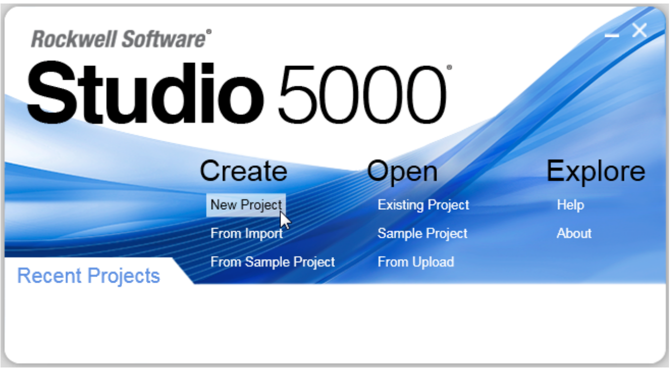



*Fig. 51: EtherNet/IP Create new project*

2. Select the appropriate PLC type and assign a name.

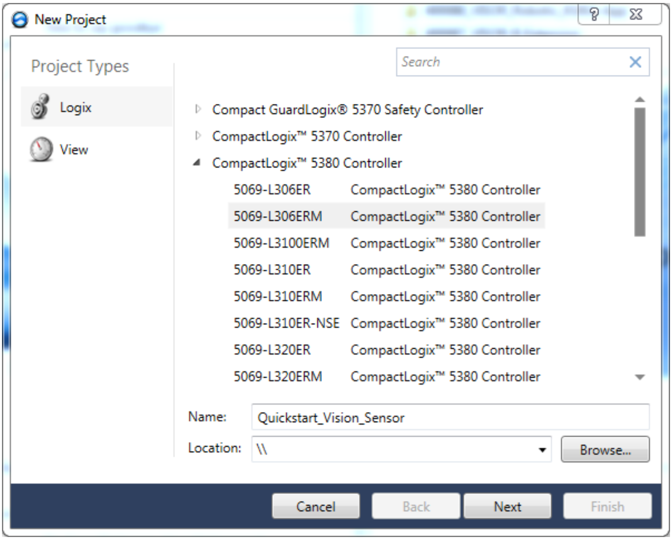



*Fig. 52: EtherNet/IP Select the PLC type.*

3. Apply the default settings. Click on "Finish" to create the project.

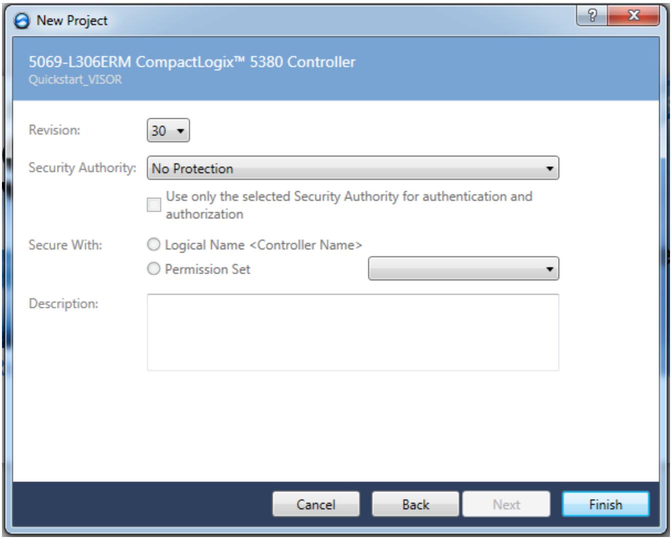



*Fig. 53: EtherNet/IP Apply the default settings.*

## 10.2 Installation of EDS file

The project view opens. In order to be able to use the EtherNet/IP functions of the VISOR® vision sensor, the latest version of the corresponding VISOR® EDS file must be installed.

If the controller does not support EDS file, follow instructions in chapter Create module/Using a Generic Device (without EDS file).

1. Install EDS file under "Tools" / "EDS Harware Installation Tool".

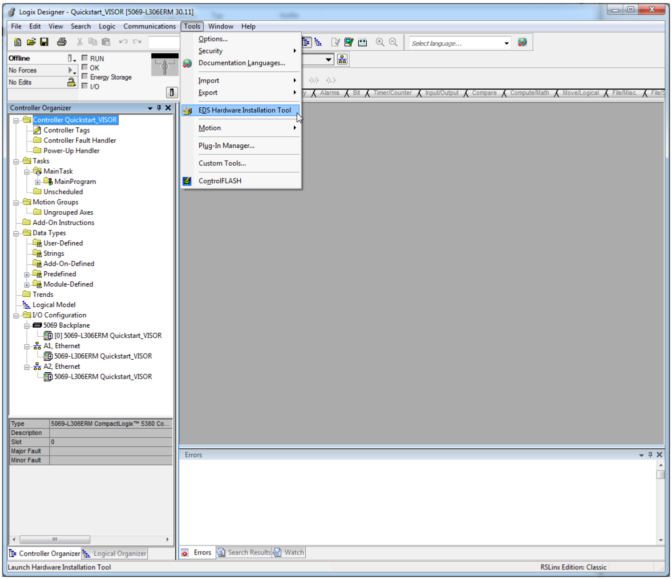



*Fig. 54: Project view, Tool, EDS Hardware Installation Tool*

2. Confirm information with "Next".


*Fig. 55: Confirming information*

3. Select "Register to EDS file(s)" in the options

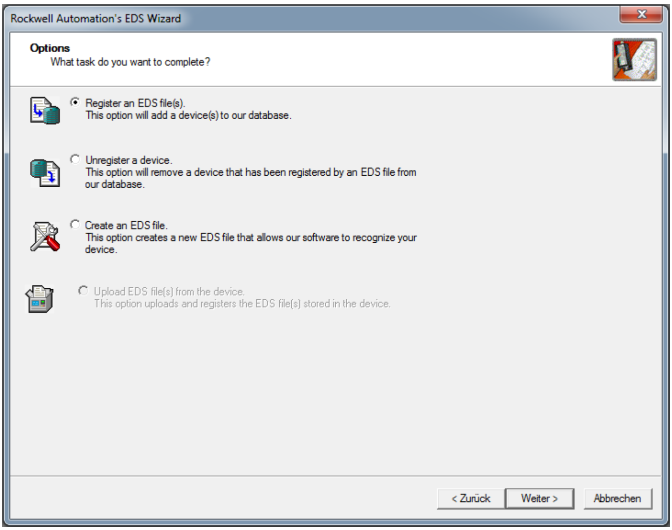



*Fig. 56: Register an EDS File(s)*

4. Select "Register a single file"

**NOTE:**
The exact same EDS file can be used for all VISOR® vision sensors.

5. Specify the path to the EDS file.

The EDS file can be found in the installation path of the VISOR® under: \SensoPart\VISOR Vision Sensor\Tools\EtherNet/IP and is also available for download at www.sensopart.com

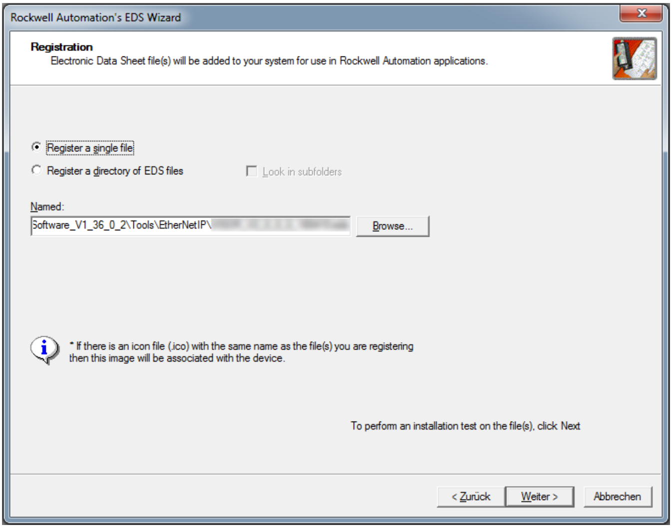



*Fig. 57: Select EDS file*

6. Confirm EDS file test.

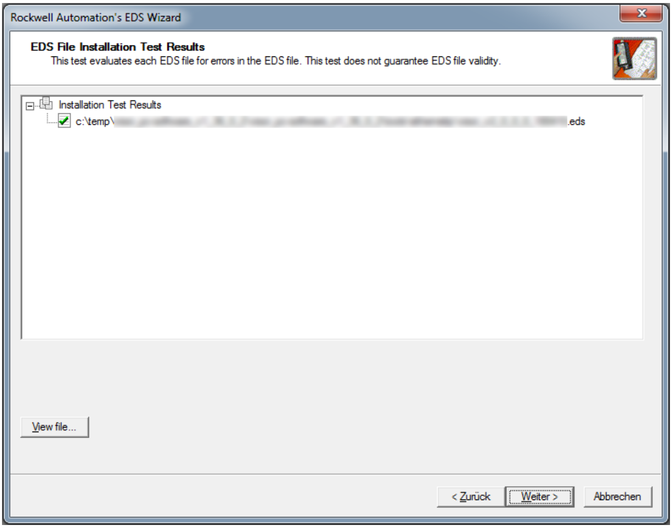



*Fig. 58: EDS file test*

7. Select icon if required or continue with standard icon.

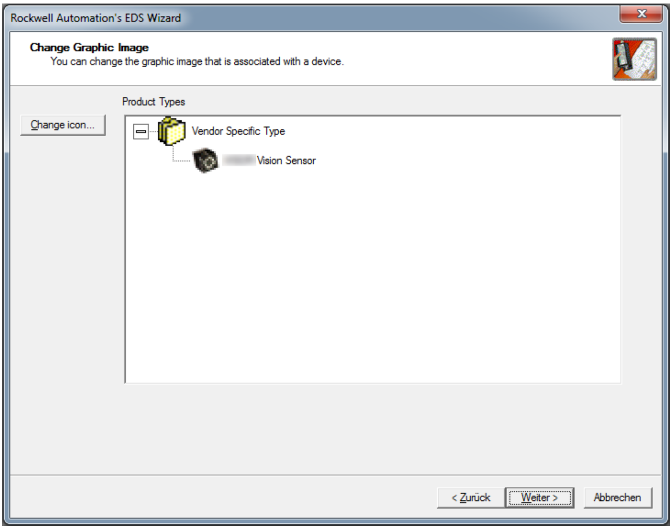



*Fig. 59: Icon*

8. Confirm the installation.

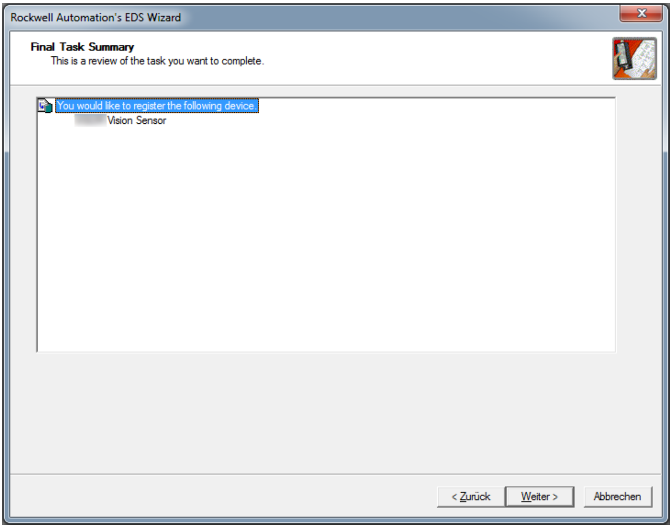



*Fig. 60: Confirming the installation*

9. Complete the installation with "Finish".

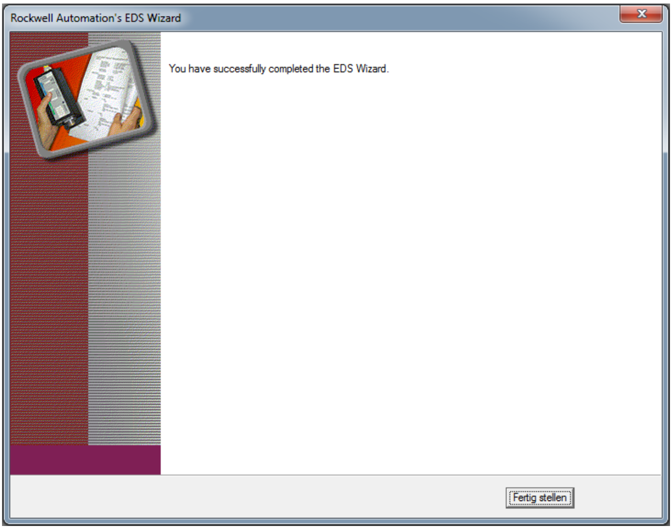


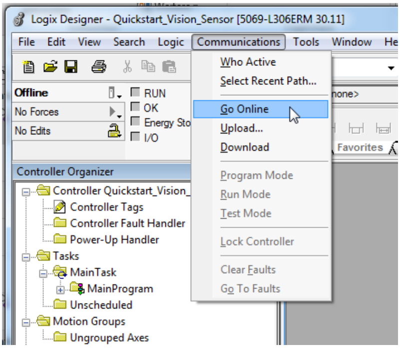

*Fig. 61: Finishing the installation*

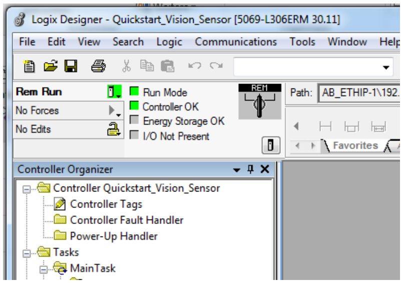
## 10.3 Create module

### 10.3.1 Selection via hardware catalog (with EDS file)

1 To go online with the project, select Communications / "Go Online".

**NOTE:**
Before this, the project path must be configured correctly.



2. Create a new module by right clicking on the desired network connection.

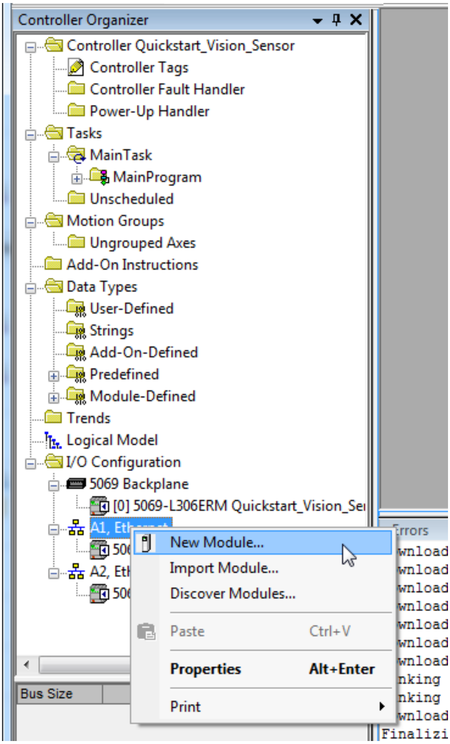
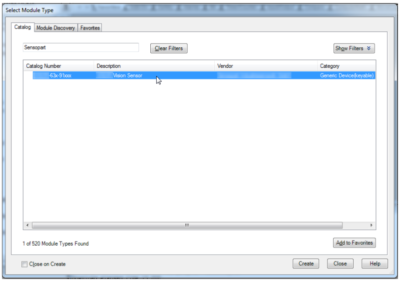
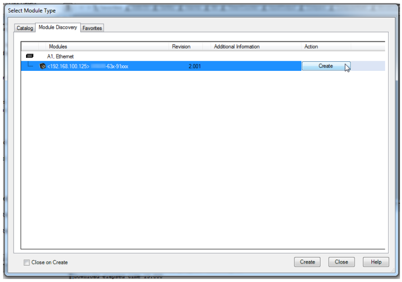



*Fig. 62: Creating a new module*

3. Select VISOR® from the catalog or search for available devices online.

**NOTE:**
For the option "Search online" the software must already be online (see Create module / step 1).




You can search for "SensoPart" in the hardware catalog. The corresponding devices are listed. Alternatively, the "Module Discovery" tab can be used to search for accessible participants.

4. Assign device name and IP address of the VISOR®.

- The device name will be used as a variable name for the data later on.
- The IP address can be read out via SensoFind.

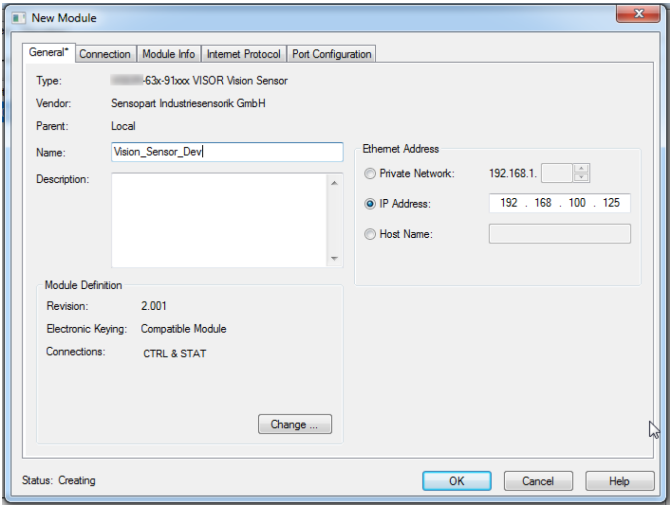
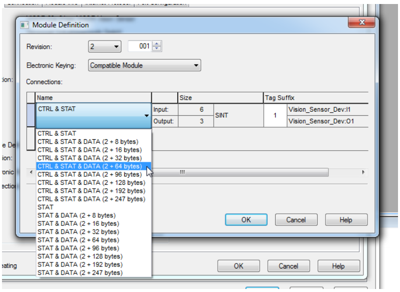
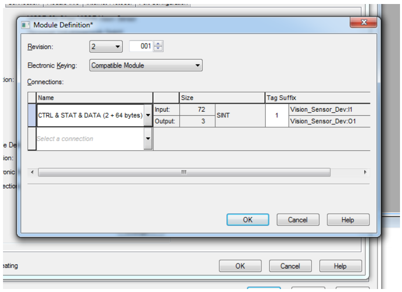



*Fig. 63: Configure the device name and IP address*

5. the desired modules and module sizes can be selected via "Change ...".




6. Set the desired refresh rate (RPI) in the "Connection" tab.

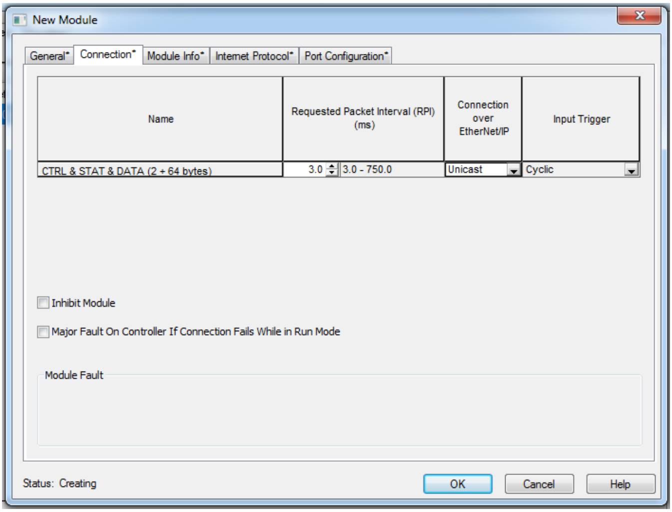



*Fig. 64: Set the refresh rate.*

7. Complete the participant's settings via "OK".

### 10.3.2 Using a Generic Device (without EDS file)

If the controller does not support EDS files, continue with the following steps.

1. Create a new module by right-clicking on the desired network connection.

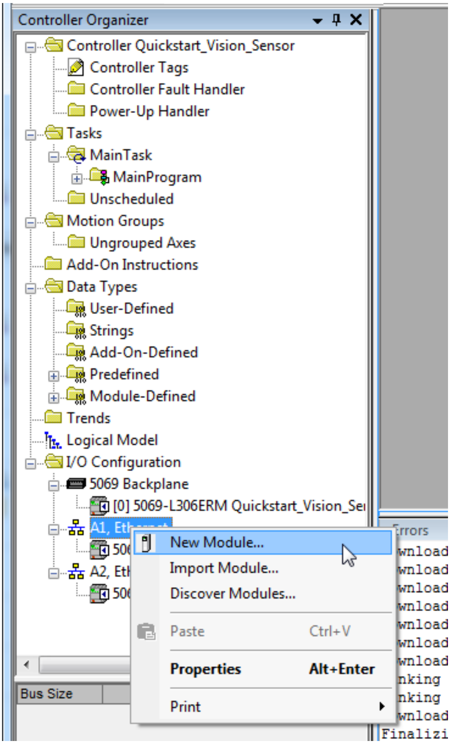



*Fig. 65: Creating a new module*

2. Select a module of type Ethernet Module - "Generic Ethernet Module" from the catalog

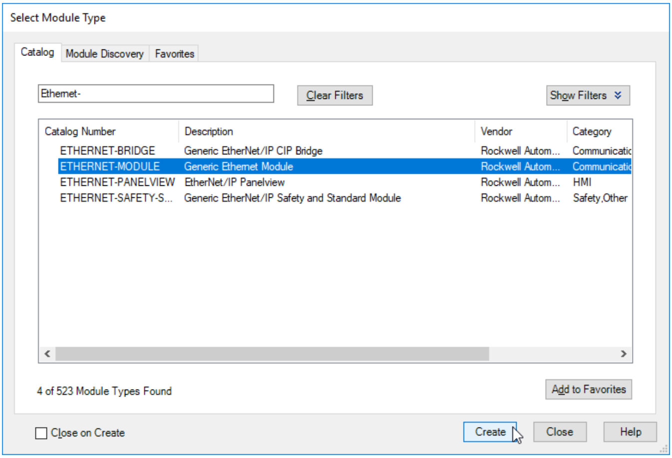



*Fig. 66: Selection of "Generic Ethernet Module"*

3. Assign device name and IP address of the VISOR® (**A**).

- The device name will be used as a variable name for the data later on.
- The IP address can be read out via SensoFind.

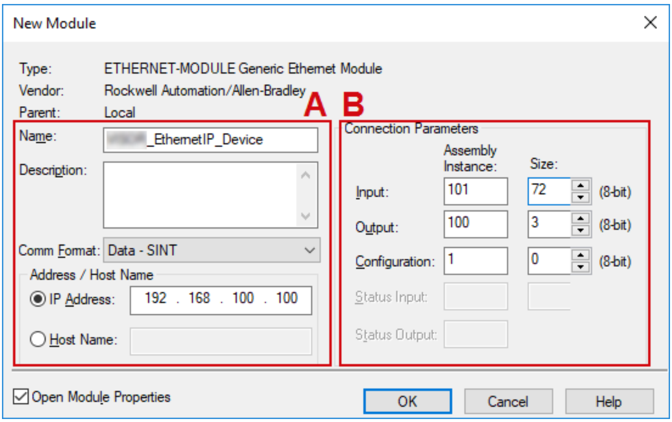



*Fig. 67: Assignment of the device name and IP address*

4. Change the data format to "Data - SINT" (8 bit format) with the "Comm Format" parameter (**A**).

5. Enter connection parameters (**B**) (see following table).

| | Assembly instance (dec) | Size (dec) | Assembly instance (hex) | Size (hex) |
|---|---|---|---|---|
| **Control + Status** | | | | |
| Input | 101 | 6 | 0x65 | 0x06 |
| Output | 100 | 3 | 0x64 | 0x03 |
| Configuration | 1 | 0 | 0x01 | 0x00 |
| **Control + Status + Data (2+8)** | | | | |
| Input | 102 | 16 | 0x66 | 0x10 |
| Output | 100 | 3 | 0x64 | 0x03 |
| Configuration | 1 | 0 | 0x01 | 0x00 |
| **Control + Status + Data (2+16)** | | | | |
| Input | 103 | 24 | 0x67 | 0x18 |
| Output | 100 | 3 | 0x64 | 0x03 |
| Configuration | 1 | 0 | 0x01 | 0x00 |
| **Control + Status + Data (2+32)** | | | | |
| Input | 104 | 40 | 0x68 | 0x28 |
| Output | 100 | 3 | 0x64 | 0x03 |
| Configuration | 1 | 0 | 0x01 | 0x00 |
| **Control + Status + Data (2+64)** | | | | |
| Input | 105 | 72 | 0x69 | 0x48 |
| Output | 100 | 3 | 0x64 | 0x03 |
| Configuration | 1 | 0 | 0x01 | 0x00 |
| **Control + Status + Data (2+96)** | | | | |
| Input | 105 | 104 | 0x69 | 0x68 |
| Output | 100 | 3 | 0x64 | 0x03 |
| Configuration | 1 | 0 | 0x01 | 0x00 |
| **Control + Status + Data (2+128)** | | | | |
| Input | 105 | 136 | 0x69 | 0x88 |
| Output | 100 | 3 | 0x64 | 0x03 |
| Configuration | 1 | 0 | 0x01 | 0x00 |
| **Control + Status + Data (2+192)** | | | | |
| Input | 105 | 200 | 0x69 | 0xCB |
| Output | 100 | 3 | 0x64 | 0x03 |

| | Assembly instance (dec) | Size (dec) | Assembly instance (hex) | Size (hex) |
|---|---|---|---|---|
| Configuration | 1 | 0 | 0x01 | 0x00 |
| **Control + Status + Data (2+247)** | | | | |
| Input | 105 | 255 | 0x69 | 0xFF |
| Output | 100 | 3 | 0x64 | 0x03 |
| Configuration | 1 | 0 | 0x01 | 0x00 |

## 10.4 Load the project onto the PLC

1. Download the project to the PLC via "Communications" / "Download" .

**NOTE:**
For this the software must already be online (see Create module / step 1).

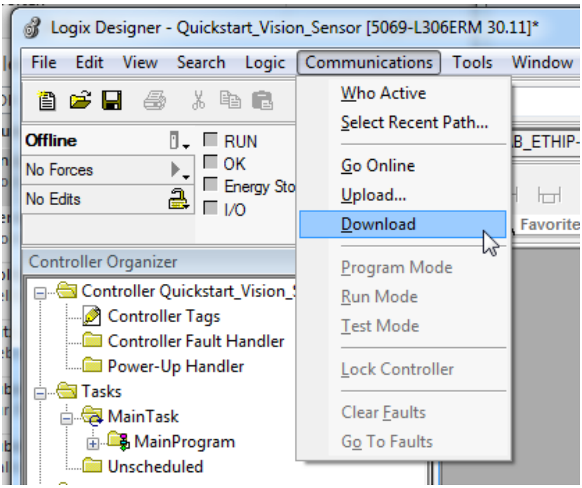



*Fig. 68: Download*

2. Check the notes and confirm with "Download".

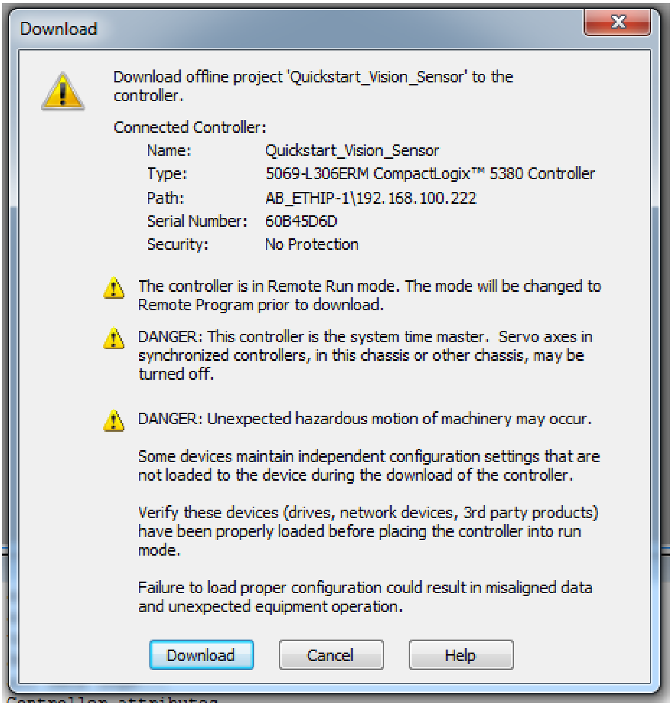



*Fig. 69: Information*

3. After a successful download, the VISOR® status is "Running".

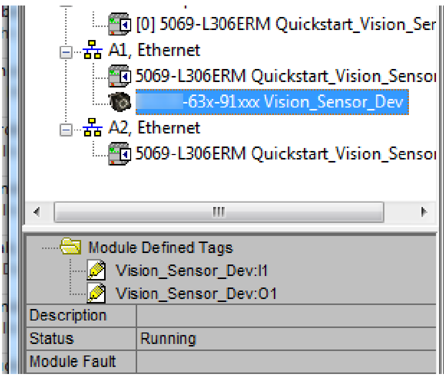



*Fig. 70: Status "Running"*

## 10.5 Mapping of output data

The input data is assigned as follows: (select module CNTL + STAT + Data (2+128))

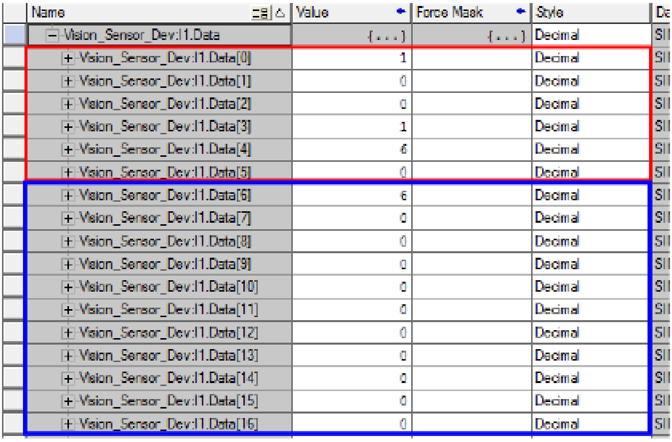
…. I1.Data[0] – …. I1.data [5] "Status" module (see description Module 2: "Status" (from VISOR® to PLC) (Page 33))

e.g. ... I1.Data [3] = Job number

…. I1.Data[4] = Image_ID

The data module is appended directly. Start of Data module from ... I1.Data[6] - …. I1.Data[135]

Here the data is inserted as indicated in SensoConfig under "Output" / "Telegram".
Additional information: Defining telegrams / data output in SensoConfig (Page 16)



| Name | Value | Force Mask | Style | De |
|---|---|---|---|---|
| Vision_Sensor_Dev:I1.Data | {...} | {...} | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[0] | 1 | | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[1] | 0 | | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[2] | 0 | | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[3] | 1 | | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[4] | 6 | | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[5] | 0 | | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[6] | 6 | | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[7] | 0 | | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[8] | 0 | | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[9] | 0 | | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[10] | 0 | | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[11] | 0 | | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[12] | 0 | | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[13] | 0 | | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[14] | 0 | | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[15] | 0 | | Decimal | SI |
| Vision_Sensor_Dev:I1.Data[16] | 0 | | Decimal | SI |

*Fig. 71: Output data*

### Conversion of binary values

All detector-specific payloads with decimal places will be transmitted as integers multiplied by 1000, and accordingly must be divided by 1000 after the data is received. The values are transferred in the format "Big-endian". The length is based on the value, e.g., score 32 bits (DWord).

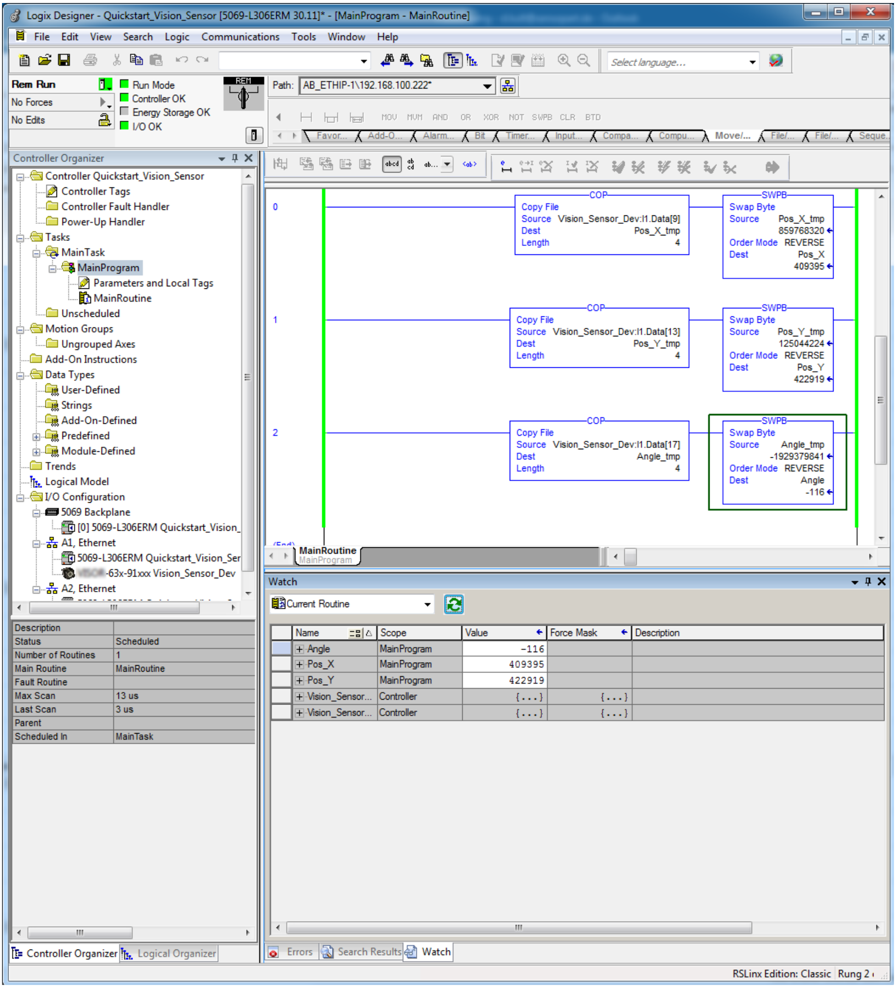
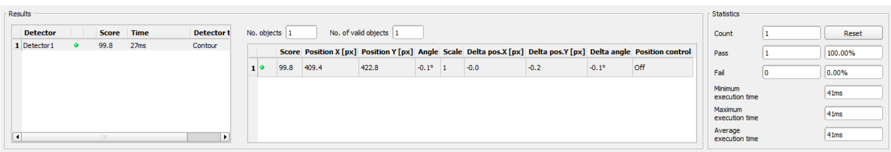



*Fig. 72: Swapping the byte order*



## 10.6 PLC example programs

The following PLC example programs show some basic functions.

### PLC example 1: Trigger when VISOR® Ready

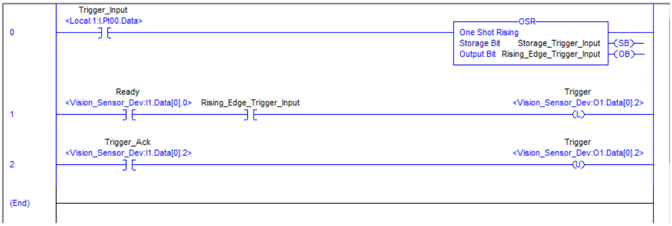



*Fig. 73: Trigger when VISOR® Ready, (without error handling)*

### PLC example 2: Job change when VISOR® Ready

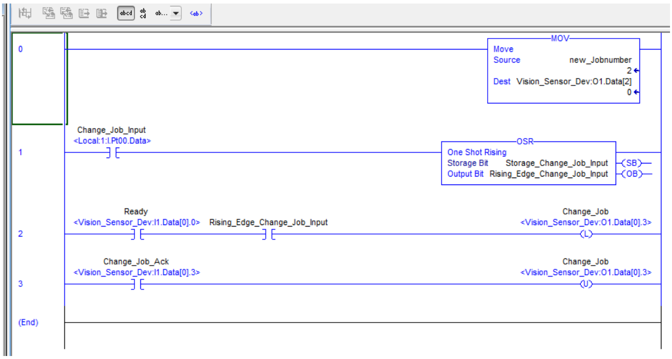



*Fig. 74: Job change when VISOR® Ready, (without error handling)*

### PLC example 3: Switch to Run when VISOR® in configuration mode

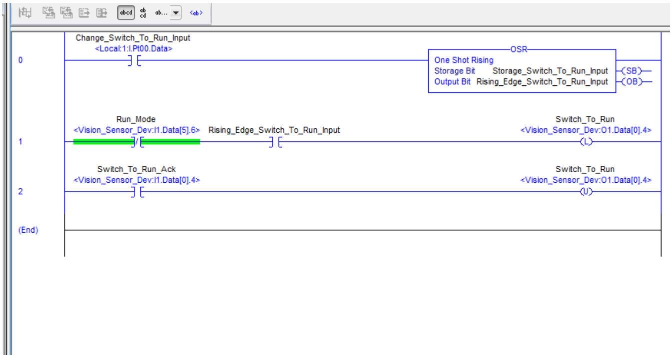



*Fig. 75: Switch to Run when VISOR® in configuration mode (without error handling)*

# 11 Telegrams and data output

This section describes the telegrams available for the VISOR® vision sensor. These telegrams can be sent to the VISOR® vision sensor through various interfaces.

- Ethernet TCP/IP
- PROFINET (Request / Response module)

The telegrams are available in ASCII and Binary format. The format is defined in the module "SensoConfig", in the tab "Telegram" of the setup "Output".

The following settings are possible:

| Communication | TCP / IP | EtherNet/IP | PROFINET |
|---|---|---|---|
| **Telegram format** | ASCII / Binary | Binary | Binary |

## 11.1 Overview telegrams

Telegrams: Availability and supported interfaces (Page 91)

### VISOR® General

- **Reset statistics (RST)** (ASCII / Binary)
  The "Reset statistics" telegram can be used to reset the VISOR® vision sensor's internal statistics counter.

### VISOR® Control

- **Trigger (TRG)** (ASCII / binary)
  With the telegram "Trigger", an image can be acquired. Some commands need additional image acquisition. The result data of the evaluation are output via the "Out" port.
- **Extended trigger (TRX)** (ASCII / binary)
  This telegram "Extended trigger" is an expansion of the "trigger" telegram. Besides the result data, there is also the option to assign an ID or to receive information about the operating mode (run/config). Unlike the "trigger" telegram, the result data of the "Extended trigger" telegram are also transferred via the "In" port.
- **Trigger Robotics (TRR)** (ASCII / Binary)
  With the telegram "Trigger", an image can be acquired. In addition to image acquisition, the robot tool center point (TCP) can be transferred. The TCP is used to calculate the position values.

- **Set Trigger ID (STI)** (ASCII / Binary)
  With the telegram "Set Trigger ID", a Trigger ID can be set. Der Identifier wird für die nächste Bildaufnahme verwendet und kann bspw. als Dateiname gesetzt werden.
- **Job change (CJB)** (ASCII / binary)
  The "Job change" telegram will trigger a job change on the VISOR® vision sensor.
- **Job change permanent (CJP)** (ASCII / Binary)
  The "Job change permanent" telegram will trigger a permanent job change on the VISOR® vision sensor. The job is run again after restarting.
- **Job change by job name (CJN)** (ASCII / Binary)
  The "Job change by job name" telegram will trigger a job change on the VISOR® vision sensor. The job will be run by job name. You can read the job names by using the "Read job list" telegram, for example.

**VISOR® Job settings**

- **Auto Working distance (AFC)** (ASCII / Binary)
  The "Auto operating distance" telegram can be used to have the working distance for the job be automatically determined.
- **Set working distance (SFC)** (ASCII / Binary)
  The "Set working distance" telegram can be used to change the working distance for the job.
- **Read working distance (GFC)** (ASCII / binary)
  The "Read working distance" telegram can be used to read the current working distance for the job.
- **Auto shutter speed (ASH)** (ASCII / Binary)
  The "Auto shutter speed" telegram can be used to have the shutter time for the job be automatically determined.
- **Set shutter speed (SSP/SST)** (ASCII / Binary)
  With the telegram "Set shutter speed", the shutter speed of the job can be changed. This telegram can, for example, be used for brightness compensation.
- **Read shutter speed (GSH)** (ASCII / Binary)
  With the telegram "Read shutter speed", the set shutter speed of the job can be read.
- **Set gain (SGA)** (ASCII / binary)
  With the telegram "Set gain", the gain of the job can be changed. This telegram can, for example, be used for brightness compensation.
- **Read gain (GGA)** (ASCII / binary)
  With the telegram "Read gain", the set gain of the job can be read.
- **Set parameters (SPP/SPT)** (ASCII / binary)
  With the telegram "Set parameter", the detector parameters can be adjusted, e.g. reference strings, detector thresholds.

- **Read parameter (GPA)** (ASCII / binary)
  With the telegram "Read parameter", the set parameters of the detectors can be read.
- **Set search range / ROI (SRP/SRT)** (ASCII / binary)
  With the telegram "Set ROI", the position of the selected detector can be changed.
- **Read search range / ROI (GRI)** (ASCII / Binary)
  With the telegram "Read ROI", the position of the selected detector can be read.
- **Read job list (GJL)** (ASCII / binary)
  The "Get job list" telegram can be used to output a list of all available jobs on the VISOR® vision sensor.
- **Read detector list (GDL)** (ASCII / binary)
  With the telegram "Read detector list", a list of all detectors in the current job will be displayed.
- **Teach-in detector (TED)** (ASCII / binary)
  The "Teach detector" telegram will result in the specified detector being re-taught (available only for Pattern matching and Contour).
- **Set trigger delay (STD)** (ASCII / Binary)
  With the telegram "Set trigger delay", a delay for starting a trigger can be set (in time (ms) or encoder steps).
- **Read trigger delay (GTD)** (ASCII / Binary)
  With the telegram "Read trigger delay", the set delay for starting a trigger can be read.
- **Save Job Permanently (SJP)** (ASCII / binary)
  The "Save job permanently" telegram will take all the parameters that were previously set temporarily and copy them to a job set.

**VISOR® Calibration**

- **Calibration: Initialize (CCD)** (ASCII / binary)
  The point pair list is initialized with the telegram "Calibration: Initialize point pair list".
- **Calibration: Add world point (CAW)** (ASCII / binary)
  With the telegram "Calibration: Add world point" a world point (fiducial or point pair) is added to the point pair list. The telegram can be used for the calibration method Point pair list (Robotics) and Calibration plate (Robotics).
- **Calibration: Point pair list (CCL)** (ASCII / binary)
  With the telegram "Calibration: Point pair list" the calibration is carried out using the point pair list in the current job.
- **Calibration: Validate point pair list** (ASCII / binary)
  With the telegram "Calibration: Validate point list", the calibration is validated using the point list.

- **Calibration: Calibration Plate (CCP)** (ASCII / Binary)
  With the telegram "Calibration: Calibration plate", the calibration is carried out using the calibration plate.
- **Set fiducial (CSF)** (ASCII / binary)
  With the telegram "Set fiducial", the fiducials are set using the point list in the current job.
- **Calibration: Add Image (CAI)** (ASCII / Binary)
  The "Add image" telegram triggers an image acquisition and if a calibration plate is found, an image is added to the calibration object. The telegram can be used for calibration method Multi-image calibration and calibration method Calibration plate (Robotics).
- **Calibration: Multi-image (CMP)** (ASCII / binary)
  With the telegram "Calibration: Multi-image" a calibration is carried out and an existing calibration object is accessed.
- **Calibration: Robotics Multi-image (CRP)** (ASCII / Binary)
  With the telegram "Multi-image, robot" a calibration is carried out using the calibration plate.
- **Calibration: Copy calibration (CCC)** (ASCII / binary)
  With the telegram "Calibration: Copy calibration", the calibration of the current job is copied to the selected destination.
- **Calibration: Set parameters (CSP)** (ASCII / binary)
  With the telegram "Calibration: Set parameter", the parameter values for the calibration can be set.
- **Calibration: read parameters (CGP)** (ASCII / binary)
  With the telegram "Calibration: Read parameter", the set parameter values of the calibration can be read.

**VISOR® Visualization**

- **Get image (GIM)** (ASCII / binary)
  The "Get image" telegram can be used to get the image from the VISOR® vision sensor.

**VISOR® Service (available only on port 1998 and in ASCII format)**

- **Update visualization data (UVR)** (ASCII)
  The "Update visualization data" telegram is used to update visualization data such as image, detector information and results.
- **Read sensor identity (GSI)** (ASCII)
  With the telegram "Read sensor identity", the current firmware status as well as the hardware type can be queried.

- **Update firmware (UFW)** (ASCII)
  With the telegram "Update firmware", a firmware update is started. The firmware file must first be loaded onto the VISOR® vision sensor.
- **Read jobset (SJS)** (ASCII)
  The "Set job set" telegram can be used to change the VISOR® vision sensor's job set. The job set file must first be loaded onto the VISOR®.
- **Save jobset (GJS)** (ASCII)
  The "Save jobset" telegram can be used to read the VISOR® vision sensor's job set.

### Data output

This section contains information about the data output (e.g. which format the individual results will have).

Data output ASCII

- General
- Base values
- Position
- Measurement
- Identification
- Identification - quality
- Color
- Counting / number
- Extended

Data output Binary

- General
- Base values
- Position
- Measurement
- Identification
- Identification - quality
- Color
- Counting / number
- Extended

## 11.2 Telegrams: Availability and supported interfaces

**Device variants**

| | | | | | |
|---|---|---|---|---|---|
| **ALL** | Allround | **S** | Standard | | |
| **OB** | Object | **A** | Advanced | | |
| **CR** | Code reader | **P** | Professional | | |
| **RO** | Robotic | | | | |
| ✓ | available | | | | |
| [ ] | Limited availability: differences between versions < 2 and ≥ 2 | | | | |

**Interfaces**

| | |
|---|---|
| **1** | Ethernet TCP IN (2006) |
| **2** | PROFINET |
| **3** | EtherNet/IP |
| **4** | Service Port (1998) |

| Telegram | ALL | | OB | | CR | | | RO | | Interfaces | | | | From version |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | P | S | A | S | A | P | A | P | 1 | 2 | 3 | 4 | |
| **VISOR® General** | | | | | | | | | | | | | | |
| Reset statistics (RST) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.18 |
| **VISOR® Control** | | | | | | | | | | | | | | |
| Trigger (TRG) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |
| Extended trigger (TRX) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | 1.6 |
| Trigger Robotics (TRR) | | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.2 |
| Set Trigger ID (STI) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.2 |
| Job change (CJB) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |
| Job Change Permanent (CJP) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.18 |
| Job change by name (CJN) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.0 |
| **VISOR® Job settings** | | | | | | | | | | | | | | |
| Auto working distance (AFC) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.0 |
| Set working distance (SFC) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.0 |
| Read working distance (GFC) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.0 |
| Auto Shutter Speed (ASH) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.0 |
| Set Shutter Speed (SSP/SST) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |
| Read shutter speed (GSH) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |
| Set gain (SGA) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.6 |
| Read gain (GGA) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.6 |
| Set Parameter (SPP/SPT) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |
| Read Parameter (GPA) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |

| Telegram | ALL | | OB | | CR | | | RO | | Interfaces | | | | From version |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | P | S | A | S | A | P | A | P | 1 | 2 | 3 | 4 | |
| Set ROI (SRP/SRT) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |
| Read ROI (GRI) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |
| Read job list (GJL) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.18 |
| Read Detector List (GDL) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.18 |
| Teach detector (TED) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |
| Set trigger delay (STD) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.22 |
| Read Trigger Delay (GTD) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.22 |
| Save Job Permanently (SJP) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.0 |
| **VISOR® Calibration** | | | | | | | | | | | | | | |
| Initialization (CCD) | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.18 |
| Add world point (CAW) | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.22 |
| Calibration: Point Pair List (CCL) | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.18 |
| Validate calibration (CVL) | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.18 |
| Calibration: Calibration Plate (CCP) | [] | ✓ | | [] | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.19 |
| Set fiducials (CSF) | | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.22 |
| Add image (CAI) | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.2 |
| Multi-Image (CMP) | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.2 |
| Robotics Multi-Image (CRP) | | ✓ | | | | | | | ✓ | ✓ | ✓ | ✓ | | 2.2 |
| Copy calibration (CCC) | | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.19 |
| Set parameters (CSP) | [] | ✓ | | [] | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.22 |
| Read parameters (CGP) | [] | ✓ | | [] | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.22 |
| **VISOR® Visualization** | | | | | | | | | | | | | | |
| Get Image (GIM) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | | | 1.0 |

| Telegram | ALL | | OB | | CR | | | RO | | Interfaces | | | | From version |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | A | P | S | A | S | A | P | A | P | 1 | 2 | 3 | 4 | |
| **VISOR® Service** | | | | | | | | | | | | | | |
| Update visualization data (UVR) | ✓ | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | | | | ✓ | 1.22 |
| Read sensor identity (GSI) | ✓ | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | | | | ✓ | 1.19 |
| Update firmware (UFW) | ✓ | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | | | | ✓ | 1.19 |
| Read job set (SJS) | ✓ | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | | | | ✓ | 1.19 |
| Save job set (GJS) | ✓ | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | | | | ✓ | 1.19 |

Please refer to the following as well: Overview telegrams (Page 87)

## 11.3 Error codes

| Error code | Error code HEX | Description |
|---|---|---|
| 000 | 0x00 | Successful |
| 001 | 0x01 | Error |
| 003 | 0x03 | Invalid parameter data |
| 005 | 0x05 | Invalid telegram |
| 006 | 0x06 | Input parameters with invalid size or invalid value |
| 007 | 0x07 | File does not exist |
| 008 | 0x08 | Recorder off |
| 009 | 0x09 | Matching image of requested type not found |
| 010 | 0x0A | Invalid file name or length |
| 011 | 0x0B | Invalid data length |
| 012 | 0x0C | Not allowed due to job set mismatch |
| 013 | 0x0D | Failed to start new job from job set |
| 016 | 0x10 | Firmware version mismatch |
| 018 | 0x12 | Calibration plate data not available |
| 020 | 0x14 | More than one vis file present |
| 021 | 0x15 | Sensor type does not match for vis file |
| 030 | 0x1E | Calibration not activated / Calibration not supported |

| Error code | Error code HEX | Description |
|---|---|---|
| 031 | 0x1F | Calibration copy error |
| 032 | 0x20 | Mismatched input conditions for destination job |
| 033 | 0x21 | Calibration / validation error |
| 034 | 0x22 | Invalid number of points |
| 035 | 0x23 | Calibration error: Add point, e.g. last job result failed |
| 036 | 0x24 | Invalid fiducial |
| 037 | 0x25 | Job set protection error: "Permanent" job change is not allowed |
| 038 | 0x26 | Parameter values are not available to write / read |
| 039 | 0x27 | Sensor is in configuration mode. The telegram was rejected |
| 040 | 0x28 | Write / read error for parameter value |
| 041 | 0x29 | No matching job found |
| 042 | 0x2A | Formatting error |
| 043 | 0x2B | Job set / Job saving error |
| 044 | 0x2C | Focus lock time exceeded |
| 045 | 0x2D | Error with multiple files |
| 046 | 0x2E | Working distance could not be determined |
| 047 | 0x2F | "Min. processing time per image" was not observed |
| 048 | 0x30 | Search range size (ROI) does not match |
| 049 | 0x31 | Search range (ROI) Freeform not selected |
| 050 | 0x32 | Calibration method does not match |
| 051 | 0x33 | No calibration plate found |
| 052 | 0x34 | Number of images too small |
| 053 | 0x35 | No calibration possible: distance between tool positions not plausible |
| 054 | 0x36 | Rotation between images not sufficient |
| 055 | 0x37 | Tilt between the images not sufficient |

## 11.4 Description Telegrams ASCII

### 11.4.1 General

#### Reset statistics (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Reset Statistics (RST) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **R** | Reset statistics |
| 2 | **S** | |
| 3 | **T** | |
| Example: | RST | |
| **Reset Statistics (RST) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **R** | Reset statistics |
| 2 | **S** | |
| 3 | **T** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| Example: | RSTP | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## 11.4.2 Control

### Trigger (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Trigger (TRG) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **T** | Trigger (simple trigger, in-port) |
| 2 | **R** | |
| 3 | **G** | |
| Example: | TRG | |
| **Trigger (TRG) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **T** | Trigger (response to command trigger without index, via port 2006. If defined: Result data without index via port 2005) |
| 2 | **R** | |
| 3 | **G** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| Example: | TRGP | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | Yes |
| Accepted when Ready is low: | | No |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Extended trigger (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Extended Trigger (TRX) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **T** | Extended trigger, (trigger with index, for correlation of trigger to corresponding result data, via port 2006) |
| 2 | **R** | |
| 3 | **X** | |
| 4 - 5 | X | Length of following data (n) |
| 6 ... n | X | Data |
| Example: | TRX06MyPart | |
| **Extended Trigger (TRX) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **T** | Extended trigger, (response to trigger with index and result data, via port 2006, for correlation of trigger to corresponding result. Result data without index via port 2005) |
| 2 | **R** | |
| 3 | **X** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 6 | X | Length of following data (n) |
| 7 ... n | X | Data of sending command |
| n+1 | C<br>R | C = Config<br>R = Run |
| n+2 ... n+9 | X | Length of following result data (n) |
| n+9 ... m | X | Result data |
| Example: | TRX06MyPartR00000000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | Yes |
| Accepted when Ready is low: | | No |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Trigger Robotics (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Trigger Robotics (TRR) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **T** | Trigger Robotics |
| 2 | **R** | |
| 3 | **R** | |
| 4 | 1 | Request version |
| 5-6 | X | Length of trigger identifier |
| 7-n | X | Trigger Identifier |
| n+1...n+8 | X | Pose_TCP Pos. X (in user unit * 1000) |
| n+9...n+16 | X | Pose_TCP Pos. Y (in user unit * 1000) |
| n+17...n+24 | X | Pose_TCP Pos. Z (in user unit * 1000) |
| n+25...n+32 | X | Pose_TCP Angle X (in degrees * 1000) |
| n+33...n+40 | X | Pose_TCP Angle Y (in degrees * 1000) |
| n+41...n+48 | X | Pose_TCP Angle Z (in degrees * 1000) |
| Example: | TRR104Part000040040000500500006006000070070000 800800009009 | |
| **Trigger Robotics (TRR) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **T** | Trigger (response to command trigger without index, via port 2006. If defined: Result data without index via port 2005) |
| 2 | **R** | |
| 3 | **R** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5-7 | X | Error codes (Page 94) |
| 7-8 | X | Length of trigger identifier |

| | | |
|---|---|---|
| 9-n | X | Trigger Identifier |
| n+1 | X | Operation Mode<br>C = Config<br>R = Run |
| n+2...n+9 | X | Length of result data |
| n+10...m | X | Result data |
| Example: | TRRP00004PartR00000000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | Yes |
| Accepted when Ready is low: | | No |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | |

Note: For "Calibration plate (Robotics)" and "Point pair list (Robotics)" only the X and Y position are taken into account. The other values (position Z and rotations) must be 0.

## Set Trigger ID (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Set Trigger ID (STI) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **S** | Set Trigger ID |
| 2 | **T** | |
| 3 | **I** | |
| 4 | 1 | Request version |
| 5-6 | x | Length of the following data (max 99) |
| 7-n | x | Trigger ID |
| Example: | STI106MyPart | |
| **Set Trigger ID (STI) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **S** | Set Trigger ID |
| 2 | **T** | |
| 3 | **I** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5-7 | x | Error codes (Page 94) |
| Example: | STIP000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | Yes |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | |

## Job change (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Job change (CJB) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **C** | Job change |
| 2 | **J** | |
| 3 | **B** | |
| 4 - 6 | X | Job number |
| Example: | CJB005 | |
| **Job change (CJB) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **C** | Job change |
| 2 | **J** | |
| 3 | **B** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 | T<br>F | Triggered<br>Freerun |
| 6 - 8 | X | Job number |
| Example 1: | CJBPT005 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

**NOTE:**
If an error occurs during the job change, it is possible to change to Job 1.

## Job Change Permanent (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Job Change Permanent (CJP) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **C** | Job change permanent (Change Job Permanently) |
| 2 | **J** | |
| 3 | **P** | |
| 4 - 6 | X | Job number |
| Example: | CJP005 | |
| **Job Change Permanent (CJP) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **C** | Job change permanent (Change Job Permanently) |
| 2 | **J** | |
| 3 | **P** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 | T<br>F | Triggered<br>Freerun |
| 6 - 8 | X | Job number |
| Example 1: | CJPPT005 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

**NOTE:**
If an error occurs during the job change, it is possible to change to Job 1.

## Job change by job name (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Job change by job name (CJN) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **C** | Job change by name |
| 2 | **J** | |
| 3 | **N** | |
| 4 | 1 | Request version |
| 5 - 7 | X | Job name length |
| 8 - n | X | Job name |
| Example: | CJN1005Myjob | |
| **Job change by job name (CJN) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **C** | Job change by name |
| 2 | **J** | |
| 3 | **N** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 | X | Trigger mode<br>T: Trigger<br>F: Free run |
| Example: | CJNP000T | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

### 11.4.3 Job settings

## Auto working distance (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Auto working distance (AFC) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **A** | Auto Working Distance (Auto Focus) |
| 2 | **F** | |
| 3 | **C** | |
| 4 | 1 | Request version |
| 5 | X | 0: Temporary<br>1: Permanent |
| 6 | X | Step size 1-5 |
| 7 - 9 | X | Focus selection<br>0: Maximum score<br>1: Min. Arbeitsabstand<br>2: Max. working distance<br>3: Average working distance<br>4: Median working distance<br>5: Maximum score and all planes |
| 10 | X | Focus unit<br>0: Millimeters<br>1: Steps |
| 11 | X | Working distance selection<br>0: Default range<br>1: Specified range |
| | **NOTE:**<br>The following byte sequence is only relevant if "Distance range selection" has been set to 1. | |
| 12 - 19 | X | Start of working area (close) |
| 20 - 27 | X | End of working area (far) |
| Example: | Example 1: AFC11100500<br>Example 2: AFC1110050100010000000100000 | |
| **Auto working distance (AFC) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |

| | | |
|---|---|---|
| 1 | **A** | Auto Working Distance (Auto Focus) |
| 2 | **F** | |
| 3 | **C** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 10 | X | Focus selection = 5 ; Number of Focus selection distances found = 1-4 ; 1 |
| | **NOTE:**<br>The following fields [Distance value / Score value] are repeated for each number of distances found. | |
| 11 - 18 | X | Distance value in mm *1000 or in steps |
| 19 - 26 | X | Score value in %*1000 |
| Example: | AFCP000002000000950000009000093000000089000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Set working distance (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Set working distance (SFC) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **S** | Working distance (Set Focus) |
| 2 | **F** | |
| 3 | **C** | |
| 4 | 1 | Request version |
| 5 | X | 0: Temporary<br>1: Permanent |
| 6 | X | Movement<br>0: Absolute<br>1: Relative<br>2: Absolute with reinitialization |
| 7 | X | Unit<br>0: 1/1000 millimeters<br>4: Steps |
| 8 - 15 | X | Distance value in mm * 1000 or in steps |
| Example: | SFC111400000010 | |
| **Set working distance (SFC) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **S** | Working distance (Set Focus) |
| 2 | **F** | |
| 3 | **C** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 15 | X | Distance value in mm * 1000 or in steps |
| Example: | SFCP00000000050 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |

| | |
|---|---|
| Status of Ready signal during processing: | No change |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | Max. 4 bytes (optional) |

## Read working distance (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Read working distance (GFC) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **G** | Read working distance (Get Focus) |
| 2 | **F** | |
| 3 | **C** | |
| 4 | 1 | Request version |
| 5 | X | Unit<br>0 - 1/1000 millimeters<br>4 - steps |
| Example: | GFC10 | |
| **Read working distance (GFC) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **G** | Read working distance (Get Focus) |
| 2 | **F** | |
| 3 | **C** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 15 | X | Distance value in mm *1000 or in steps |
| Example: | GFCP00000092500 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Auto shutter speed (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Auto shutter speed (ASH) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **A** | Auto shutter speed |
| 2 | **S** | |
| 3 | **H** | |
| 4 | 1 | Request version |
| 5 | X | 0: Temporary<br>1: Permanent |
| Example: | ASH11 | |
| **Auto shutter speed (ASH) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **A** | Auto shutter speed |
| 2 | **S** | |
| 3 | **H** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 15 | X | Auto Shutter speed value in ms * 1000 |
| 16 - 23 | X | Score in % * 1000 |
| Example: | ASHP0000000178000057500 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Set shutter speed (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Set shutter speed (SSP/SST) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **S** | Set Shutter Speed |
| 2 | **S** | |
| 3 | **P**<br>**T** | Permanent<br>Temporary |
| 4 - 5 | X | Number of digits of the shutter speed value, e.g. 04 |
| 6 - 9 | X | New shutter speed value in ms * 1000<br>e.g. 8000 = 8 ms |
| Example: | SSP048000 | |
| **Set shutter speed (SSP/SST) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **S** | Set Shutter Speed |
| 2 | **S** | |
| 3 | **P**<br>**T** | Permanent<br>Temporary |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| Example: | SSPP | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Read shutter speed value (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Read Shutter Speed Value (GSH) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **G** | Read Shutter Speed value (Get Shutter) (from active job) |
| 2 | **S** | |
| 3 | **H** | |
| Example: | GSH | |
| **Read Shutter Speed Value (GSH) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **G** | Read Shutter Speed Value (Get Shutter) |
| 2 | **S** | |
| 3 | **H** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 | X | Shutter speed value, length |
| 6 ... n | X | Shutter speed value in ms * 1000 |
| Example Run Mode: | GSHP41200 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Set gain (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Set gain (SGA) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **S** | Set Gain |
| 2 | **G** | |
| 3 | **A** | |
| 4 | X | 0: Temporary<br>1: Permanent |
| 5 - 9 | X | New gain value (in value * 1000), e.g. 2.0 = 02000 |
| Example: | SGA102000 | |
| **Set gain (SGA) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **S** | Set Gain |
| 2 | **G** | |
| 3 | **A** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 9 | X | Current gain value * 1000 |
| Example: | SGAP02000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Read gain value (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Read gain value (GGA) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **G** | Read gain value (Get Gain) |
| 2 | **G** | |
| 3 | **A** | |
| Example: | GGA | |
| **Read gain value (GGA) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **G** | Read gain value (Get Gain) |
| 2 | **G** | |
| 3 | **A** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 9 | X | Current gain value (value *1000), e.g. 1.0 = 01000 |
| Example: | GGAP01000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Set parameter (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Set parameters (SPP/SPT) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **S** | Set parameters |
| 2 | **P** | |
| 3 | **P**<br>**T** | P Permanent<br>T Temporary |
| 4 - 6 | X | Detector number |
| 7 - 9 | X | Command: Parameter number, see below, table Overview detector parameters |
| 10 - 14 | X | Length of value (max. 512 bytes) |
| 15 ... n | X | Value |
| Example: | SPP0010010000560000 | |
| **Set parameters (SPP/SPT) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **S** | Set parameters |
| 2 | **P** | |
| 3 | **P**<br>**T** | P Permanent<br>T Temporary |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |

| | | |
|---|---|---|
| 5 - 8 | X | SI08 - Signed Integer 08<br>UI08 - Unsigned Integer 08<br>SI16 - Signed Integer 16<br>UI16 - Unsigned Integer 16<br>SI32 - Signed Integer 32<br>UI32 - Unsigned Integer 32<br>SI40 - Signed Integer 40<br>UI40 - Unsigned Integer 40<br>FLOT - Float<br>DOBL - Double<br>STRG - String<br>BOOL - Boolean<br>SP08 - Special Signed 8<br>UDEF - Undefined<br>IARR - Integer Array<br>ZERO - Default Zero Parameter |
| Example: | SPPPSTRG | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

### Overview Detector Parameters (set / read)

| Detector | Function | Value | Multiplier | Length |
|---|---|---|---|---|
| **Alignment** | | | | |
| Pattern matching<br>Contour matching | Threshold value Min. | 1 | 1000 | n |
| | Threshold value Max. | 2 | 1000 | n |
| | Result offset<br>0: "Off"<br>1: "Image plane (in pixels)"<br>2: "Align (2D)"<br>3: "Robot (3D)" | 30 | 1 | n |
| | Result offset Image plane: Pos. X | 31 | 1000 | n |
| | Result offset Image plane: Pos. Y | 32 | 1000 | n |
| | Result offset Image plane: angle | 33 | 1000 | n |
| | Result offset<br>Align (2D), Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z | 34 | 1000 | 48 (6 * 8 bytes per value) |
| | Calculate Result offset* with transmitted position<br>• Align (2D): Pos. X, Pos. Y, 0, 0, 0, Angle Z<br>• Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z<br>*A valid position for the detector must be available | 35 | 1000 | 48 (6 * 8 bytes per value) |
| Edge detector | Probe 1: Transition<br>0: Any<br>1: Dark to light<br>2: Light to dark | 101 | 1 | n |
| | Probe 2: Transition<br>0: Any<br>1: Dark to light<br>2: Light to dark | 102 | 1 | n |

| Detector | Function | Value | Multiplier | Length |
|---|---|---|---|---|
| | Probe 3: Transition<br>0: Any<br>1: Dark to light<br>2: Light to dark | 103 | 1 | n |
| | Probe 1: Threshold value Min. | 104 | 1000 | n |
| | Probe 2: Threshold value Min. | 105 | 1000 | n |
| | Probe 3: Threshold value Min. | 106 | 1000 | n |
| **Detector** | | | | |
| Pattern matching<br>Contour<br>Contour 3D | Threshold value Min. | 1 | 1000 | n |
| | Threshold value Max. | 2 | 1000 | n |
| | Result offset<br>0: "Off"<br>1: "Image plane (in pixels)"<br>2: "Align (2D)"<br>3: "Robot (3D)" | 30 | 1 | n |
| | Result offset Image plane: Pos. X | 31 | 1000 | n |
| | Result offset Image plane: Pos. Y | 32 | 1000 | n |
| | Result offset Image plane: angle | 33 | 1000 | n |
| | Result offset<br>Align (2D), Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z | 34 | 1000 | 48 (6 * 8 bytes per value) |
| | Calculate Result offset* with transmitted position<br>• Align (2D): Pos. X, Pos. Y, 0, 0, 0, Angle Z<br>• Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z<br>*A valid position for the detector must be available | 35 | 1000 | 48 (6 * 8 bytes per value) |
| Gray | Threshold value Min. | 1 | 1000 | n |
| | Threshold value Max. | 2 | 1000 | n |
| | Grayscale value Min. | 101 | 1000 | n |

| Detector | Function | Value | Multiplier | Length |
|---|---|---|---|---|
| | Grayscale value Max. | 102 | 1000 | n |
| | Invert grayscale value | 103 | 1 | n |
| Contrast Brightness | Threshold value Min. | 1 | 1000 | n |
| | Threshold value Max. | 2 | 1000 | n |
| Caliper | Threshold value Distance Min. | 101 | 1000 | n |
| | Threshold value Distance Max. | 102 | 1000 | n |
| | Invert distance threshold value<br>0: not inverted<br>1: inverted | 103 | 1 | 1 |
| | Distance mode<br>0: Minimum<br>1: Maximum<br>2: Mean<br>3: Median<br>4: Smallest opposite<br>5: Largest opposite | 104 | 1 | n |
| | Probe 1: Threshold value Min. | 105 | 1000 | n |
| | Probe 2: Threshold value Min. | 106 | 1000 | n |
| | Probe 1: Smoothing | 107 | 1000 | n |
| | Probe 2: Smoothing | 108 | 1000 | n |
| | Probe 1: Transition<br>0: Any<br>1: Dark to light<br>2: Light to dark | 109 | 1 | n |
| | Probe 2: Transition<br>0: Any<br>1: Dark to light<br>2: Light to dark | 110 | 1 | n |
| | Probe 1: Number of search stripes | 111 | 1 | n |
| | Probe 2: Number of search stripes | 112 | 1 | n |
| BLOB | Grayscale value Min. | 101 | 1000 | n |
| | Grayscale value Max. | 102 | 1000 | n |
| | Invert grayscale value<br>0: not inverted<br>1: inverted | 103 | 1 | 1 |

| Detector | Function | Value | Multiplier | Length |
|---|---|---|---|---|
| | Threshold value Number of BLOBs Min. | 120 | 1 | n |
| | Threshold value Number of BLOBs Max. | 121 | 1 | n |
| | Invert number threshold value<br>0: not inverted<br>1: inverted | 122 | 1 | 1 |
| | Number of set features (read only) | 123 | 1 | n |
| | Selection of a feature from the list | 124 | 1 | n |
| | Feature threshold value Min. | 125 | 1000 | n |
| | Feature threshold value Max. | 126 | 1000 | n |
| | Invert feature threshold value | 127 | 1 | 1 |
| Barcode<br>Datacode<br>OCR | Reference string | 101 | - | n (length of string) |
| Color Value<br>Color Area | Color space (read only) | 21 | 0 = RGB<br>1 = HSV<br>2 = LAB | 3 |
| | Channel selection (read only) | 22 | Bit field one digit per color channel | 4 |
| | Color channel 1: Threshold value Min. | 101 | 1000 | n |
| | Color channel 1: Threshold value Max. | 102 | 1000 | n |
| | Color channel 1: Invert threshold value | 103 | 1 | n |
| | Color channel 2: Threshold value Min. | 104 | 1000 | n |
| | Color channel 2: Threshold value Max. | 105 | 1000 | n |
| | Color channel 2: Invert threshold value | 106 | 1 | n |
| | Color channel 3: Threshold value Min. | 107 | 1000 | n |

| Detector | Function | Value | Multiplier | Length |
| --- | --- | --- | --- | --- |
| | Color channel 3: Threshold value Max. | 108 | 1000 | n |
| | Color channel 3: Invert threshold value | 109 | 1 | n |
| Color List | Color space (read only) | 21 | 0 = RGB<br>1 = HSV<br>2 = LAB | 3 |
| | Channel selection (read only) | 22 | Bit field one digit per color channel | 4 |
| | Color distance threshold value | 101 | 1000 | n |
| | Set color distance threshold value active | 102 | 1 | n |
| | Number of colors in list | 103 | 1 | n |
| | Selection of a color from the list | 104 | 1 | n |
| | Color value of the selected color (color channel 1, color channel 2, color channel 3, color channel 4 [constantly 0]) | 105 | 1000 | 32 |
| Busbar<br>Wafer | Threshold value Min. | 1 | 1000 | n |
| | Threshold value Max. | 2 | 1000 | n |
| Result processing | Name of the active expression | 122 | - | n (length of string) |
| | Current expression | 124 | - | n (length of string) |

## Read parameter (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Read parameter (GPA) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **G** | Read parameter (Get Parameter) |
| 2 | **P** | |
| 3 | **A** | |
| 4 - 6 | X | Detector number<br>e.g. 001 |
| 7 - 9 | X | Command: Parameter number, see table Overview detector parameters |
| Example: | GPA001001 | |
| **Read parameter (GPA) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **G** | Read parameter (Get Parameter) |
| 2 | **P** | |
| 3 | **A** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 8 | X | SI08 - Signed Integer 08<br>UI08 - Unsigned Integer 08<br>SI16 - Signed Integer 16<br>UI16 - Unsigned Integer 16<br>SI32 - Signed Integer 32<br>UI32 - Unsigned Integer 32<br>SI40 - Signed Integer 40<br>UI40 - Unsigned Integer 40<br>FLOT - Float<br>DOBL - Double<br>STRG - String<br>BOOL - Boolean<br>SP08 - Special Signed 8<br>UDEF - Undefined<br>IARR - Integer Array<br>ZERO - Default Zero Parameter |
| 9 - 13 | X | Length of value (n)<br>e.g. 00005 |

| | | |
|---|---|---|
| 14 ... n | X | Value |
| Example: | GPAPSTRG00005Test1 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Set search range (ROI) (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Set ROI (SRP/SRT) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **S** | Set search range (Set ROI) |
| 2 | **R** | |
| 3 | **P**<br>**T** | P = Permanent<br>T = Temporary |
| 4 - 11 | X | ROI Info Length in bytes, from byte 5 to end<br>39 Byte: circle<br>55 bytes: rectangle, ellipse, free shape |
| 12 - 14 | X | Detector no.<br>e.g. 001 |
| 15 - 16 | X | ROI Index<br>00: for yellow search range<br>01: for red teach range<br>02: Position control |
| 17 - 18 | X | ROI shape<br>01: Circle<br>02: Rectangle<br>03: Ellipse<br>04: Free shape |
| 19 - 26 | X | Center X (value in pixels * 1000), e.g. 160 pixels = 00160000 |
| 27 - 34 | X | Center Y (value in pixels * 1000), e.g. 120 pixels = 00120000 |
| 35 - 42 | X | Half width / X-radius (value in pixels * 1000), e.g. 80 pixels = 00080000 |
| 43 - 50 | X | Half height (not for circle) (value in pixels * 1000), e.g. 40 pixels = 00040000 |
| 51 - 58 | X | Angle (not for circle) (value in ° * 1000), e.g. 180° = 00180000 |

| | | |
|---|---|---|
| Example: | SRP00000055001000200160000001200000<br>0800000004000000180000<br>Length=55, detector=1, yellow search range, rectangle, center X=160, center Y=120, half width= 80, half height=40, orientation=180 | |
| **Set ROI (SRP/SRT) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **S** | Set search range (Set ROI) |
| 2 | **R** | |
| 3 | **P**<br>**T** | Permanent<br>Temporary |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| Example: | SRPP | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |
| Parameter: | | The parameters are given in the coordinate system of the Alignment and not in the coordinate system of the image. |

## Read search range (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Read search range (GRI) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **G** | Read search range (Get ROI) |
| 2 | **R** | |
| 3 | **I** | |
| 4 - 6 | X | Detector no.<br>e.g. 001 |
| 7 - 8 | X | ROI Index<br>00: for yellow search range<br>01: for red teach range<br>02: Position control |
| Example: | GRI00100 | |
| **Read search range (GRI) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **G** | Read search range (Get ROI) |
| 2 | **R** | |
| 3 | **I** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 12 | X | ROI Info Length in bytes, from byte 5 to end<br>39 Byte: circle<br>55 bytes: rectangle, ellipse, free shape |
| 13 - 15 | X | Detector no.<br>e.g. 001 |
| 16 - 17 | X | ROI Index<br>00: for yellow search range<br>01: for red teach range<br>02: Position control |

| | | |
|---|---|---|
| 18 - 19 | X | ROI shape<br>01: Circle<br>02: Rectangle<br>03: Ellipse<br>04: Free shape |
| 20 - 27 | X | Center X (value in pixels * 1000) |
| 28 - 35 | X | Center Y (value in pixels * 1000) |
| 36 - 43 | X | Half width / X-radius (value in pixels * 1000) |
| 44 - 51 | X | Half height (not for circle) (value in pixels * 1000), e.g. 40 pixels = 00040000 |
| 52 - 59 | X | Angle (not for circle) (value in ° * 1000), e.g. 180° = 00180000 |
| Example: | GRIP000000550010002001600000012000000 080000000400000000090000<br>(Length= 55, detector 1, search range, rectangle, center X= 160, center Y= 120, half width= 80, half height= 40, angle= 90) | |

| | |
|---|---|
| Additional information: | |
| Accepted in run mode: | Yes |
| Accepted in configuration mode: | No |
| Accepted when Ready is low: | Yes |
| Status of Ready signal during processing: | Low |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | Max. 4 bytes (optional) |

## Read job list (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Read job list (GJL) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **G** | Read job list (Get Job List) |
| 2 | **J** | |
| 3 | **L** | |
| Example: | GJL | |
| **Read job list (GJL) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **G** | Read job list (Get Job List) |
| 2 | **J** | |
| 3 | **L** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Response version |
| 8 - 10 | X | Number of jobs |
| 11 - 13 | X | Active job number |
| **NOTE:**<br>The following byte sequence is repeated for each job from 1 to "Number of jobs". The byte numbers shift accordingly. | | |
| 14 - 16 | X | Number of characters for the job name. This can be used to specify a unique name for job n. |
| 17 ... n | X | From this position, the name for job n follows in the specified length. |
| n+1 ... n + 3 | X | Number of subsequent bytes. A description for job n can be specified. |
| n + 4 ... m | X | From this position, the description for job n follows in the specified length. |
| m + 1 ... m + 3 | X | Number of subsequent bytes. This can be used to specify a unique name for the author of job n. |
| m + 4 ... k | X | From this position, the name for the author of job n follows in the specified length. |
| k + 1 ... k + 19 | X | Date of creation of Job n (19 bytes) |

| | | |
|---|---|---|
| k + 20 ... k + 39 | X | Date of last modification of job n (19 bytes) |
| Example: | GJLP001001001007testjob010DefaultJob 004Test2014112720141128 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Read detector list (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Read detector list (GDL) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **G** | Get Detector List |
| 2 | **D** | |
| 3 | **L** | |
| Example: | GDL | |
| **Read detector list (GDL) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **G** | Get Detector List |
| 2 | **D** | |
| 3 | **L** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Job number of the current job |
| 8 - 10 | X | Number of detectors in the current job |
| | **NOTE:**<br>The following byte sequence is repeated for each detector in the job. The byte numbers shift accordingly. | |
| 11 - 13 | X | Number of subsequent bytes. This allows a unique name for the detector n to be specified. |
| 14 ... n | X | From this position, the name for detector n follows, in the given length. |

| | | |
|---|---|---|
| n + 1 ... n+ 5 | X | 001 - Pattern matching<br>004 - Contour<br>005 - Gray<br>006 - Contrast<br>007 - Brightness<br>010 - Wafer<br>011 - OCR<br>013 - Datacode<br>014 - Barcode<br>017 - Busbar<br>018 - Color Value<br>019 - Color Area<br>020 - Color List<br>021 - Caliper<br>022 - BLOB |
| Example: | GDLP001001012testdetector00005 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Teach detector (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Teach detector (TED) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **T** | Teach detector |
| 2 | **E** | |
| 3 | **D** | |
| 4 - 6 | X | 0 = Alignment<br>≥ 1 Detectors |
| 7 | X | 0: Temporary<br>1: Permanent |
| 8 | X | 0: No trigger, teach-in with next image acquisition<br>1: Trigger is executed for teach-in |
| Example: | TED00111 | |
| **Teach detector (TED) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **T** | Teach detector |
| 2 | **E** | |
| 3 | **D** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| Example: | TEDP | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Set trigger delay (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Set trigger delay (STD) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **S** | Set Trigger Delay |
| 2 | **T** | |
| 3 | **D** | |
| 4 | 1 | Request version |
| 5 | X | 0: Temporary<br>1: Permanent |
| 6 - 13 | X | Trigger delay<br>in msec (max. 3000 msec)<br>in encoder steps (max. 65535 steps) |
| Example: | STD1100001000 | |
| **Set trigger delay (STD) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **S** | Set Trigger Delay |
| 2 | **T** | |
| 3 | **D** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| Example: | STDP000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Read trigger delay (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Read trigger delay (GTD) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **G** | Read Trigger Delay (Get Trigger Delay) |
| 2 | **T** | |
| 3 | **D** | |
| 4 | 1 | Request version |
| Example: | GTD1 | |
| **Get trigger delay (GTD) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **G** | Read Trigger Delay (Get Trigger Delay) |
| 2 | **T** | |
| 3 | **D** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error code |
| 8 - 15 | X | Trigger delay<br>in msec (max. 3000 msec)<br>in encoder steps (max. 65535 steps) |
| Example: | GTDP00000001000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Save job permanently (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Save Job Permanently (SJP) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **S** | Save Job Permanently (Store Job Permanently) |
| 2 | **J** | |
| 3 | **P** | |
| Example: | SJP | |
| **Save Job Permanently (SJP) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **S** | Save Job Permanently (Store Job Permanently) |
| 2 | **J** | |
| 3 | **P** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| Example: | SJPP | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

### 11.4.4 Calibration

#### Calibration: Initialization (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Initialize (CCD) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **C** | Initialize (Calibration: Clear Data) |
| 2 | **C** | |
| 3 | **D** | |
| Example: | CCD | |
| **Initialize (CCD) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **C** | Initialize (Calibration: Clear Data) |
| 2 | **C** | |
| 3 | **D** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| Example: | CCDP | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Calibration: Add world point (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Add world point (CAW) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Add World Point |
| 2 | **A** | |
| 3 | **W** | |
| 4 | 1 | Request version |
| 5 | X | 1: Calibration plate (Robotics) Fiducials only<br>4: Point pair list (Robotics) World point and image point |
| 6 - 10 | 0 | Constant (5 bytes) |
| 11 - 18 | X | World X (in user unit * 1000) |
| 19 - 26 | X | World Y (in user unit * 1000) |
| 27 - 34 | 0 | Constant (8 bytes) |
| Example: | CAW1000010010000000200000000000000 (World X = 100 mm; World Y = 200mm) | |
| **Calibration: Add world point (CAW) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Add World Point |
| 2 | **A** | |
| 3 | **W** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 12 | X | Current number of points |
| 13 - 20 | X | Image point X |
| 21 - 28 | X | Image point Y |
| Example: | CAWP00000001002880000566000 (Reference point 1; Image X = 288; Image Y = 566) | |
| Additional information: | | |
| Accepted in run mode: | | Yes |

| | |
|---|---|
| Accepted in configuration mode: | No |
| Accepted when Ready is low: | Yes |
| Status of Ready signal during processing: | No change |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | Max. 4 bytes (optional) |

Note: For the CAW request, the overall job result must be positive.

## Calibration: Point pair list (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Calibration by point pair list (CCL) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Point pair list |
| 2 | **C** | |
| 3 | **L** | |
| 4 | X | 0: Temporary<br>1: Permanent |
| Example: | CCL1 | |
| **Calibration: Point pair list (CCL) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Point pair list |
| 2 | **C** | |
| 3 | **L** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 9 | X | Current highest point pair index |
| 10 - 17 | X | Deviation calibration, RMSE |
| 18 - 25 | X | Deviation calibration, mean |
| 26 - 33 | X | Deviation calibration, max. |
| 34 - 41 | X | Deviation calibration, min. |
| Example: | CCLP00010123456781234567812345678123456 78 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Calibration: Validate point pair list (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Validate point pair list (CVL) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Validate Point Pair List |
| 2 | **V** | |
| 3 | **L** | |
| Example: | CVL | |
| **Calibration: Validate point pair list (CVL) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Validate Point Pair List |
| 2 | **V** | |
| 3 | **L** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 9 | X | Current highest point pair index |
| 10 - 17 | X | Deviation calibration, RMSE |
| 18 - 25 | X | Deviation calibration, mean |
| 26 - 33 | X | Deviation calibration, max. |
| 34 - 41 | X | Deviation calibration, min. |
| Example: | CVLP00010123456781234567812345678123456781 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Calibration: Calibration plate (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Calibration: Calibration Plate (CCP) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Calibration plate |
| 2 | **C** | |
| 3 | **P** | |
| 4 | 1 | Request version |
| 5 | X | 0: Temporary<br>1: Permanent |
| 6 | X | 0: No fiducials are used. Origin of Measuring coordinate system identical to origin of Calibration Plate Coordinate System.<br>1: No fiducials are used. Measuring coordinate system identical with Camera coordinate system.<br>2: Uses world system, fiducial job<br>3: Uses world system, fiducial command **CAW** |
| 7 | X | 0: Calibration internal and external sensor parameters<br>1: Validation of calibration<br>2: Calibration internal sensor parameters<br>5: Calibration transformation Measuring coordinate system |
| Example: | CCP1110 | |
| **Calibration: Calibration Plate (CCP) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Calibration plate |
| 2 | **C** | |
| 3 | **P** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 12 | X | Number of currently detected calibration points |

| | | |
|---|---|---|
| 13 - 20 | X | Deviation calibration, RMSE |
| 21 - 28 | X | Deviation calibration, mean |
| 29 - 36 | X | Deviation calibration, max. |
| 37 - 44 | X | Deviation calibration, min. |
| 45 - 52 | X | CPF_MF X (in user unit * 1000) |
| 53 - 60 | X | CPF_MF Y (in user unit * 1000) |
| 61 - 68 | 0 | CPF_MF Z (in user unit * 1000) |
| 69 - 76 | 0 | CPF_MF Angle X (in degrees * 1000) |
| 77 - 84 | 0 | CPF_MF Angle Y (in degrees * 1000) |
| 85 - 92 | X | CPF_MF Angle Z (in degrees * 1000) |
| 93 - 100 | X | Deviation fiducials, mean |
| 101 - 108 | X | Deviation fiducials, max. |
| 109 - 116 | X | Deviation fiducials, min. |
| Example: | CCPP00000012000010010000200200003003000040040<br>000500500006006000070070000800800009009000001001 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Calibration: Set fiducial (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Set fiducial (CSF) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Set fiducial |
| 2 | **S** | |
| 3 | **F** | |
| 4 | 1 | Request version |
| 5 | X | 0: Temporary<br>1: Permanent |
| Example: | CSF11 | |
| **Calibration: Set fiducial (CSF) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Set fiducial |
| 2 | **S** | |
| 3 | **F** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 15 | X | X value (in user unit * 1000) |
| 16 - 23 | X | Y value (in user unit * 1000) |
| 24 - 31 | X | Z value (in user unit * 1000) |
| 32 - 39 | X | Angle X value (in degrees * 1000) |
| 40 - 47 | X | Angle Y value (in degrees * 1000) |
| 48 - 55 | X | Angle Z value (in degrees * 1000) |
| 56 - 63 | X | Deviation fiducials, mean |
| 64 - 71 | X | Deviation fiducials, max. |
| 72 - 79 | X | Deviation fiducials, min. |
| Example: | CSFP0000000100100002002000030030000400400005005000060060000100100002020000 3003 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |

| | |
|---|---|
| Accepted when Ready is low: | Yes |
| Status of Ready signal during processing: | No change |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | Max. 4 bytes (optional) |

## Calibration: Add image (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Add image (CAI) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Add Image |
| 2 | **A** | |
| 3 | **I** | |
| 4 | 1 | Request version |
| 5 | X | Mode<br>1: Multi-image calibration<br>2: Hand-Eye calibration (Robotics)<br>3: Base-Eye calibration (Robotics) |
| 6-8 | 0 | Append at the end of the list (5 bytes) |
| 9 | X | Define Measurement plane<br>0: Do not use image to define Meas-urement plane<br>1: Use image to define Measurement plane |
| 10-11 | X | "Robot: Order of rotation"<br>00: Use rotation order specified in job<br>01: Yaw-Pitch-Roll (e.g. Stäubli)<br>02: Roll-Pitch-Yaw (e.g. Kuka, Fanuc, Hanwha, ABB**, UR**)<br>** when using the corresponding con-version function |
| 12-19 | X | Pose_TCP Pos. X<br>(in user unit * 1000) |
| 20-27 | X | Pose_TCP Pos. Y<br>(in user unit * 1000) |
| 28-35 | X | Pose_TCP Pos. Z<br>(in user unit * 1000) |
| 36-43 | X | Pose_TCP Angle X<br>(in degrees * 1000) |
| 44-51 | X | Pose_TCP Angle Y<br>(in degrees * 1000) |
| 52-59 | X | Pose_TCP Angle Z<br>(in degrees * 1000) |

| | | |
|---|---|---|
| Example: | CAI11 001 1 02<br>00004004000005005000006006000007007000008008000009009 | |
| **Calibration: Add image (CAI) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Add Image |
| 2 | **A** | |
| 3 | **I** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5-7 | X | Error codes (Page 94) |
| 8-10 | X | Current number of images in list |
| 11-15 | X | Total number of detected points |
| Example: | CAIP 000 001 00021 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |

## Calibration: Multi-image (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Multi-image (CMP) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration Calibrate Multi-Image Plate |
| 2 | **M** | |
| 3 | **P** | |
| 4 | 1 | Request version |
| 5 | X | 0: Temporary<br>1: Permanent |
| 6 | X | Origin of the world coordinate system:<br>0: World coordinate system identical with the Calibration Plate Coordinate System (center of the plate).<br>1: Origin of World coordinate system so that it is identical to origin of Image Coordinate System (upper left pixel).<br>2: Use World coordinate system of fiducials, as specified in the job file.<br>3: Use World coordinate system of fiducials as set in request CAW. |
| 7 | X | Mode<br>0: Calibration (internal and external parameters)<br>1: Validieren (vorhandene Kalibrierung verwenden; mindestens ein Kalibrierpunkt wird hinzugefügt. Über Rückprojektion kann zurückgeschlossen werden, ob der Punkt zur aktuellen Kalibrierung passt, oder verschoben ist)<br>2: Calibration (internal parameters only)<br>3: Calibration (external parameters only using new internal parameters)<br>4: Calibration (external parameters only)<br>5: Calibrate Measurement plane only (CPF_MF) |
| Example: | CMP1105 | |

| Calibration: Multi-image (CMP) Response string from sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration Calibrate Multi-image |
| 2 | **M** | |
| 3 | **P** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5-7 | X | Error codes (Page 94) |
| 8-10 | X | Field of view coverage (%) |
| 11-15 | X | Total number of detected points |
| 16-18 | X | Number of images used |
| 19-21 | X | Number of invalid images |
| 22 | X | Sufficient tilt between calibration plate poses<br>0: not sufficient<br>1: sufficient |
| 23-30 | X | Deviation calibration, RMSE [px] |
| 31-38 | X | Deviation calibration, max. [px] |
| 39-46 | X | Deviation fiducials, RMSE (in user unit * 1000) |
| 47-54 | X | Deviation fiducials, max. (in user unit * 1000) |
| Example: | CMPP 000 089 00312 011 002 0 00001001000020020000300300004004 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Calibration: Robotics multi-image (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Robotics multi-image (CRP) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Robotics multi-image (Calibrate Robotics Plate) |
| 2 | **R** | |
| 3 | **P** | |
| 4 | 1 | Request version |
| 5 | x | 0: Temporary<br>1: Permanent |
| 6 | X | Origin of the world coordinate system:<br>4: Set world frame to User Robot Frame |
| 7 | X | Mode<br>0: Calibration (internal and external para-meters)<br>1: Validieren (vorhandene Kalibrierung ver-wenden; mindestens ein Kalibrierpunkt wird hin-zugefügt. Über Rückprojektion kann zurückgeschlossen werden, ob der Punkt zur aktuellen Kalibrierung passt, oder verschoben ist)<br>2: Calibration (internal parameters only)<br>4: Calibration (external parameters only)<br>5: Calibrate Measurement plane only (CPF_MF)<br>6: Calibrate Hand-Eye/Base-Eye |
| Example: | CRP1140 | |
| **Calibration: Robotics multi-image (CRP) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Robotics multi-image (Calibrate Robotics Plate) |
| 2 | **R** | |
| 3 | **P** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5-7 | X | Error codes (Page 94) |
| 8-10 | X | Field of view coverage (%) |

| | | |
|---|---|---|
| 11-15 | X | Total number of detected points |
| 16-18 | X | Number of images used |
| 19-21 | X | Number of invalid images |
| 22-29 | X | Deviation calibration, RMSE [px] |
| 30-37 | X | Deviation calibration, max. [px] |
| 38-45 | X | Deviations calibration plate pose Translation RMSE (in user unit * 1000) |
| 46-53 | X | Deviations calibration plate pose Translation Max. (in user unit * 1000) |
| 54-61 | X | Deviations calibration plate pose Rotation RMSE (in degrees * 1000) |
| 62-69 | X | Deviations calibration plate pose Rotation Max. (in degrees * 1000) |
| Example: | CRPP 000 092 01349 012 004 0000100100002002 00003003000040040000500500006006 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Calibration: Copy Calibration (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Copy calibration (CCC) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Copy calibration |
| 2 | **C** | |
| 3 | **C** | |
| 4 | 1 | Request version |
| 5 | 1 | Constant |
| 6 - 8 | X | Destination<br>0 : Copy to all jobs<br>>0: Copy to specified job |
| 9 | X | 0: Always copy when the calibration is active.<br>1: Only copy if the calibration method is the same. |
| Example: | CCC110021 | |
| **Calibration: Copy calibration (CCC) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Copy calibration |
| 2 | **C** | |
| 3 | **C** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 10 | X | Job number of the job where the error occurred<br>00: Successful<br>>0 - Job number of the job where the error first occurred |
| Example: | CCCP000000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |

| | | |
|---|---|---|
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Calibration: Set parameter (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Set parameter (CSP) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Set Parameter |
| 2 | **S** | |
| 3 | **P** | |
| 4 | 1 | Request version |
| 5 | X | 0: Temporary<br>1: Permanent |
| 6 - 8 | X | Parameter number, see table Calibration parameters CSP and CGP |
| 9 - 16 | X | Length of value |
| 17 ... n | X | Value for selected parameter, see table Calibration parameters CSP and CGP |
| Example: | CSP11002000000019 | |
| **Calibration: Set parameter (CSP) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Set Parameter |
| 2 | **S** | |
| 3 | **P** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| Example: | CSPP000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Calibration parameters: see table Calibration parameters for telegrams CSP and CGP

## Calibration: Read parameter (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Read parameter (CGP) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Read Parameter |
| 2 | **G** | |
| 3 | **P** | |
| 4 | 1 | Request version |
| 5 - 7 | X | Parameter number, see calibration parameters CSP and CGP |
| Example: | CGP1001 | |
| **Calibration: Read parameter (CGP) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **C** | Calibration: Read Parameter |
| 2 | **G** | |
| 3 | **P** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 10 | X | Parameter number, see calibration parameters CSP and CGP |
| 11 - 18 | X | Length of the following data |
| 19 ... n | X | Parameter values, depending on the selected parameter |
| Example: | CGPP000001000000011 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

**Calibration parameters for telegrams CSP and CGP**

| Parameter description | Parameter number | Value | length | Calibration status after CSP |
|---|---|---|---|---|
| Status calibration | 001 | 0: Invalid<br>1: Valid | 1 byte | –* |
| Calibration method | 002 | 0: None<br>2: Point pair list (Robotics)<br>3: Calibration plate (Measurement)<br>4: Calibration plate (Robotics)<br>5: Hand-Eye calibration (Robotics)<br>6: Base-Eye calibration (Robotics) | 1 byte | invalid |
| Unit (user unit) | 004 | 0: Millimeter [mm]<br>1: Centimeter [cm]<br>2: Meter [m]<br>3: Inch ["]<br>4: Arbitrary unit [au] | 1 byte | no change |
| Internal parameters | 010 | Focal length (in mm *1000)<br>Kappa (*1000)<br>Pixel pitch X (in µm * 1000)<br>Pixel pitch Y (in µm * 1000)<br>Coordinate origin X (in pixels * 1000)<br>Coordinate origin Y (in pixels * 1000)<br>Image size X (number of pixels)<br>Image size Y (number of pixels) | 64<br>(8 * 8 bytes per value) | –* |
| Reference Camera- to Measuring coordinate system (CF_MF) | 011 | Translation X, Y, Z (in user unit * 1000)<br>Angle X, Y, Z (in degrees * 1000) | 48<br>(6 * 8 bytes per value) | –* |
| Reference Camera- to Calibration Plate Coordinate System (CF_CPF) | 012 | Translation X, Y, Z (in user unit * 1000)<br>Angle X, Y, Z (in degrees * 1000) | 48<br>(6 * 8 bytes per value) | –* |

| Parameter description | Parameter number | Value | length | Calibration status after CSP |
|---|---|---|---|---|
| Reference Robot- to Camera coordinate system (RF_CF) | 013 | Translation X, Y, Z (in user unit * 1000)<br>Angle X, Y, Z (in degrees * 1000) | 48 (6 * 8 bytes per value) | –* |
| Reference Calibration plate- to Measuring coordinate system (CPF_MF) | 014 | Translation X, Y, Z (in user unit * 1000)<br>Angle X, Y, Z (in degrees * 1000) | 48 (6 * 8 bytes per value) | –* |
| Reference Robot- to Measuring coordinate system (RF_MF) | 015 | Translation X, Y, Z (in user unit * 1000)<br>Angle X, Y, Z (in degrees * 1000) | 48 (6 * 8 bytes per value) | –* |
| Reference TCP- to Camera coordinate system (TCP_CF) | 016 | Translation X, Y, Z (in user unit * 1000)<br>Angle X, Y, Z (in degrees * 1000) | 48 (6 * 8 bytes per value) | –* |
| Reference robot- to TCP coordinate system (RF_TCP) | 017 | Translation X, Y, Z (in user unit * 1000)<br>Angle X, Y, Z (in degrees * 1000) | 48 (6 * 8 bytes per value) | no change |
| Z-shift Measurement plane | 021 | Value (in user unit * 1000) | 8 Byte | no change |
| Focal length in [mm] | 022 | [mm * 1000] | 8 Byte | invalid (CSP for C-Mount only) |
| Calibration plate type | 023 | Character string with name of the description file | n | invalid |
| Fiducial 1 | 024 | Translation X, Y, Z (in user unit * 1000) | 24 (3 * 8 bytes per value) | invalid |
| Fiducial 2 | 025 | | | |
| Fiducial 3 | 026 | | | |
| Fiducial 4 | 027 | | | |

| Parameter description | Parameter number | Value | length | Calibration status after CSP |
|---|---|---|---|---|
| Number of existing calibration plate types | 037 | Request - Selection of type:<br>0: All<br>1: Measurement<br>2: Robotics<br>Response:<br>Number of plates | Request: 1<br>Response: 5 | –* |
| Available calibration plate types (file names) | 038 | Request - Selection of type:<br>0: All<br>1: Measurement<br>2: Robotics<br>Request - Index:<br>0: All file names<br>>0: Index selection<br>Response:<br>File names of Calibration plates | Request: 1 / 5<br>Response: n (String) | –* |
| Robot: Order of rotation | 039 | "Robot: Order of rotation"<br>00: Use rotation order specified in job<br>01: Yaw-Pitch-Roll (e.g. Stäubli)<br>02: Roll-Pitch-Yaw (e.g. Kuka, Fanuc, Hanwha, ABB**, UR**)<br>** when using the corresponding conversion function | 1 | invalid |
| Average sensor resolution | 041 | Value (in user unit/pixel * 1000) | 8 bytes | –* |

* CSP not possible (parameter is read-only and cannot be set).

## 11.4.5 Visualization

### Get image (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Get image (GIM) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **G** | Get Image |
| 2 | **I** | |
| 3 | **M** | |
| 4 | X | 0: Last image<br>1: Last bad image<br>2: Last good image |
| Example: | GIM1 | |
| **Get image (GIM) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **G** | Get Image |
| 2 | **I** | |
| 3 | **M** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 | X | Error codes (Page 94) |
| 6 | X | Image type<br>0: Grayscale<br>3: Bayer-Pattern_BG<br>When converting the color image from Bayer into RGB, the appropriate image type must be considered.<br>Pre-processing filters of the category "Arrange-ment" have an influence on the Bayer type.<br>Bayer Pattern begins with blue - green. |
| 7 | X | Image result<br>1: Good image<br>0: Failed image |

| | | |
|---|---|---|
| 8 - 11 | X | Number of rows<br>e.g. 0480 / 0200 |
| 12 - 15 | X | Number of columns<br>e.g. 0640 / 0320 |
| 16 - 19 | X | End of the message string if specified. Otherwise start image data from Byte no. 16. |
| 20 ... n | X | Binary image data (rows * columns) |
| Example: | GIMP0004800640... | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

### 11.4.6 Service (available only on port 1998 and in ASCII format)

## Update visualization data (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Update visualization data (UVR) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **U** | Update visualization data (Update Visualization Results) |
| 2 | **V** | |
| 3 | **R** | |
| 4 | 1 | Request version |
| 5 | X | Image:<br>0: No image is created<br>1: Grayscale / RGB image without filter, BMP format<br>2: Grayscale image / Bayer pattern without filter, BMP format<br>3: Grayscale / RGB image with filter, BMP format<br>4: Grayscale image / Bayer pattern with filter, BMP format<br>5: Grayscale / RGB image without filter, JPEG format (low compression)<br>6: Grayscale / RGB image with filter, JPEG format (low compression)<br>7: Grayscale / RGB image without filter, JPEG format (compression high)<br>8: Grayscale / RGB image with filter, JPEG format (compression high) |
| 6 | X | Result XML:<br>0: Result file is not created<br>1: Result file is created |
| 7 | X | Statistic XML:<br>0: Statistics file is not created<br>1: Statistics file is created |

| | | |
|---|---|---|
| 8 | X | Image type:<br>0: Last image (Any)<br>1: Last fail image (Fail)<br>2: Last pass image (Pass)<br>3: Next image (Any)<br>4: Next fail image (Fail)<br>5: Next pass image (Pass) |
| 9 - 11 | X | Directory number (constant)<br>001: visu001 |
| Example: | UVR11110001 | |
| **Update visualization data (UVR) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **U** | Update visualization data (Update Visualization Results) |
| 2 | **V** | |
| 3 | **R** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 | X | Data available:<br>0: New data available when ready.txt is written<br>1: No new data available. |
| 9 - 11 | X | Directory number (constant)<br>001: visu001 |
| Example: | UVRP0000001 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

The created files are available for download in the directory /tmp/[Directory number]:

- image.bmp
- overlay.xml

With the file "overlay.xml", all relevant information for creating the overlay can be obtained. The file is created in XML format. The most important elements are described in the table below

| Name | | Value | Description |
|---|---|---|---|
| detector | type | pattern_match-ing<br>contour<br>contrast<br>brightness<br>gray<br>caliper<br>blob<br>ocr<br>datacode<br>barcode | Detector Type |
| | number | Integer | Position in detector list |
| | name | String | Name of the detector defined in the con-figuration |
| roi | purpose | Search<br>teach<br>position_con-trol<br>result | Type of overlay element. The different types have different colors. |
| | shape | rectangle<br>rectangle_mask<br>ellipse | Shape of the overlay element |
| center | x | Float | Center position in X (pixels) |
| | y | Float | Center position in Y (pixels) |
| size | half_width | Float | Half width of overlay element |
| | half_height | Float | Half height of overlay element |
| angle | angle | Float | Angle of overlay element (degrees) |
| number | value | Float | Number of element types in this detector |
| line | x1 | Float | Start point X line 1 (pixels) |
| | y1 | Float | Start point Y line 1 (pixels) |
| | x2 | Float | Start point X line 2 (pixels) |
| | y2 | Float | Start point Y line 2 (pixels) |

Depending on the detector type (detector → type), there are different elements that can be displayed. The following table indicates which element can be displayed on which detector.

| Detector | Search | teach | position_con-trol | result |
|---|---|---|---|---|
| Pattern matching | Yes | Yes | Yes | 1 |
| Contour | Yes | Yes | Yes | 200 |
| Contrast | Yes | No | No | 0 |
| Brightness | Yes | No | No | 0 |
| Gray | Yes | No | No | 0 |
| Caliper | Yes | No | No | 0 |
| BLOB | Yes | No | No | 1000 |
| OCR | Yes | No | No | 1 |
| Datacode | Yes | No | No | 5 |
| Barcode | Yes | No | No | 5 |

## Read sensor identity (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Read sensor identity (GSI) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **G** | Read sensor identity (Get Sensor Identity) |
| 2 | **S** | |
| 3 | **I** | |
| 4 | 1 | Request version |
| Example: | GSI1 | |
| **Read sensor identity (GSI) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **G** | Read sensor identity (Get Sensor Identity) |
| 2 | **S** | |
| 3 | **I** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 10 | X | Length of the following data |
| 11 ... n | X | Version of the firmware as well as information about the hardware. Areas are clearly separated by a semicolon. |
| Example: | GSIP0000262.0.0.3; V20-RO-P3-R-M-M2-L | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

## Update firmware (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Update firmware (UFW) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **U** | Update firmware |
| 2 | **F** | |
| 3 | **W** | |
| 4 | 1 | Request version |
| Example: | UFW1 | |
| **Update firmware (UFW) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **U** | Update firmware |
| 2 | **F** | |
| 3 | **W** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| Example: | UFWP000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

After the command is sent, the /tmp/ on the VISOR® vision sensor will be checked for a valid firmware file. The name must correspond to the typical name allocation (e.g. as after the download from the SensoPart homepage). The end is reached as soon as the camera signals ready (pin 4 GN) again. Alternatively, the telegram "GSI1" can be used to check whether a valid response is being sent.

**NOTE:**
The voltage supply must be ensured during the firmware update. An update may take up to 10 minutes.

## Read job set (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Read job set (SJS) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **S** | Read job set (Set Jobset) |
| 2 | **J** | |
| 3 | **S** | |
| 4 | 1 | Request version |
| 5 - 7 | X | Length of subsequent file name. Maximum length 250 characters. |
| 8 ... n | X | Optional file name. If no file name is specified, the default name "Jobset.job" is used. |
| Example: | SJS1010jobset.job | |
| **Read job set (SJS) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **S** | Read job set (Set Jobset) |
| 2 | **J** | |
| 3 | **S** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 10 | X | Active job number in the loaded job set |
| Example: | SJSP000001 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | No |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

The job set with the specified name will be searched for in the /tmp/ directory on the VISOR® vision sensor. If the file exists, this job set is activated. The file is then removed.

## Save job set (ASCII)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Save job set (GJS) Request string to sensor (ASCII)** | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | **G** | Save job set from VISOR® |
| 2 | **J** | |
| 3 | **S** | |
| 4 | **1** | Request version |
| 5 - 7 | X | Length of subsequent file name. Maximum length 250 characters. |
| 8 ... n | X | Optional file name. If no file name is specified, the default name "Jobset.job" is used. |
| Example: | GJS1010jobset.job | |
| **Save job set (GJS) Response string from sensor (ASCII)** | | |
| Byte no. | Content | Meaning |
| 1 | **G** | Save job set from VISOR® |
| 2 | **J** | |
| 3 | **S** | |
| 4 | P<br>F | P: (Pass) Success<br>F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| Example: | GJSP000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

The read job set file is now available for download in the /tmp/ directory under the specified name.

## 11.4.7 Data output ASCII

Output data (ASCII), dynamically composed according to user settings in the software under: SensoConfig / Output / Data output.

Basic string structure:

| |
|---|
| <START> (((<OPTIONAL FIELDS> <SEPARATOR> <PAYLOAD>))) <CHKSUM> <TRAILER> |

**Output data (ASCII):**

| **<OPTIONAL FIELDS>** | | | | |
|---|---|---|---|---|
| **Parameter** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Selected fields | With this checkbox all selected fields are displayed. The checkbox "Selected fields" itself is not displayed. | 16 | The output sequence is from left to right and from top to bottom, i.e. one byte is set per active checkbox, starting with the LSB. | All types |
| Telegram length | Number of characters including the characters for the telegram length itself. | 1 ... 10 | E.g. output string with 10 characters; telegram length 10 + 2 characters (one byte per decimal place) = 12 | All types |
| Status byte | Returns the Trigger mode. | 3 | PPF = Trigger<br>PFP = Free run | All types |

| <OPTIONAL FIELDS> | | | | |
|---|---|---|---|---|
| **Parameter** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Detector results | Output of overall result for each detector. | 4 ... 261 | Byte 1 = AND conjunction of all detectors<br>Byte 2 = Overall Alignment result<br>Byte 3 = Overall result of current job<br>Followed by the number of detectors; one byte per decimal place<br>Followed by one byte for each detector; P = Detector pass<br>F = Detector fail | All types |
| Digital outputs | Returns the logic gate result for each digital output. | 2 ... 7 | Byte 1 Number of active outputs (logic gate result assigned)<br>Followed by bytes 2 – 7; one byte per output<br>P = Detector pass<br>F = Detector fail 0 = Inactive output (gap between two active outputs) | All types |

| <OPTIONAL FIELDS> | | | | |
|---|---|---|---|---|
| **Parameter** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| log. Outputs | Returns the logic gate result for each logic output. | 1 ... 259 | Starting from byte 1 Number of active outputs (logic gate result assigned); 1 byte per decimal place<br>Following bytes:<br>One byte per logic output<br>P = Detector pass<br>F = Detector fail<br>0 = Inactive output (gap between two active outputs) | All types |
| Execution time | Returns the execution time for the last evaluation. | 1 ... 3 | Signed integer | All types |
| Active job | Returns the job for the last evaluation. | 1 ... 3 | Unsigned int U8 | All types |

**<PAYLOAD>**

**Overview of detector-specific payload - Values**

**GENERAL**

| <PAYLOAD> General | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| "All evaluations" counter | Total number of checks | 1 ... 11 | Signed integer | GENERAL |
| Pass parts counter | Number of inspections with result "OK" | 1 ... 11 | Signed integer | GENERAL |

| <PAYLOAD> General | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Fail parts counter | Number of inspections with result "Error" | 1 ... 11 | Signed integer | GENERAL |
| Timeout | Indicates that the maximum cycle time has been exceeded. | 1 | BOOL | GENERAL |
| Recording | Indicates the number of image acquisition repetitions for the last evaluation Only in combination with repeat mode. | 1 ... 3 | INT | GENERAL |
| String | This field can be used to enter a constant string into the data output. | 1 ... 50 | STRING | GENERAL |

**Base values**

| <PAYLOAD> Base values | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Score | [%] | 1 ... 6 | Signed integer | All detectors |
| Overall result | Boolean detector result | 1 | BOOL | All detectors |
| Execution time | Execution time of individual detector in [msec]. | 1 ... 11 | Signed integer | All detectors |

**Position**

| <PAYLOAD> Position / location | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Pos. X | X coordinate for the found position, 1/1000 [user unit] | 1 ... 11 | Signed integer | |
| Pos. Y | Y coordinate for the found position, 1/1000 [user unit] | 1 ... 11 | Signed integer | |
| Pos. Z | Z coordinate of the found position, 1/1000 [user unit] | | Signed integer | With Result off-set: |
| Delta Pos. X | X position delta between the taught object and the found object, 1/1000 [user unit] | 1 ... 11 | Signed integer | |
| Delta Pos. Y | Y position delta between the taught object and the found object, 1/1000 [user unit] | 1 ... 11 | Signed integer | |
| Delta Pos. Z | Z position delta between the taught object and the found object, 1/1000 [user unit] | 1 ... 11 | Signed integer | With Result off-set: |
| Angle X | Orientation of the found object, relative to the X-axis, 1/1000 [°] | 1 ... 11 | Signed integer | With Result off-set: |

| <PAYLOAD> Position / location | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Angle Y | Orientation of the found object, relative to the Y-axis, 1/1000 [°] | 1 ... 11 | Signed integer | With Result off-set: |
| Angle Z | Orientation of the found object, relative to the Z-axis, 1/1000 [°] | 1 ... 11 | Signed integer | |
| Angle (45) | Orientation of bounding box for found code [°], Value range: -45° to 45° | 1 ... 6 | Signed integer | |
| Angle (180) | Orientation of object width (long axis) [°], Value range: -90° to 90° 0° = East, counterclockwise | 1 ... 7 | Signed integer | |
| Angle (360) | Orientation of object width (long axis) [°], Value range: -180° to 180° 0° = East, counterclockwise | 1 ... 7 | Signed integer | |
| Delta Angle X | Angle between taught-in and found object, referred to the X-axis, 1/1000 [°] | 1 ... 7 | Signed integer | With Result off-set: |

| <PAYLOAD> Position / location | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Delta Angle Y | Angle between taught-in and found object, referred to the Y-axis, 1/1000 [°] | 1 ... 7 | Signed integer | With Result off-set: |
| Delta Angle Z | Angle between taught-in and found object, referred to the Z-axis, 1/1000 [°] | 1 ... 7 | Signed integer | |
| Pose 3D (X, Y, Z, Angle X, Angle Y, Angle Z) | Coordinates of the found object, 1/1000 [user unit] Angle: 1/1000 degrees | 1...7 bytes per value; separated by specified separator | Signed integer | With Result off-set: |
| Delta Pose 3D (X, Y, Z, Angle X, Angle Y, Angle Z) | Delta coordinates of the found object, 1/1000 [user unit] Angle: 1/1000 degrees | 1...7 bytes per value; separated by specified separator | Signed integer | With Result off-set: |
| Position control | | 1 | BOOL | |

**Measurement**

| <PAYLOAD> Measurement | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| height | Height of geometric element [user unit]*, Height $\geq 0$, height $\leq$ width | 1 ... 11 | Signed integer | |

| <PAYLOAD> Measurement | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Width | Width of geometric element [user unit]*, Width ≥ 0, width ≥ height | 1 ... 11 | Signed integer | |
| Radius | Radius of fitted circle [user unit] | 1 ... 11 | Signed integer | |
| Area | Area of BLOB without holes, 1/1000 [pixels] | 1 ... 11 | Signed integer | |
| Area (incl. holes) | Area of BLOB including holes, 1/1000 [pixels] | 1 ... 11 | Signed integer | |
| Distance | Calculated distance [user unit] | 1 ... 11 | Signed integer | |

### Identification

| <PAYLOAD> Identification | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| String length | Length of read code [bytes] | 1 ... 3 | Signed integer | |

| <PAYLOAD> Identification | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| String | Content of the read code.<br>Depending on the code, the string length may vary. If a fixed string length is desired, the minimum string length (detector-specific payload) and the maximum string length (detector settings) must be set to the same value (e.g. 127). | 0 ... 255 | STRING | |
| String comparison | Content check for the read information. The content of the read information is checked on the basis of regular expressions<br>(see detector Datacode, Reference string tab) | 1 | BOOL | |
| Truncated | Code complete or truncated<br>F: Code complete<br>P: Code truncated | 1 | BOOL | |

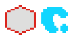
### Identification - quality

| <PAYLOAD> Identification - Quality | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Quality - overall | Output of all Q parameters. Depending on the selected code type and standard. | 1 byte per value; separated by specified separator<br>For 2D code parameter Q9 (mean light): 1...3 | Unsigned Char;<br>for 2D Code Q9 (Meanlight) Unsigned Short | |
| Quality - individual | Output of individual quality values: Selection Q1-Q24 depending on the selected code type and standard.<br>Numbers: 1-4<br>Letters: A-F | 1<br>For 2D code parameter Q9 (mean light): 1...3 | Unsigned Char;<br>for 2D Code Q9 (Meanlight) Unsigned Short | |
| Min. Quality | Used to check whether the minimum required quality is being met | 1 ... 7 | Unsigned int | |

### Color

| <PAYLOAD> Color | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Color value:<br>• Red, green, blue<br>• Hue, saturation, lightness<br>• Luminance, a, b | Value for color parameter | 0 ... 7 | Signed integer |  |
| Color distance | Distance of the current color versus the taught-in color | 0 – 7 | Signed integer |  |

### Counting / number

| <PAYLOAD> Counting / number | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Number of objects | Number of objects found [units] | 1 ... 5 | Signed integer |  |
| Number of valid objects | Number of valid objects found [units] | 1 ... 5 | Signed integer |  |
| Number of search stripes | Number of parallel search stripes into which the width of the search range is divided. [units] | 1 ... 5 | Signed integer |  (Edge detector only)  |

| <PAYLOAD> Counting / number | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Number of valid search stripes | Number of search stripes used to generate results [units] | 1 ... 3 | Signed integer | (Edge detector only) |
| Result vector | Vector containing the result (1/0) of the instances found | | | |
| Too many BLOBs | | 1 | BOOL | |

**Extended**

| <PAYLOAD> Extended | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Scaling | Current scaling factor to the taught-in reference. 1/1000 (factor). Value range of 0.5 to 2 | 3 ... 4 | Unsigned int | (Contour matching only) |
| Eccentricity | Numerical eccentricity Value range of 0.0 to 1.0 | N | Signed integer | |

| **<PAYLOAD> Extended** | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Security | Output of the security values of the individual characters. The reliability value specifies how reliably the reader was able to interpret a character.<br>Value range of 0 to 100 [%] | N | Unsigned int |  |
| Reference string met | The output string matches the reference string. | 1 | BOOL |  |
| contrast | Code contrast<br>Value range of 0 to 100 [%] | N | Unsigned int |  |
| Correction | Number of modules corrected by error corrections<br>[units] | N | Unsigned int |  |
| Contour length | Number of pixels of outer contour,<br>1/1000 [pixels] | N | Signed integer |  |
| Compactness | BLOB compactness (circle =1; other > 1). The more the shape of the BLOB deviates from a circle, the greater the compactness value will be. | N | Signed integer |  |
| Center of gravity X | X coordinate of centroid,<br>1/1000 | N | Signed integer |  |

| <PAYLOAD> Extended | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Center of gravity Y | Y coordinate of centroid, 1/1000 | N | Signed integer | |
| Gray scale value, average | Average gray scale value of all the pixels that belong to the BLOB. | N | Signed integer | |
| Min. signal threshold | Lower threshold for the binarization of the objects. 0...255 | 1 ... 3 | Unsigned int | |
| Max. signal threshold | Upper threshold for the binarization of the objects. 0...255 | 1 ... 3 | Unsigned int | |
| Inverted signal threshold | Specifies whether the range Min <-> Max is inverted.<br>P: inverted<br>F: not inverted | 1 | Unsigned Char | |
| Deviation, inside | Returns the largest deviation between the BLOB contour and the contour of the geometric element (deviation inside the fitted circle).<br>[User unit * 1000] | 1 ... 7 | Signed integer | |
| Deviation, outside | Returns the largest deviation between the BLOB contour and the contour of the geometric element (deviation outside the fitted circle).<br>[user unit] | 1 ... 7 | Signed integer | |

| <PAYLOAD> Extended | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Deviation, mean | Returns the mean of the absolute "inside" and "outside" deviation values between the BLOB contour and the contour of the geometric element. | 1 ... 7 | Signed integer | |
| Axial ratio | Ratio of the long to the short axis (a / b) | 1 ... 7 | Signed integer | |
| Face up / down, area | Face up / down position, based on: area, position indicated by sign, 1/1000 | N | Signed integer | |
| Result index | List index | N | Signed integer | |
| Search stripe distance | Calculated distance [user unit] / 1000 per pair of search stripes | 1 ... 11 | Signed integer | |

| <CHKSUM> | | | | |
|---|---|---|---|---|
| **Parameter** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Check sum | XOR check sum of all bytes in the telegram. Is transmitted as the last byte. | 1 | Unsigned int | All types |

| <TRAILER> | | | | |
|---|---|---|---|---|
| **Parameter** | **Description** | **Length ASCII [Byte]** | **Data type** | **Available for** |
| Start | User-defined, up to a max. of 8 char- acters | 0 ... 8 | Unsigned int | All types |

***NOTE:**
If no calibration has been performed, all values refer to pixels.

## 11.5 Description Telegrams BINARY

### 11.5.1 General

### Reset statistics (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Reset Statistics (RST) Request string to sensor (BINARY)** | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x04 | Reset statistics |
| **Reset Statistics (RST) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x04 | Reset statistics |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

## 11.5.2 Control

### Trigger (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Trigger (TRG) Request string to sensor (BINARY)** | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x01 | Trigger, (simple trigger without index, via port 2006) |
| **Trigger (TRG) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x01 | Trigger, (response to trigger command without index, via port 2006. If defined: Result data without index via port 2005) |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | Yes |
| Accepted when Ready is low: | | | No |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

## Extended trigger (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Extended Trigger (TRX) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length 6 bytes + length of subsequent data (n) |
| 5 | Unsigned Char | 0x13 | Extended trigger (trigger with index, for correlation of trigger to corresponding result data, via port 2006) |
| 6 | Unsigned Char | 0xXX | Length of following data (n) |
| 7 ... n | Unsigned Char | 0xXX | Data |
| **Extended Trigger (TRX) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length |
| 5 | Unsigned Char | 0x13 | Extended Trigger command, (response to trigger with index and result data, via port 2006, for correlation of trigger to corresponding result, Result data without index, via port 2005 also) |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Length of following data (n) |
| 9 ... n | Unsigned Char | 0xXX | Data of sending command |
| n+1 | Unsigned Char | 0xXX | Operating mode<br>0 = Config mode<br>1 = Run mode |
| n + 2 ... n + 5 | Unsigned int | 0xXX | Length of result data |
| n + 6 ... m | Unsigned Char | 0xXX | Result data |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |

| | |
|---|---|
| Accepted in configuration mode: | Yes |
| Accepted when Ready is low: | No |
| Status of Ready signal during processing: | Low |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) |

## Trigger Robotics (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Trigger Robotics (TRR) Request string to sensor (BINARY)** | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length 31 (0x1F) + Length of trigger identifier in Bytes |
| 5 | Unsigned Char | 0x37 | Trigger Robotics |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | Length of trigger identifier in bytes |
| 8-n | Unsigned Char | 0xXX | Trigger Identifier |
| n+1…n+4 | Unsigned int | 0xXX | Pose_TCP Pos. X (in user unit * 1000) |
| n+5…n+8 | Unsigned int | 0xXX | Pose_TCP Pos. Y (in user unit * 1000) |
| n+9…n+12 | Unsigned int | 0xXX | Pose_TCP Pos. Z (in user unit * 1000) |
| n+13…n+16 | Unsigned int | 0xXX | Pose_TCP Angle X (in degrees * 1000) |
| n+17…n+20 | Unsigned int | 0xXX | Pose_TCP Angle Y (in degrees * 1000) |
| n+20…n+24 | Unsigned int | 0xXX | Pose_TCP Angle Z (in degrees * 1000) |
| **Trigger Robotics (TRR) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length 8 (0x08) + Length of trigger identifier in Bytes |
| 5 | Unsigned Char | 0x37 | Trigger Robotics, (Response to command Trigger without index, via port 2006. If defined: Result data without index via port 2005) |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |

| | | | |
|---|---|---|---|
| 8 | Unsigned Char | 0xXX | Length of trigger identifier |
| 9-n | Unsigned Char | 0xXX | Trigger Identifier |
| n+1 | Unsigned Char | 0xXX | Operation Mode<br>0x00 = Config<br>0x01 = Run |
| n+2…n+5 | Unsigned int | 0xXX | Length of the result data in bytes |
| n+6…m | Unsigned int | 0xXX | Result data |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | Yes |
| Accepted when Ready is low: | | | No |

Note: For "Calibration plate (Robotics)" and "Point pair list (Robotics)" only the X and Y position are taken into account. The other values (position Z and rotations) must be 0.

## Set Trigger ID (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Set Trigger ID (STI) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length<br>7 Bytes + length of Trigger ID |
| 5 | Unsigned Char | 0x2E | Set trigger ID |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | Length of the following data (max 99) |
| 8-n | Unsigned Char | 0xXX | Trigger ID |
| Example: | 0x00 0x00 0x00 0x0D 0x2E 0x01 0x06 0x30 0x31 0x32 0x33 0x34 0x35 | | |
| **Set Trigger ID (STI) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x2E | Set trigger ID |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Example: | 0x00 0x00 0x00 0x07 0x2E 0x00 0x00 | | |
| Additional information: | | | |
| Accepted in run mode: | | Yes | |
| Accepted in configuration mode: | | Yes | |
| Accepted when Ready is low: | | Yes | |

## Job change (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Job change (CJB) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x06 | Telegram length |
| 5 | Unsigned Char | 0x02 | Change job |
| 6 | Unsigned Char | 0xXX | Job no. XX = 1 - n |
| **Job change (CJB) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x09 | Telegram length |
| 5 | Unsigned Char | 0x02 | Change job |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Trigger mode<br>0x00: Trigger<br>0x01: Free run |
| 9 | Unsigned Char | 0xXX | Job no. XX = 1 - n |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

**NOTE:**
If an error occurs during the job change, it is possible to change to Job 1.

## Job Change Permanent (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Job Change Permanent (CJP) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x06 | Telegram length |
| 5 | Unsigned Char | 0x22 | Job change permanent |
| 6 | Unsigned Char | 0xXX | Job no. XX = 1 - n |
| **Job Change Permanent (CJP) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x09 | Telegram length |
| 5 | Unsigned Char | 0x22 | Job change permanent |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Trigger Mode<br>0x00: Trigger<br>0x01: Free run |
| 9 | Unsigned Char | 0xXX | Job no. XX = 1 - n |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

**NOTE:**
If an error occurs during the job change, it is possible to change to Job 1.

## Job change by job name (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Job change by job name (CJN) Request string to sensor (BINARY)** | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length 7 bytes + length job name (n) |
| 5 | Unsigned Char | 0x2C | Job change by job name |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | Job name length (n) |
| 8 - n | Unsigned Char | 0xXX | Job name |
| **Job change by job name (CJN) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x08 | Telegram length |
| 5 | Unsigned Char | 0x2C | Job change by job name |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Trigger mode<br>0x00: Trigger<br>0x01: Free run |
| Additional information: | | | |
| Accepted in run mode: | | Yes | |
| Accepted in configuration mode: | | No | |
| Accepted when Ready is low: | | Yes | |
| Status of Ready signal during processing: | | Low | |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) | |

### 11.5.3 Job settings

## Auto working distance (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Auto working distance (AFC) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length 11 Bytes (0x0B) + selected options 8 Bytes (0x08) |
| 5 | Unsigned Char | 0x32 | Auto working distance |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary<br>0x01: Permanent |
| 8 | Unsigned Char | 0xXX | Step size of search (0x01 - 0x05) |
| 9 | Unsigned Char | 0xXX | Selection of distance value<br>0x00: Highest score<br>0x01: Min. Working distance<br>0x02: Max. working distance<br>0x03: Average working distance<br>0x04: Median working distance<br>0x05: Highest score - output of all working distances found |
| 10 | Unsigned Char | 0xXX | Unit<br>0x00: 1/1000 millimeters (µm)<br>0x01: Motor steps |
| 11 | Unsigned Char | 0xXX | Selection of search range<br>0x00: Entire range<br>0x01: Selected range |
| 12...15 | Unsigned int | X | Start of search range (only if search range selection == 0x01) |
| 16...19 | Unsigned int | X | End of search range (only if selection Search range == 0x01) |
| **Auto working distance (AFC) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |

| | | | |
|---|---|---|---|
| 1 - 4 | Unsigned int | 0xXX | Telegram length 11 Bytes (0x0B) + working distances + score values |
| 5 | Unsigned Char | 0x32 | Auto working distance |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | Unsigned int | X | Number of output working distances |
| 12 - n | Unsigned int | X | Distance value in 1/1000 mm or motor steps (4 bytes per output working distance) |
| n-m | Unsigned int | X | Score value to distance value multiplied by 1000 (4 bytes per output working distance) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

## Set working distance (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Set working distance (SFC) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0D | Telegram length |
| 5 | Unsigned Char | 0x31 | Set working distance |
| 6 | Unsigned Char | 0xX1 | Request version |
| 7 | Unsigned Char | 0xXX | 0: Temporary<br>1: Permanent |
| 8 | Unsigned Char | 0xXX | Movement<br>0: Absolute<br>1: Relative<br>2: Absolute with reinitialization |
| 9 | Unsigned Char | 0xXX | Unit<br>0: 1/1000 millimeters<br>4: Steps |
| 10 - 13 | Signed integer | 0xXX | Working distance |
| **Set working distance (SFC) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0B | Telegram length |
| 5 | Unsigned Char | 0x31 | Set working distance |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | INT | 0xXX | Current working distance |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

## Read working distance (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Read working distance (GFC) Request string to sensor (BINARY)** | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x30 | Read working distance |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | Unit<br>0x00: 1/1000 millimeter<br>0x04: Steps |
| **Read working distance (GFC) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0B | Telegram length |
| 5 | Unsigned Char | 0x30 | Read working distance |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | INT | 0xXX | Current working distance |
| Additional information: | | | |
| Accepted in run mode: | | Yes | |
| Accepted in configuration mode: | | No | |
| Accepted when Ready is low: | | Yes | |
| Status of Ready signal during processing: | | No change | |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) | |

## Auto shutter speed (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Auto shutter speed (ASH) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x07 | Auto shutter speed |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary<br>0x01: Permanent |
| **Auto shutter speed (ASH) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0F | Telegram length |
| 5 | Unsigned Char | 0x07 | Auto shutter speed |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | INT | 0xXX | Auto shutter speed value |
| 12 - 15 | INT | 0xXX | Score |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

## Set shutter speed value (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Set shutter speed (SSP/SST) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x09 | Telegram length |
| 5 | Unsigned Char | 0xXX | 0x0E Set shutter speed temporarily<br>0x0F Set shutter speed permanently |
| 6 - 9 | Unsigned int | 0xXX | Shutter speed value in 1/1000 ms |
| **Set shutter speed (SSP/SST) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0xXX | 0x0E Set shutter speed temporarily<br>0x0F Set shutter speed permanently |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

## Read shutter speed value (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Read Shutter Speed Value (GSH) Request string to sensor (BINARY)** | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x17 | Read shutter speed value |
| **Read Shutter Speed Value (GSH) Response string from sensor (BINARY)** | | | |
| 1 - 4 | Unsigned int | 0x0B | Telegram length |
| 5 | Unsigned Char | 0x17 | Read shutter speed value |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | Unsigned int | 0xXX | Shutter speed value |
| Additional information: | | | |
| Accepted in run mode: | | Yes | |
| Accepted in configuration mode:: | | No | |
| Accepted when Ready is low: | | Yes | |
| Status of Ready signal during processing: | | No change | |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) | |

## Set gain value (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Set gain value (SGA) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0A | Telegram length |
| 5 | Unsigned Char | 0x1B | Set gain value |
| 6 | Unsigned Char | 0xXX | 0: Temporary<br>1: Permanent |
| 7 - 10 | Unsigned int | 0xXX | Gain value * 1000 |
| **Set gain value (SGA) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0B | Telegram length |
| 5 | Unsigned Char | 0x1B | Set gain value |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | Unsigned int | 0xXX | Current gain value (value *1000) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

## Read gain value (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Read gain value (GGA) Request string to sensor (BINARY)** | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x1C | Read gain value |
| **Read gain value (GGA) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0B | Telegram length |
| 5 | Unsigned Char | 0x1C | Read gain value |
| 6 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 7 | | 0xXX | |
| 8 - 11 | Unsigned int | 0xXX | Current gain value * 1000 |
| Additional information: | | | |
| Accepted in run mode: | | Yes | |
| Accepted in configuration mode: | | No | |
| Accepted when Ready is low: | | Yes | |
| Status of Ready signal during processing: | | No change | |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) | |

## Set parameter (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Set parameters (SPP/SPT) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length = 9 bytes + length of the selected parameter |
| 5 | Unsigned Char | 0xXX | 0x05: Set parameter permanently<br>0x06: Set parameter temporarily |
| 6 | Unsigned Char | 0xXX | Detector no., XX = 1- n |
| 7 | Unsigned Char | 0xXX | Command Set Reference string / value, see table Overview detector Parameter |
| 8 - 9 | Unsigned Short | 0xXX | Length of new reference string / value (n), see table Overview of detector Parameter |
| 10 ... n | Unsigned Char | 0xXX | Reference string / value |
| **Set parameters (SPP/SPT) Response string from sensor (BINARY)** | | | |
| (may be delayed up to 4-5 seconds) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x08 | Telegram length + length of the selected parameter in bytes |
| 5 | Unsigned Char | 0xXX | 0x05: Set parameter permanently<br>0x06: Set parameter temporarily |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |

| | | | |
|---|---|---|---|
| 8 | Unsigned Char | 0xXX | Parameter type<br>0x00: I8<br>0x01: U8<br>0x02: I16<br>0x03: U16<br>0x04: I32<br>0x05: U32<br>0x06: I40<br>0x07: U40<br>0x08: Float<br>0x09: Double<br>0x0A: String<br>0x0B: Boolean<br>0x0C: Special signed8<br>0x0D: Undefined |

| | |
|---|---|
| Additional information: | |
| Accepted in run mode: | Yes |
| Accepted in configuration mode: | No |
| Accepted when Ready is low: | Yes |
| Status of Ready signal during processing: | Low |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) |

**Overview Detector Parameters (set / read)**

| Detector | Function | Value | Multiplier | Length |
|---|---|---|---|---|
| **Alignment** | | | | |
| Pattern matching<br>Contour matching | Threshold value Min. | 0x01 | 1000 | 4 |
| | Threshold value Max. | 0x02 | 1000 | 4 |
| | Result offset<br>0: "Off"<br>1: "Image plane (in pixels)"<br>2: "Align (2D)"<br>3: "Robot (3D)" | 0x1E | 1 | 1 |
| | Result offset Image plane: Pos. X | 0x1F | 1000 | 4 |
| | Result offset Image plane: Pos. Y | 0x20 | 1000 | 4 |
| | Result offset Image plane: angle | 0x21 | 1000 | 4 |
| | Result offset<br>Align (2D), Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z | 0x22 | 1000 | 24 (6 * 4 bytes per value) |
| | Calculate Result offset* with transmitted position<br>• Align (2D): Pos. X, Pos. Y, 0, 0, 0, Angle Z<br>• Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z<br>*A valid position for the detector must be available | 0x23 | 1000 | 24 (6 * 4 bytes per value) |
| Edge detector | Probe 1: Transition<br>0: Any<br>1: Dark to light<br>2: Light to dark | 0x65 | 1 | 1 |
| | Probe 2: Transition<br>0: Any<br>1: Dark to light<br>2: Light to dark | 0x66 | 1 | 1 |

| Detector | Function | Value | Multiplier | Length |
|---|---|---|---|---|
| | Probe 3: Transition<br>0: Any<br>1: Dark to light<br>2: Light to dark | 0x67 | 1 | 1 |
| | Probe 1: Threshold value Min. | 0x68 | 1000 | 4 |
| | Probe 2: Threshold value Min. | 0x69 | 1000 | 4 |
| | Probe 3: Threshold value Min. | 0x6A | 1000 | 4 |
| **Detector** | | | | |
| Pattern matching<br>Contour<br>Contour 3D | Threshold value Min. | 0x01 | 1000 | 4 |
| | Threshold value Max. | 0x02 | 1000 | 4 |
| | Result offset<br>0: "Off"<br>1: "Image plane (in pixels)"<br>2: "Align (2D)"<br>3: "Robot (3D)" | 0x1E | 1 | 1 |
| | Result offset Image plane: Pos. X | 0x1F | 1000 | 4 |
| | Result offset Image plane: Pos. Y | 0x20 | 1000 | 4 |
| | Result offset Image plane: angle | 0x21 | 1000 | 4 |
| | Result offset<br>Align (2D), Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z | 0x22 | 1000 | 24 (6 * 4 bytes per value) |
| | Calculate Result offset* with transmitted position<br>• Align (2D): Pos. X, Pos. Y, 0, 0, 0, Angle Z<br>• Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z<br>*A valid position for the detector must be available | 0x23 | 1000 | 24 (6 * 4 bytes per value) |
| Gray | Threshold value Min. | 0x01 | 1000 | 4 |
| | Threshold value Max. | 0x02 | 1000 | 4 |
| | Grayscale value Min. | 0x65 | 1000 | 4 |

| Detector | Function | Value | Multiplier | Length |
| --- | --- | --- | --- | --- |
| | Grayscale value Max. | 0x66 | 1000 | 4 |
| | Invert grayscale value | 0x67 | 1 | 4 |
| Contrast Brightness | Threshold value Min. | 0x01 | 1000 | 4 |
| | Threshold value Max. | 0x02 | 1000 | 4 |
| Caliper | Threshold value Distance Min. | 0x65 | 1000 | 4 |
| | Threshold value Distance Max. | 0x66 | 1000 | 4 |
| | Invert distance threshold value | 0x67 | 1 | 1 |
| | Distance mode<br>0: Minimum<br>1: Maximum<br>2: Mean<br>3: Median<br>4: Smallest opposite<br>5: Largest opposite | 0x68 | 1 | 1 |
| | Probe 1: Threshold value Min. | 0x69 | 1000 | 4 |
| | Probe 2: Threshold value Min. | 0x6A | 1000 | 4 |
| | Probe 1: Smoothing | 0x6B | 1000 | 4 |
| | Probe 2: Smoothing | 0x6C | 1000 | 4 |
| | Probe 1: Transition<br>0: Any<br>1: Dark to light<br>2: Light to dark | 0x6D | 1 | 1 |
| | Probe 2: Transition<br>0: Any<br>1: Dark to light<br>2: Light to dark | 0x6E | 1 | 1 |
| | Probe 1: Number of search stripes | 0x6F | 1 | 1 |
| | Probe 2: Number of search stripes | 0x70 | 1 | 4 |
| BLOB | Grayscale value Min. | 0x65 | 1000 | 4 |
| | Grayscale value Max. | 0x66 | 1000 | 4 |
| | Invert grayscale value<br>0: not inverted<br>1: inverted | 0x67 | 1 | 1 |
| | Threshold value Number of BLOBs Min. | 0x78 | 1 | 1 |

| Detector | Function | Value | Multiplier | Length |
| --- | --- | --- | --- | --- |
| | Threshold value Number of BLOBs Max. | 0x79 | 1 | 1 |
| | Invert number threshold value<br>0: not inverted<br>1: inverted | 0x7A | 1 | 1 |
| | Number of set features (read only) | 0x7B | 1 | 1 |
| | Selection of a feature from the list | 0x7C | 1 | 1 |
| | Feature threshold value Min. | 0x7D | 1000 | 4 |
| | Feature threshold value Max. | 0x7E | 1000 | 4 |
| | Invert feature threshold value | 0x7F | 1 | 1 |
| Barcode<br>Datacode<br>OCR | Reference string | 0x65 | - | n (length of string) |
| | Reference string | 0x65 | - | n (length of string) |
| | Reference string | 0x65 | - | n (length of string) |
| Color Value<br>Color Value | Color space (read only) | 0x15 | 0x00 = RGB<br>0x01 = HSV<br>0x02 = LAB | 1 |
| | Channel selection (read only) | 0x16 | Bit field one digit per color channel | 1 |
| | Color channel 1: Threshold value Min. | 0x65 | 1000 | 4 |
| | Color channel 1: Threshold value Max. | 0x66 | 1000 | 4 |
| | Color channel 1: Invert threshold value | 0x67 | 1 | 1 |
| | Color channel 2: Threshold value Min. | 0x68 | 1000 | 4 |
| | Color channel 2: Threshold value Max. | 0x69 | 1000 | 4 |
| | Color channel 2: Invert threshold value | 0x6A | 1 | 1 |

| Detector | Function | Value | Multiplier | Length |
|---|---|---|---|---|
| | Color channel 3: Threshold value Min. | 0x6B | 1000 | 4 |
| | Color channel 3: Threshold value Max. | 0x6C | 1000 | 4 |
| | Color channel 3: Invert threshold value | 0x6D | 1 | 1 |
| Color List | Color space (read only) | 0x15 | 0 = RGB<br>1 = HSV<br>2 = LAB | 3 |
| | Channel selection (read only) | 0x16 | Bit field one digit per color channel | 4 |
| | Color distance threshold value | 0x65 | 1000 | N |
| | Set color distance threshold value active | 0x66 | 1 | N |
| | Number of colors in list | 0x67 | 1 | N |
| | Selection of a color from the list | 0x68 | 1 | N |
| | Color value of the selected color (color channel 1, color channel 2, color channel 3, color channel 4 [constantly 0]) | 0x69 | 1000 | 32 |
| Busbar<br>Wafer | Threshold value Min. | 0x01 | 1000 | N |
| | Threshold value Max. | 0x02 | 1000 | N |
| Result processing | Name of the active expression | 0x7A | - | n (length of string) |
| | Current expression | 0x7V | - | n (length of string) |

## Read parameter (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Read parameter (GPA) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x0A | Get parameter |
| 6 | Unsigned Char | 0xXX | Detector no., XX = 1- n |
| 7 | Unsigned Char | 0xXX | Command Set Reference string / value, see table Overview detector Parameter |
| **Read parameter (GPA) Response string from sensor (BINARY)** | | | |
| (may be delayed up to 4-5 seconds) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length = 10 Bytes + length of the selected parameter in Byte |
| 5 | Unsigned Char | 0x0A | Get parameter |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Parameter type string |
| 9 - 10 | Unsigned Short | 0xXX | Length of read parameter (n) |
| 11 ... n + n | Unsigned Char | 0xXX | Reference string / value |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

## Set search range (ROI) (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Set ROI (SRP/SRT) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length in bytes<br>24 bytes: circle<br>32 bytes: rectangle, ellipse, free form |
| 5 | Unsigned Char | 0xXX | 0x10: Set parameter temporarily<br>0x11: Set parameter permanently |
| 6 - 9 | Unsigned int | 0xXX | 19 bytes: circle<br>27 bytes: rectangle, ellipse, free form |
| 10 | Unsigned Char | 0xXX | Detector no. |
| 11 | Unsigned Char | 0xXX | Search range (ROI) Type<br>0x00: Search area (yellow)<br>0x01: Teach area (red)<br>0x02: Position control (blue) |
| 12 | Unsigned Char | 0xXX | Search range (ROI) Shape<br>0x01: Circle<br>0x02: Rectangle<br>0x03: Ellipse<br>0x04: Free shape |
| 13 - 16 | Unsigned int | 0xXX | ROI parameter: Center X (value in [px] * 1000) |
| 17 - 20 | Unsigned int | 0xXX | ROI parameter: Center Y (value in [px] * 1000) |
| 21 - 24 | Unsigned int | 0xXX | ROI parameter: half width or radius X (value in [px] * 1000) |
| | Only for ellipse / rectangle / free form: | | |
| 25 - 28 | Unsigned int | 0xXX | ROI parameter: half height or radius Y (value in pixels * 1000) |
| 29 - 32 | Unsigned int | 0xXX | ROI parameter: Angle in ° degree (value in ° [degrees] * 1000) |
| Set ROI (SRP/SRT) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 | Unsigned int | 0x07 | Telegram length |

| | | | |
|---|---|---|---|
| 5 | Unsigned Char | 0xXX | 0x10: Set parameter permanently<br>0x11: Set parameter temporarily |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |

| | |
|---|---|
| Additional information: | |
| Accepted in run mode: | Yes |
| Accepted in configuration mode: | No |
| Accepted when Ready is low: | Yes |
| Status of Ready signal during processing: | Low |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) |
| Parameter: | The parameters are given in the coordinate system of the Alignment and not in the coordinate system of the image. |

## Read search range (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Read search range (GRI) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x12 | Get ROI |
| 6 | Unsigned Char | 0xXX | Detector no. |
| 7 | Unsigned Char | 0xXX | Search range (ROI) Type<br>0x00: Search area (yellow)<br>0x01: Teach area (red)<br>0x02: Position control (blue) |
| **Read search range (GRI) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length |
| 5 | Unsigned Char | 0x12 | Get ROI |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | Unsigned int | 0xXX | Search range (ROI) Info Length in bytes from Byte 8 |
| 12 | Unsigned Char | 0xXX | Detector no. |
| 13 | Unsigned Char | 0xXX | Search range (ROI) Type<br>0x00: Search area (yellow)<br>0x01: Teach area (red)<br>0x02: Position control (blue) |
| 14 | Unsigned Char | 0xXX | Search range (ROI) Shape<br>0x01: Circle<br>0x02: Rectangle<br>0x03: Ellipse<br>0x04: Free form |
| 15 - 18 | Unsigned int | 0xXX | ROI parameter: Center X (value in pixels * 1000) |
| 19 - 22 | Unsigned int | 0xXX | ROI parameter: Center Y (value in pixels * 1000) |

| | | | |
|---|---|---|---|
| 23 -26 | Unsigned int | 0xXX | ROI parameter: Half width / radius X (value in pixels [px] * 1000) |
| | Only for ellipse / rectangle / free form: | | |
| 27 - 30 | Unsigned int | 0xXX | ROI parameter: Half height / radius Y (value in pixels [px] * 1000) |
| 31 - 34 | Unsigned int | 0xXX | ROI parameter: Angle in ° (value in ° * 1000) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

## Read job list (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Read job list (GJL) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x14 | Read job list |
| **Read job list (GJL) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length |
| 5 | Unsigned Char | 0x14 | Read job list |
| 6 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0x01 | Constant |
| 9 | Unsigned Char | 0xXX | Number of jobs |
| 10 | Unsigned Char | 0xXX | Active job number |
| **NOTE:** The following byte sequence is repeated for each job from 1 to "Number of jobs". The byte numbers shift accordingly. | | | |
| 11 | Unsigned Char | 0xXX | Number of subsequent bytes. This can be used to specify a unique name for job n. |
| 11 ... n | Char | 0xXX | From this position, the name for job n follows in the specified length. |
| n + 1 ... n + 3 | Unsigned Char | 0xXX | Number of subsequent bytes. A description for job n can be specified. |
| n + 4 ... m | Char | 0xXX | From this position, the description for Job 1 follows in the specified length. |
| m + 1 ... m+ 3 | Unsigned Char | 0xXX | Number of subsequent bytes. This can be used to specify a unique name for the author of job n. |
| m + 4 ... k | Char | 0xXX | From this position, the name for the author of job n follows in the specified length. |

| | | | |
|---|---|---|---|
| k + 1 ... k + 7 | Unsigned int | 0xXX | Date of creation of Job n (7 bytes) |
| k + 8 ... k + 14 | Unsigned int | 0xXX | Date of last modification of job n (7 bytes) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

## Read detector list (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Read detector list (GDL) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x15 | Read detector list |
| **Read detector list (GDL) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length |
| 5 | Unsigned Char | 0x18 | Read detector list |
| 6 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Job number of current job |
| 9 | Unsigned Char | 0xXX | Number of detectors in the current job |
| | **NOTE:** The following byte sequence is repeated for each detector in the job. The byte numbers shift accordingly. | | |
| 10 | Unsigned Char | 0xXX | Number of subsequent bytes. This allows a unique name for the detector n to be specified. |
| 11 ... n | Unsigned Char | 0xXX | From this position, the name for detector n follows, in the given length. |

| | | | |
|---|---|---|---|
| n + 1 ... n + 2 | Unsigned Char | 0xXX | Detector<br>0x01: Pattern matching<br>0x04: Contour<br>0x05: Gray<br>0x06: Contrast<br>0x07: Brightness<br>0x0A: Wafer<br>0x0B: OCR<br>0x0D: Datacode<br>0x0E: Barcode<br>0x11: Busbar<br>0x12: Color Value<br>0x13: Color Area<br>0x14: Color List<br>0x15: Caliper<br>0x16: BLOB |
| Additional information: | | | |
| Accepted in run mode: | | Yes | |
| Accepted in configuration mode: | | No | |
| Accepted when Ready is low: | | Yes | |
| Status of Ready signal during processing: | | No change | |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) | |

## Teach detector (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Teach detector (TED) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x08 | Telegram length |
| 5 | Unsigned Char | 0x18 | Teach detector |
| 6 | Unsigned Char | 0xXX | 0x00: Alignment<br>≥ 0x01: Detector selection |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary<br>0x01: Permanent |
| 8 | Unsigned Char | 0xXX | 0x00: No trigger, teach-in with next image acquis-ition<br>0x01: Trigger is executed for teach-in |
| **Teach detector (TED) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x00 | Telegram length |
| 5 | Unsigned Char | 0x18 | Teach detector |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

## Set trigger delay (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Set trigger delay (STD) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no.. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x08 | Telegram length |
| 5 | Unsigned Char | 0x27 | Set trigger delay |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary<br>0x01: Permanent |
| 8 - 11 | Unsigned int | 0xXX | Trigger delay<br>in msec (max. 3000 msec)<br>in encoder steps (max. 65535 steps) |
| **Set trigger delay (STD) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x27 | Set trigger delay |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

## Get trigger delay (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Get trigger delay (GTD) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no.. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x06 | Telegram length |
| 5 | Unsigned Char | 0x28 | Get trigger delay |
| 6 | Unsigned Char | 0xX1 | Request version |
| **Get trigger delay (GTD) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0B | Telegram length |
| 5 | Unsigned Char | 0x28 | Get trigger delay |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | Unsigned int | 0xXX | Trigger delay<br>in msec (max. 3000 msec)<br>in encoder steps (max. 65535 steps) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

## Save job permanently (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Save Job Permanently (SJP) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x0D | Saving of all telegrams that were previously executed temporarily |
| **Save Job Permanently (SJP) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0B | Telegram length |
| 5 | Unsigned Char | 0x0D | Save job permanently |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

### 11.5.4 Calibration

## Calibration: Initialization (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Initialization (CCD) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x1F | Initialize (Calibration: Clear Data) |
| **Calibration: Initialization (CCD) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x1F | Initialize (Calibration: Clear Data) |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| **Additional information:** | | | |
| Accepted in run mode: | | Yes | |
| Accepted in configuration mode: | | No | |
| Accepted when Ready is low: | | Yes | |
| Status of Ready signal during processing: | | No change | |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) | |

## Calibration: Add world point (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Add world point (CAW) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x15 | Telegram length |
| 5 | Unsigned Char | 0x26 | Calibration: Add world point |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x01: Fiducials only Calibration plate (Robotics)<br>0x04: World point and pixel Point pair list (Robotics) |
| 9 - 10 | Unsigned Short | 0x00 | Constant (2 bytes) |
| 11 - 14 | Unsigned int | 0xXX | World X (in mm *1000) |
| 15 - 18 | Unsigned int | 0xXX | World Y (in mm *1000) |
| 19 - 22 | Unsigned Char | 0x00 | Constant (4 bytes) |
| **Calibration: Add world point (CAW) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x11 | Telegram length |
| 5 | Unsigned Char | 0x26 | Calibration: Add world point |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 9 | Unsigned Short | 0xXX | Current number of points |
| 10 -13 | Unsigned int | 0xXX | Image point X |
| 14 - 17 | Unsigned int | 0xXX | Image point Y |
| Additional information: | | | |
| Accepted in run mode: | | Yes | |
| Accepted in configuration mode: | | No | |
| Accepted when Ready is low: | | Yes | |
| Status of Ready signal during processing: | | No change | |

| | |
|---|---|
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) |

Note: For the CAW request, the overall job result must be positive.

## Calibration: Point pair list (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Point pair list (CCL) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x06 | Telegram length |
| 5 | Unsigned Char | 0x1E | Calibration: Point pair list |
| 6 | Unsigned Char | 0xXX | 0x00: Temporary<br>0x01: Permanent |
| **Calibration: Point pair list (CCL) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x19 | Telegram length |
| 5 | Unsigned Char | 0x1E | Calibration: Point pair list |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 9 | Unsigned Short | 0xXX | Current highest point pair index |
| 10 - 13 | Unsigned int | 0xXX | Deviation calibration, RMSE |
| 14 - 17 | Unsigned int | 0xXX | Deviation calibration, mean |
| 18 - 21 | Unsigned int | 0xXX | Deviation calibration, max. |
| 22 - 25 | Unsigned int | 0xXX | Deviation calibration, min. |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

## Calibration: Validate point pair list (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Validate point pair list (CVL) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x20 | Calibration: Validate point pair list |
| **Calibration: Validate point pair list (CVL) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x19 | Telegram length |
| 5 | Unsigned Char | 0x20 | Calibration: Validate point pair list |
| 6 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 9 | Unsigned Short | 0xXX | Current highest point pair index |
| 10 - 13 | Unsigned int | 0xXX | Deviation calibration, RMSE |
| 14 - 17 | Unsigned int | 0xXX | Deviation calibration, mean |
| 18 - 21 | Unsigned int | 0xXX | Deviation calibration, max. |
| 22 - 25 | Unsigned int | 0xXX | Deviation calibration, min. |
| | | | |
| Accepted in run mode: | | Yes | |
| Accepted in configuration mode: | | No | |
| Accepted when Ready is low: | | Yes | |
| Status of Ready signal during processing: | | No change | |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) | |

## Calibration: Calibration plate (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Calibration: Calibration Plate (CCP) Request string to sensor (BINARY)** | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x09 | Telegram length |
| 5 | Unsigned Char | 0x24 | Calibration: Calibration plate |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary<br>0x01: Permanent |
| 8 | Unsigned Char | 0xXX | 0x00: No fiducials are used. Origin of Measuring coordinate system identical to origin of Calibration Plate Coordinate System.<br>0x01: No fiducials are used. Measuring coordinate system identical with Camera coordinate system.<br>0x02: Uses world system, fiducial Job<br>0x03: Uses world system, fiducial Command CAW |
| 9 | Unsigned Char | 0xXX | 0x00: Calibration internal and external sensor para-meters<br>0x01: Validation of calibration<br>0x02: Calibration internal sensor parameters<br>0x05: Calibration Transformation Measuring coordinate system |
| **Calibration: Calibration Plate (CCP) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x3D | Telegram length |
| 5 | Unsigned Char | 0x24 | Calibration: Calibration plate |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 9 | Unsigned Short | 0xXX | Number of currently detected calibration points |
| 10 - 13 | Unsigned int | 0xXX | Deviation calibration, RMSE |
| 14 - 17 | Unsigned int | 0xXX | Deviation calibration, mean |
| 18 - 21 | Unsigned int | 0xXX | Deviation calibration, max. |

| | | | |
|---|---|---|---|
| 22 - 25 | Unsigned int | 0xXX | Deviation calibration, min. |
| 26 -29 | Unsigned int | 0xXX | CPF_MF X (in user unit * 1000) |
| 30 - 33 | Unsigned int | 0xXX | CPF_MF Y (in user unit * 1000) |
| 34 - 37 | Unsigned int | 0x00 | CPF_MF Z (in user unit * 1000) |
| 38 - 41 | Unsigned int | 0x00 | CPF_MF Angle X (in degrees * 1000) |
| 42 - 45 | Unsigned int | 0x00 | CPF_MF Angle Y (in degrees * 1000) |
| 46 - 49 | Unsigned int | 0xXX | CPF_MF Angle Z (in degrees * 1000) |
| 50 - 53 | Unsigned int | 0xXX | Deviation fiducials, mean |
| 54 - 57 | Unsigned int | 0xXX | Deviation fiducials, max. |
| 58 - 61 | Unsigned int | 0xXX | Deviation fiducials, min. |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

## Calibration: Set fiducial (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Set fiducial (CSF) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x2B | Calibration: Set fiducial |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary<br>0x01: Permanent |
| **Calibration: Set fiducial (CSF) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x2B | Telegram length |
| 5 | Unsigned Char | 0x2B | Calibration: Set fiducial |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | Unsigned int | 0xXX | X value |
| 12 - 15 | Unsigned int | 0xXX | Y value |
| 16 - 19 | Unsigned int | 0xXX | Z value |
| 20 - 23 | Unsigned int | 0xXX | Angle X value |
| 24 - 27 | Unsigned int | 0xXX | Angle Y value |
| 28 - 31 | Unsigned int | 0xXX | Angle Z value |
| 32 - 35 | Unsigned int | 0xXX | Deviation fiducials, mean |
| 36 - 39 | Unsigned int | 0xXX | Deviation fiducials, max. |
| 40 - 43 | Unsigned int | 0xXX | Deviation fiducials, min. |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

## Calibration: Add image (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Add image (CAI) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x22 | Telegram length<br>34 (0x22) Bytes |
| 5 | Unsigned Char | 0x34 | Calibration: Add image |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | Mode<br>0x01: Multi-image calibration<br>0x02: Hand-Eye calibration (Robotics)<br>0x03: Base-Eye calibration (Robotics) |
| 8 | Unsigned Short | 0x00 | Constant |
| 9 | Unsigned Char | 0xXX | Define Measurement plane<br>0x00: Do not use image to define Measurement plane<br>0x01: Use image to define Measurement plane |
| 10 | Unsigned Char | 0xXX | "Robot: Order of rotation"<br>0x00: Use order of rotation specified in job<br>0x01: Yaw-Pitch-Roll (e.g. Stäubli)<br>0x02: Roll-Pitch-Yaw (e.g. Kuka, Fanuc, Hanwha, ABB**, UR**)<br>** when using the corresponding conversion function |
| 11-14 | Unsigned Char | | Pose_TCP Pos. X<br>(in user unit * 1000) |
| 15-18 | Unsigned Char | | Pose_TCP Pos. Y<br>(in user unit * 1000) |
| 19-22 | Unsigned Char | | Pose_TCP Pos. Z<br>(in user unit * 1000) |
| 23-26 | Unsigned Char | | Pose_TCP Angle X<br>(in degrees * 1000) |
| 27-30 | Unsigned Char | | Pose_TCP Angle Y<br>(in degrees * 1000) |

| | | | |
|---|---|---|---|
| 31-34 | Unsigned Char | | Pose_TCP Angle Z (in degrees * 1000) |
| **Calibration: Add image (CAI) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1-4 | Unsigned int | 0x0A | Telegram length |
| 5 | Unsigned Char | 0x34 | Calibration: Add image |
| 6-7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Short | 0xXX | Current number of images in list |
| 9-10 | Unsigned Char | 0xXX | Total number of detected points |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | Yes |
| Accepted when Ready is low: | | | No |

## Calibration: Multi-image (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Multi-image (CMP) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x09 | Telegram length 9 (0x09) Bytes |
| 5 | Unsigned Char | 0x35 | Calibration: Multi-image |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary<br>0x01: Permanent |
| 8 | Unsigned Char | 0xXX | Origin of the world coordinate system:<br>0x00: World coordinate system identical with the Calibration Plate Coordinate System (center of the plate).<br>0x01: Origin of World coordinate system so that it is identical to origin of Image Coordinate System (upper left pixel).<br>0x02: (only for Calibration plate (Robotics)) Use World coordinate system of fiducials, as specified in the job file.<br>0x03: (only for Calibration plate (Robotics)) Use World coordinate system of fiducials as set in request CAW. |
| 9 | Unsigned Char | 0xXX | Mode<br>0x00: Calibration (internal and external parameters)<br>0x01: Validieren (vorhandene Kalibrierung verwenden; mindestens ein Kalibrierpunkt wird hinzugefügt. Über Rückprojektion kann zurückgeschlossen werden, ob der Punkt zur aktuellen Kalibrierung passt, oder verschoben ist)<br>0x02: Calibration (internal parameters only)<br>0x03: Calibration (external parameters only using new internal parameters)<br>0x04: Calibration (external parameters only)<br>0x05: Calibrate Measurement plane only (CPF_MF) |

| Calibration: Multi-image (CMP) Response string from sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1-4 | Unsigned int | 0x1D | Telegram length 29 (0x1D) Bytes |
| 5 | Unsigned Char | 0x35 | Calibration: Multi-image |
| 6-7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Field of view coverage (%)<br>0x00: no coverage<br>0x64: Coverage 100% |
| 9-10 | Unsigned Short | 0xXX | Total number of detected points |
| 11 | Unsigned Char | 0xXX | Number of images used |
| 12 | Unsigned Char | 0xXX | Number of invalid images |
| 13 | Unsigned Char | 0xXX | Sufficient tilt between calibration plate poses<br>0x00: not sufficient<br>0x01: sufficient |
| 14-17 | Unsigned int | 0xXX | Deviation calibration plate RMSE [px] |
| 18-21 | Unsigned int | 0xXX | Deviation calibration plate Max. [px] |
| 22-25 | Unsigned int | 0xXX | Deviation fiducials, RMSE (in user unit * 1000) |
| 26-29 | Unsigned int | 0xXX | Deviation fiducials, max. [px] |
| Additional information: | | | |
| Accepted in run mode: | | Yes | |
| Accepted in configuration mode: | | No | |
| Accepted when Ready is low: | | Yes | |

## Calibration: Robotics multi-image (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Calibration: Robot multi-picture (CRP) Request string to sensor (BINARY)** | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x09 | Telegram length (bytes)<br>9 Byte |
| 5 | Unsigned Char | 0x36 | Calibration: Calibration plate robotics |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary<br>0x01: Permanent |
| 8 | Unsigned Char | 0xXX | Origin of the world coordinate system:<br>0x04: Set origin of coordinate system equal to Robot Coordinate System |
| 9 | Unsigned Char | X | Mode<br>0x00: Calibration (internal and external parameters)<br>0x01: Validieren (vorhandene Kalibrierung verwenden; mindestens ein Kalibrierpunkt wird hinzugefügt. Über Rückprojektion kann zurückgeschlossen werden, ob der Punkt zur aktuellen Kalibrierung passt, oder verschoben ist)<br>0x02: Calibration (internal parameters only)<br>0x03: Calibration (external parameters only using new internal parameters)<br>0x04: Calibration (external parameters only)<br>0x05: Calibrate Measurement plane only (CPF_MF)<br>0x06: Hand-Eye calibration (Robotics) / Base-Eye calibration (Robotics) |
| **Calibration: Robot multi-picture (CRP) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1-4 | Unsigned int | 0x2C | Telegram length 44 (0x2C) Bytes |
| 5 | Unsigned Char | 0x36 | Calibration: Calibration plate robotics |

| | | | |
|---|---|---|---|
| 6-7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Field of view coverage<br>0x00: not sufficient<br>0x01: sufficient |
| 9-10 | Unsigned Short | 0xXX | Total number of detected points |
| 11 | Unsigned Char | 0xXX | Number of images used |
| 12 | Unsigned Char | 0xXX | Number of invalid images |
| 13-16 | Unsigned int | 0xXX | Deviation calibration plate RMSE [px] |
| 17-20 | Unsigned int | 0xXX | Deviation calibration plate Max. [px] |
| 21-24 | Unsigned int | 0xXX | Deviations calibration plate pose Translation RMSE (in user unit * 1000) |
| 25-28 | Unsigned int | 0xXX | Deviations calibration plate pose Translation Max. (in user unit * 1000) |
| 29-32 | Unsigned int | 0xXX | Deviations calibration plate pose Rotation RMSE (in degrees * 1000) |
| 33-36 | Unsigned int | 0xXX | Deviations calibration plate pose Rotation Max. (in degrees * 1000) |
| Additional information: | | | |
| Accepted in run mode: | | Yes | |
| Accepted in configuration mode: | | No | |
| Accepted when Ready is low: | | Yes | |

## Calibration: Copy Calibration (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Calibration: Copy calibration (CCC) Request string to sensor (BINARY)** | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x09 | Telegram length |
| 5 | Unsigned Char | 0x25 | Calibration: Copy Calibration |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0x01 | Constant |
| 8 | Unsigned Char | 0xXX | Destination<br>0 : Copy to all jobs<br>>0: Copy to specified job |
| 9 | Unsigned Char | 0xXX | 0: Always copy when the calibration is active.<br>1: Only copy if the calibration method is the same. |
| **Calibration: Copy calibration (CCC) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x08 | Telegram length |
| 5 | Unsigned Char | 0x25 | Calibration: Copy Calibration |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | 00: Successful<br>>0 : Job number at which the error occurs. |
| Additional information: | | | |
| Accepted in run mode: | | Yes | |
| Accepted in configuration mode: | | No | |
| Accepted when Ready is low: | | Yes | |
| Status of Ready signal during processing: | | No change | |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) | |

## Calibration: Set parameter (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Set parameter (CSP) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length in Byte, 16 Bytes (0x13) + length of selected parameter |
| 5 | Unsigned Char | 0x29 | Calibration: Set parameter |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary<br>0x01: Permanent |
| 8 | Unsigned Char | 0xXX | Parameter number, see table Calibration parameters for telegrams CSP and CGP |
| 9 - 12 | Unsigned int | 0xXX | Length of the following data |
| 13 ... n | Unsigned Char | 0xXX | Parameter value, see table Calibration parameters for telegrams CSP and CGP |
| **Calibration: Set parameter (CSP) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x29 | Calibration: Set parameter |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Calibration parameters: see table Calibration parameters for telegrams CSP and CGP

## Calibration: Read parameter (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| Calibration: Read parameter (CGP) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x2A | Calibration: Read parameter |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | Parameter number (Page 241) |
| **Calibration: Read parameter (CGP) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length in bytes, 12 bytes (0x0C) + length of selected parameter |
| 5 | Unsigned Char | 0x2A | Calibration: Read parameter |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Parameter number (Page 241) |
| 9 - 12 | Unsigned int | 0xXX | Length of the following data |
| 13 ... n | Unsigned Char | 0xXX | Parameter value (Page 241) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

### Calibration parameters for telegrams CSP and CGP

| Parameter description | Parameter number | Parameter value | Length | Calibration status after CSP |
|---|---|---|---|---|
| Status calibration | 0x01 | 0x00: Invalid<br>0x01: Valid | 1 byte | –* |
| Selection of calibration method | 0x02 | 0x00: None<br>0x02: Point pair list (Robotics)<br>0x03: Calibration plate (Measurement)<br>0x04: Calibration plate (Robotics)<br>0x05: Hand-Eye calibration (Robotics)<br>0x06: Base-Eye calibration (Robotics) | 1 byte | invalid |
| User unit | 0x04 | 0x00: Millimeter [mm]<br>0x01: Centimeter [cm]<br>0x02: Meter [m]<br>0x03: Inch ["]<br>0x04: Arbitrary unit [au] | 1 byte | no change |
| Internal parameters | 0x0A | Focal length (in mm *1000)<br>Kappa (*1000)<br>Pixel pitch X (in µm * 1000)<br>Pixel pitch Y (in µm * 1000)<br>Coordinate origin X (in pixels * 1000)<br>Coordinate origin Y (in pixels * 1000)<br>Image size X (number of pixels)<br>Image size Y (number of pixels) | 0x20 (8 * 4 bytes per value) | –* |
| Reference Camera- to Measuring coordinate system (CF_MF) | 0x0B | Translation X, Y, Z (in user unit * 1000)<br>Angle X, Y, Z (in degrees * 1000) | 0x18 (6 * 4 bytes per value) | –* |
| Reference Camera- to Calibration Plate Coordinate System (CF_CPF) | 0x0C | Translation X, Y, Z (in user unit * 1000)<br>Angle X, Y, Z (in degrees * 1000) | 0x18 (6 * 4 bytes per value) | –* |

| Parameter description | Parameter number | Parameter value | Length | Calibration status after CSP |
|---|---|---|---|---|
| Reference Robot- to Camera coordinate system (RF_CF) | 0x0D | Translation X, Y, Z (in user unit * 1000)<br>Angle X, Y, Z (in degrees * 1000) | 0x18<br>(6 * 4 bytes per value) | –* |
| Reference Calibration plate- to Measuring coordinate system (CPF_MF) | 0x0E | Translation X, Y, Z (in user unit * 1000)<br>Angle X, Y, Z (in degrees * 1000) | 0x18<br>(6 * 4 bytes per value) | –* |
| Reference Robot- to Measuring coordinate system (RF_MF) | 0x0F | Translation X, Y, Z (in user unit * 1000)<br>Angle X, Y, Z (in degrees * 1000) | 0x18<br>(6 * 4 bytes per value) | –* |
| Reference TCP- to Camera coordinate system(TCP_CF) | 0x10 | Translation X, Y, Z (in user unit * 1000)<br>Angle X, Y, Z (in degrees * 1000) | 0x18<br>(6 * 4 bytes per value) | –* |
| Reference robot- to TCP coordinate system (RF_TCP) | 0x11 | Translation X, Y, Z (in user unit * 1000)<br>Angle X, Y, Z (in degrees * 1000) | 0x18<br>(6 * 4 bytes per value) | no change |
| Z-shift Measurement plane | 0x15 | (in user unit * 1000) | 4 bytes | no change |
| Focal length in [mm] | 0x16 | [mm * 1000] | 4 bytes | invalid (CSP for C-Mount only) |
| Calibration plate type | 0x17 | Character string with name of the description file | n | invalid |
| Fiducial 1 | 0x18 | Translation X, Y, Z (in user unit * 1000) | 0x0C<br>(3* 4 bytes per value) | invalid |
| Fiducial 2 | 0x19 | | | |
| Fiducial 3 | 0x1A | | | |
| Fiducial 4 | 0x1B | | | |

| Parameter description | Parameter number | Parameter value | Length | Calibration status after CSP |
|---|---|---|---|---|
| Number of existing calibration plate types | 0x25 | Request - Selection of type:<br>0x00: All<br>0x01: Measurement<br>0x02: Robotics<br>Response:<br>Number of plates | Request: 1<br>Response:<br>2 | –* |
| Available calibration plate types (file names) | 0x26 | Request - Selection of type:<br>0x00: All<br>0x01: Measurement<br>0x02: Robotics<br>Request - Index:<br>0: All file names<br>>0: Index selection<br>Response:<br>File names of Calibration plates | Request: 1<br>Response:<br>5 (String) | –* |
| Robot: Order of rotation | 0x27 | "Robot: Order of rotation"<br>0x00: Use order of rotation specified in job<br>0x01: Yaw-Pitch-Roll (e.g. Stäubli)<br>0x02: Roll-Pitch-Yaw (e.g. Kuka, Fanuc, Hanwha, ABB**, UR**)<br>** when using the corresponding conversion function | 1 byte | invalid |
| Average sensor resolution | 0x29 | Value (in user unit/pixel * 1000) | 4 bytes | –* |

* CSP not possible (parameter read-only, cannot be set).

### 11.5.5 Visualization

## Get image (BINARY)

Telegrams: Availability and supported interfaces (Page 91)

Overview telegrams (Page 87)

| **Get image (GIM) Request string to sensor (BINARY)** | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x06 | Telegram length |
| 5 | Unsigned Char | 0x03 | Get image |
| 6 | Unsigned Char | 0xXX | 0x00: Last image<br>0x01: Last failed image<br>0x02: Last good image |
| **Get image (GIM) Response string from sensor (BINARY)** | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length in bytes, 13 bytes (0x0D) + number of bytes depending on the image format<br>e.g. 00 04 B0 0D (Dez. 307213) |
| 5 | Unsigned Char | 0x03 | Get image |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Image type<br>0: Grayscale<br>3: Bayer Pattern_BG<br>When converting the color image from Bayer into RGB, the appropriate image type must be considered. |
| 9 | Unsigned Char | 0xXX | Image result<br>00: Failed image<br>01: Good image |
| 10 - 11 | Unsigned Short | 0xXX | Number of rows<br>e.g. 01 E0 = 480 |

| | | | |
|---|---|---|---|
| 12 - 13 | Unsigned Short | 0xXX | Number of columns<br>e.g. 02 80 = 640 |
| 14 ... n | Unsigned Char | 0xXX | Binary image data (rows * columns) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

### 11.5.6 Data output BINARY

Output data (BINARY), dynamically composed according to user settings in the software under: SensoConfig / Output / Telegram.

Basic string structure:

| <START> (((<OPTIONAL FIELDS> <PAYLOAD>))) <CHKSUM> <TRAILER> |
|---|

**NOTE:**
The length and data types of the payload are standard values. The factor and bit depth can be set via "Telegram" / "Payload".

**Output data (BINARY):**

| **<OPTIONAL FIELDS>** | | | | |
|---|---|---|---|---|
| **Parameter** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Selected fields | With this checkbox all selected fields are displayed. The checkbox "Selected fields" itself is not displayed. | 2 | The output sequence is from left to right and from top to bottom, i.e. one bit is set per active checkbox, starting with the lowest-value one. | All types |
| Telegram length | Number of characters including the characters for the telegram length itself. | 2 | Unsigned Short | All types |
| Status byte | Returns the Trigger mode. | 2 | 0x06 0x00 = Trigger; 0x05 0x00 = Free run | All types |

| <OPTIONAL FIELDS> | | | | |
|---|---|---|---|---|
| **Parameter** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Detector results | Output of overall result for each detector.<br>Byte 1<br>Bit 1 (LSB) = Global job result (1 = Pass, 0 = Fail)<br>Bit 2 = Boolean result Alignment only, Alignment inactive = True | 3 ... 35 | | All types |
| Digital outputs | Returns the logic gate result for each digital output. | N | Bytes 1 and 2: Number of active Outputs<br>Bytes 3 – n: Outputs, bit-coded | All types |
| log. Outputs | Returns the logic gate result for each logic output. | N | Byte 1 and byte 2: Number of active log. Outputs<br>Byte 3 – n All active logic outputs, | All types |
| Execution time | Returns the execution time for the last evaluation. | 4 | Signed integer | All types |
| Active job | Returns the job for the last evaluation. | 1 | Unsigned int U8 | All types |

**<PAYLOAD>**

**Overview of detector-specific payload - Values**

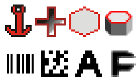
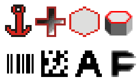
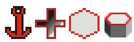
**GENERAL**

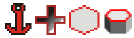
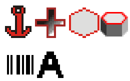
| <PAYLOAD> General | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| "All evaluations" counter | Total number of checks | 4 | Signed integer | GENERAL |
| Pass parts counter | Number of inspections with result "OK" | 4 | Signed integer | GENERAL |
| Fail parts counter | Number of inspections with result "Error" | 4 | Signed integer | GENERAL |
| Timeout | Indicates that the maximum cycle time has been exceeded. | 1 | BOOL | GENERAL |
| Recording | Indicates the number of image acquisition repetitions for the last evaluation Only in combination with repeat mode. | 4 | INT | GENERAL |
| String length | This field can be used to enter a constant string into the data output. | 0 ... 5 | STRING | GENERAL |

### Base values

| <PAYLOAD> Base values | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Score | [%] | 4 | Signed integer | All detectors |
| Overall result | Boolean detector result | 1 | BOOL | All detectors |
| Execution time | Execution time of individual detector in [msec]. | 4 | Signed integer | All detectors |

### Position

| <PAYLOAD> Position / location | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Pos. X | X coordinate for the found position, 1/1000 [user unit] | 4 | Signed integer |  |
| Pos. Y | Y coordinate for the found position, 1/1000 [user unit] | 4 | Signed integer |  |
| Pos. Z | Z coordinate of the found position, 1/1000 [user unit] | | Signed integer |  With Result off-set:  |
| Delta Pos. X | X position delta between the taught object and the found object, 1/1000 [user unit] | 4 | Signed integer |  |

| <PAYLOAD> Position / location | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Delta Pos. Y | Y position delta between the taught object and the found object, 1/1000 [user unit] | 4 | Signed integer |  |
| Delta Pos. Z | Z position delta between the taught object and the found object, 1/1000 [user unit] | 4 | Signed integer |  With Result off-set:  |
| Angle X | Orientation of the found object, relative to the X-axis, 1/1000 [°] | 4 | Signed integer |  With Result off-set:  |
| Angle Y | Orientation of the found object, relative to the Y-axis, 1/1000 [°] | 4 | Signed integer |  With Result off-set:  |
| Angle Z | Orientation of the found object, relative to the Z-axis, 1/1000 [°] | 4 | Signed integer |  |
| Angle (45) | Orientation of bound-ing box for found code [°], value range: -45° to 45° | 4 | Signed integer |  |
| Angle (180) | Orientation of object width (long axis) [°], Value range: -90° ... +90° 0° = East, coun-terclockwise | 4 | Signed integer |  |

| **<PAYLOAD> Position / location** | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Angle (360) | Orientation of object width (long axis) [°], Value range -180° ... +180°. 0° = East, counterclockwise | 4 | Signed integer | |
| Delta Angle X | Angle between taught object and found object, 1/1000 [°] | 4 | Signed integer | With Result offset: |
| Delta Angle Y | Angle between taught object and found object, 1/1000 [°] | 4 | Signed integer | With Result offset: |
| Delta Angle Z | Angle between taught object and found object, 1/1000 [°] | 4 | Signed integer | |
| Pose 3D (X, Y, Z, Angle X, Angle Y, Angle Z) | Coordinates of the found object, 1/1000 [user unit] Angle: 1/1000 degrees | 4 bytes per value each | Signed integer | With Result offset: |
| Delta Pose 3D (X, Y, Z, Angle X, Angle Y, Angle Z) | Delta coordinates of the found object, 1/1000 [user unit] Angle: 1/1000 degrees | 4 bytes per value each | Signed integer | With Result offset: |
| Position control | | 1 | BOOL | |

**Measurement**

| <PAYLOAD> Measurement | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Height | Height of geometric element [user unit], Height ≥ 0, height ≤ width | 4 | Signed integer | |
| Width | Width of geometric element [user unit] Width ≥ 0, width ≥ height | 4 | Signed integer | |
| Radius | Radius of fitted circle [user unit] | 4 | Signed integer | |
| Area | Area of BLOB without holes, 1/1000 [pixels] | 4 | Signed integer | |
| Area (incl. holes) | Area of BLOB including holes, 1/1000 [pixels] | 4 | Signed integer | |
| Distance | Calculated distance [user unit] | 4 | Signed integer | |

**Identification**

| <PAYLOAD> Identification | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| String length | Length of read code [bytes] | 4 | Signed integer | |

| <PAYLOAD> Identification | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| String length | Content of the read code.<br>Depending on the code, the string length may vary. If a fixed string length is desired, the minimum string length (detector-specific payload) and the maximum string length (detector settings) must be set to the same value (e.g. 127). | N | STRING | |
| String comparison | Content check for the read information. The content of the read information is checked on the basis of regular expressions<br>(see detector Datacode, Reference string tab) | 1 | BOOL | |
| Truncated | Code complete or truncated<br>0: Code complete<br>1: Code truncated | 1 | BOOL | |

### Identification - quality

| <PAYLOAD> Identification - Quality | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Quality - overall | Output of all Q parameters. Depending on the selected code type and standard. | 1 byte per value; separated by specified separator For 2D code parameter Q9 (mean light): 1...3 | Unsigned Char; for 2D Code Q9 (Meanlight) Unsigned Short | |
| Quality - individual | Output of individual quality values: Selection Q1-Q24 depending on the selected code type and standard.<br>Numbers: 1-4<br>Letters: A-F | 1 | Unsigned Char; for 2D Code Q9 (Meanlight) Unsigned Short | |
| Min. Quality | Used to check whether the minimum required quality is being met | 4 | Unsigned int | |

### Color

| <PAYLOAD> Color | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Color value:<br>• Red, green, blue<br>• Hue, saturation, lightness<br>• Luminance, a, b | Value for color parameter | 4 | Signed integer | |
| Color distance | Distance of the current color versus the taught-in color | 4 | Signed integer | |

### Counting / number

| <PAYLOAD> Counting / number | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Number of objects | Number of objects found [units] | 4 | Signed integer | |
| Number of valid objects | Number of valid objects found [units] | 4 | Signed integer | |
| Number of search stripes | Number of parallel search stripes into which the width of the search range is divided.<br>[units] | 4 | Signed integer | (Edge detector only) |

| **<PAYLOAD> Counting / number** | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Number of valid search stripes | Used to check whether the number of search stripes found falls within a specific range. [Good/Bad or units] | 4 | Signed integer | (Edge detector only) |
| Result vector | Vector containing the result (1/0) of the instances found | N | BOOL | |
| Too many BLOBs | | 1 | BOOL | |

**Extended**

| **<PAYLOAD> Extended** | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Scaling | Outputs the scaling range, 1/1000. Within the scaling range, scaled-up or scaled-down objects will be detected. Value range of 0.5 to 2 | 4 | Signed integer | (Contour matching only) |
| Eccentricity | Numerical eccent-ricity Value range of 0.0 to 1.0 | 4 | Signed integer | |

| <PAYLOAD> Extended | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Security | Output of the security values of the individual characters. The reliability value specifies how reliably the reader was able to interpret a character. Value range of 0 to 100 [%] | 4 | Signed integer | |
| Reference string met | The output string matches the reference string. | 1 | BOOL | |
| contrast | Code contrast Value range of 0 to 100 [%] | 4 | Signed integer | |
| Correction | Number of modules corrected by error corrections [units] | 4 | Signed integer | |
| Contour length | Number of pixels of outer contour, 1/1000 [pixels] | 4 | Signed integer | |
| Compactness | BLOB compactness (circle =1; other > 1). The more the shape of the BLOB deviates from a circle, the greater the compactness value will be. | 4 | Signed integer | |
| Center of gravity X | X coordinate of centroid, 1/1000 | 4 | Signed integer | |

| <PAYLOAD> Extended | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Center of grav-ity Y | Y coordinate of centroid, 1/1000 | 4 | Signed integer | |
| Gray scale value, average | Average gray scale value of all the pixels that belong to the BLOB. | 4 | Signed integer | |
| Min. threshold | Lower threshold for the binarization of the objects. 0...255 | 4 | Signed integer | |
| Max. threshold | Upper threshold for the binarization of the objects. 0...255 | 4 | Signed integer | |
| Inverted threshold | Specifies whether the range Min <-> Max is inverted.<br>P: inverted<br>F: not inverted | 1 | Unsigned Char | |
| Deviation, inside | Returns the largest deviation between the BLOB contour and the contour of the geometric ele-ment (deviation inside the fitted circle).<br>[user unit] | 4 | Signed integer | |
| Deviation, out-side | Returns the largest deviation between the BLOB contour and the contour of the geometric ele-ment (deviation out-side the fitted circle).<br>[user unit] | 4 | Signed integer | |

| **<PAYLOAD> Extended** | | | | |
|---|---|---|---|---|
| **Value** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Deviation, mean | Returns the mean of the absolute "inside" and "outside" deviation values between the BLOB contour and the contour of the geometric element. | 4 | Signed integer | |
| Axial ratio | Ratio of the long to the short axis (a / b) | 4 | Signed integer | |
| Face up / down, area | Face up / down position, based on: area, position indicated by sign, 1/1000 | 4 | Signed integer | |
| Result index | List index | 4 | Signed integer | |
| Search stripe distance | Calculated distance [user unit] / 1000 per pair of search stripes | 4 | Signed integer | |

| **<CHKSUM>** | | | | |
|---|---|---|---|---|
| **Parameter** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Check sum | XOR check sum of all bytes in the telegram. Is transmitted as the last byte. | 1 | Unsigned int | All types |

| <TRAILER> | | | | |
|---|---|---|---|---|
| **Parameter** | **Description** | **Length BINARY [Byte]** | **Data type** | **Available for** |
| Start | Characters appen-ded at the end of the string | 0 ... 8 | Unsigned int | All types |

**NOTE:**
If no calibration has been performed, all values refer to pixels.

All detector-specific data with decimal places is transmitted as integers (multiplied by 1000) and must accordingly be divided by 1000 after the data is received. The values are transferred in the format "Big-endian".

Example: "Score" value (BINARY protocol)

In SensoConfig/SensoView "Score" = 35 is displayed.
Via Ethernet, the following four bytes, for example, are received: 000,000,139,115
Formula for conversion: (Byte4*256 + Byte3) *65536 + Byte2*256 + Byte1 = Value

Because big-endian (from the sensor) is sent, the following applies:
000 = HiWordByte, 000 = HiLowByte, 139 = HiByte, 115 = LoByte
(0*256 + 0) * 65536 + (139 * 256) + 115 = 35699 / 1000 = 35.699 (= real score value)
Angle data or other negative values are represented in two's complement.

# We look ahead
Yesterday, today and in the future

SENSOPART

Germany
SensoPart Industriesensorik GmbH
79288 Gottenheim
Tel.: +49 7665 94769-0
info@sensopart.de

France
SensoPart France SARL
F-77420 Champs sur Marne
Tel.: +33 1 64 73 00 61
info@sensopart.fr

Great Britain
SensoPart UK Ltd.
Melton Mowbray, Leicestershire, LE13 0PB
Tel.: +44 1664 561539
uk@sensopart.com

USA
SensoPart Inc.
Perrysburg, OH 43551
Tel.: +1 866 282 7610
usa@sensopart.com

China
SensoPart (Shanghai)
Shanghai, 201803
Tel.: +86 216 901 7660
china@sensopart.cn

EN

068-14859-02