VOS Ident

1-D and 2-D Code Readers

Manual

Your automation, our passion.

PEPPERL+FUCHS

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

Worldwide

Pepperl+Fuchs Group

Lilienthalstr. 200

68307 Mannheim

Germany

Phone: +49 621 776 - 0

E-mail: info@de.pepperl-fuchs.com

North American Headquarters

Pepperl+Fuchs Inc.

1600 Enterprise Parkway

Twinsburg, Ohio 44087

USA

Phone: +1 330 425-3555

E-mail: sales@us.pepperl-fuchs.com

Asia Headquarters

Pepperl+Fuchs Pte. Ltd.

P+F Building

18 Ayer Rajah Crescent

Singapore 139942

Phone: +65 6779-9091

E-mail: sales@sg.pepperl-fuchs.com

https://www.pepperl-fuchs.com

1 Introduction

1.1 Content of this Document

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

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

Note

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

The documentation comprises the following parts:

- This document
- Datasheet

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

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

1.2 Target Group, Personnel

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

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

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

1.3 Symbols Used

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

Warning Messages

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

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

Danger!

This symbol indicates an imminent danger.

Non-observance will result in personal injury or death.

Warning!

This symbol indicates a possible fault or danger.

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

Caution!

This symbol indicates a possible fault.

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

Informative Symbols

Note

This symbol brings important information to your attention.

Action

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

2 Product Description

2.1 Introduction

Vision sensors VOS2000-F226W-8MM-I, VOS2000-F226W-16MM-I, VOS2000-F226-C-I and VOS5000-F227-C-I (hereinafter referred to as sensors) are image-processing sensors for non-contact detection of 1-D and 2-D codes in industrial environments.

You must configure the sensor using the free Vision Configuration Tool software to use the sensor to detect an object. The software must be installed on a PC. The software allows you to adapt existing jobs or simulate them offline (i.e., without the sensor being connected).

During the configuration process, you create one or more test tasks called jobs. A job consists of:

- A programmed reference image for each object to be tested
- The vision tools that check for one or more features in the defined image areas of the object
- The function blocks assigned to the digital outputs, e.g., output 1 signals the result "Pass" and output 2 signals "Fail"

You can save up to 32 jobs on a single sensor. Several vision tools can be parameterized and executed simultaneously within a job. Existing interfaces can be used to switch between jobs during operation.

The measurement results from the camera can be preprocessed in the sensor using an integrated scripting language and outputted digitally to a data interface such as TCP/IP, PROFINET, or EtherNet/IP.

2.2 Versions

The sensor types are offered in two versions. Each version offers a different housing design. VOS2000 ("F226" housing design) as a compact camera with internal illumination or as a C-mount version.

The VOS5000 offers a higher resolution and uses the F227 housing design. This is a 5 megapixel camera with a C-mount lens and a connection for external lighting.

"F226" Housing Design

M12 lens

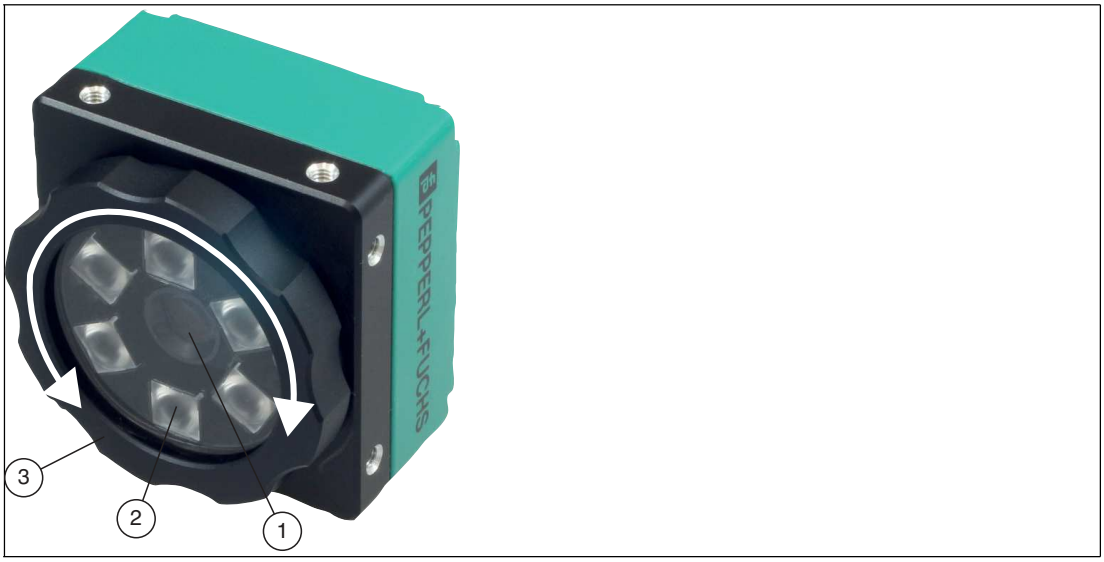


Figure 2.1 M12 lens (1), cover (3), and LED ring light (1)

The VOS2000-F226W-8MM-I sensor is equipped with an M12 lens and a focal length of 8 mm. The VOS2000-F226W-16MM-I is equipped with a M12 lens and has a focal length of 16 mm.

The M12 option comes with a special IP67 cover that allows you to adjust the focus by turning the lens cap without removing the cover.

The camera with the M12 lens also has an LED ring light, which is mounted inside the cover. The ring light is available in white.

C-mount lens

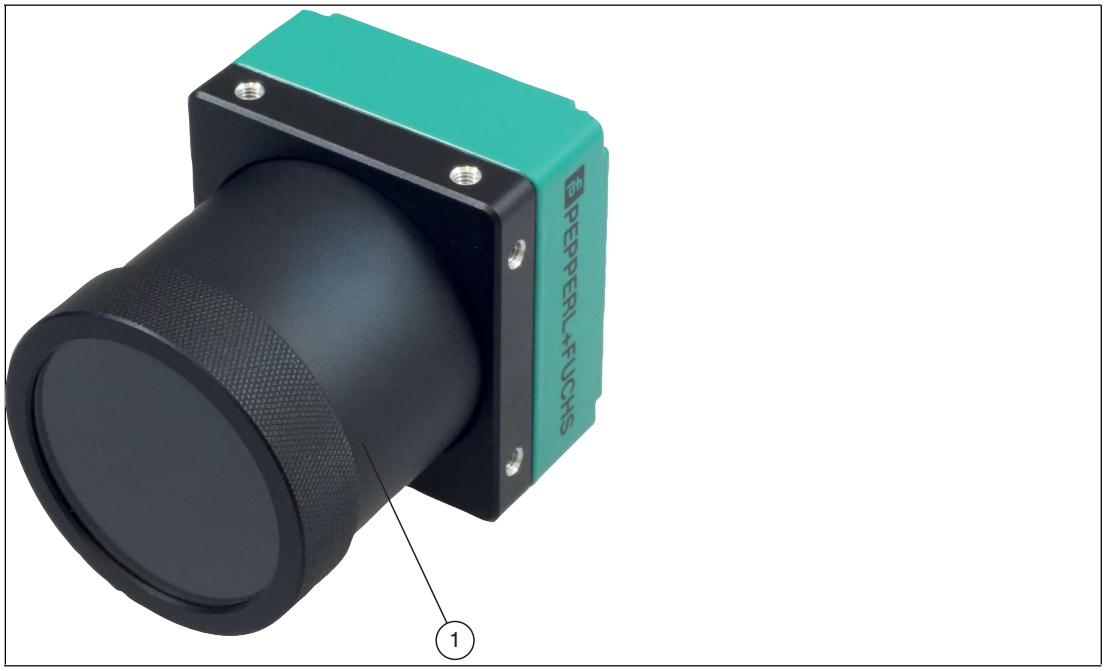


Figure 2.2 C-mount lens (1)

The VOS2000-F226-C-I sensor has a C-mount lens. The C-mount lens does not feature internal illumination. Optional external lighting can be used.

"F227" Housing Design

C-mount lens

Figure 2.3 C-mount lens plus lighting

The VOS5000-F227-C-I sensor has a C-mount lens. The C-mount lens does not feature internal illumination. Optional external lighting can be used.

3 Delivery, Transport and Storage

3.1 Unpacking

Check the product for damage while unpacking. If the product should be damaged, inform the post office or parcel service and notify the supplier.

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

Should you have any questions, please contact Pepperl+Fuchs.

3.2 Transport

The following guidelines must be observed and followed to ensure safe transport:

Note

Damage to the device due to improper transport!

- Do not remove packaging prior to assembly.
- Note the symbols on the packaging.
- Pack the device for transport such that it is protected from impact and humidity. Use the original packaging, which provides optimal protection.

3.3 Storage

When storing the device, pack it to protect it from shock and contamination. The original packaging provides optimal protection for the device. Store the device in a dry environment protected from humidity and potential harmful debris.

4 Mounting

4.1 Sensing Range

The size of the field of view depends on the operating distance, the lens used, and the optical resolution. The appropriate lens must be selected according to the measuring task. Note the sensing range of the respective lens when planning the system.

Note

Definition

- **Operating distance**: This is the distance between the optical surface of the sensor and the test object.
- **Field of view**: This is the image surface visible to a sensor at a given operating distance. If you need a larger field of view, you must increase the distance between the sensor and the object being tested. As the field of view increases in size, the optical resolution decreases. This affects the maximum achievable accuracy.

The figures below show fields of view at different operating distances for each lens focal length. The values in the table below serve as a reference point. The actual values may differ on account of the lens tolerances.

4.1.1 VOS2000 Sensing Range

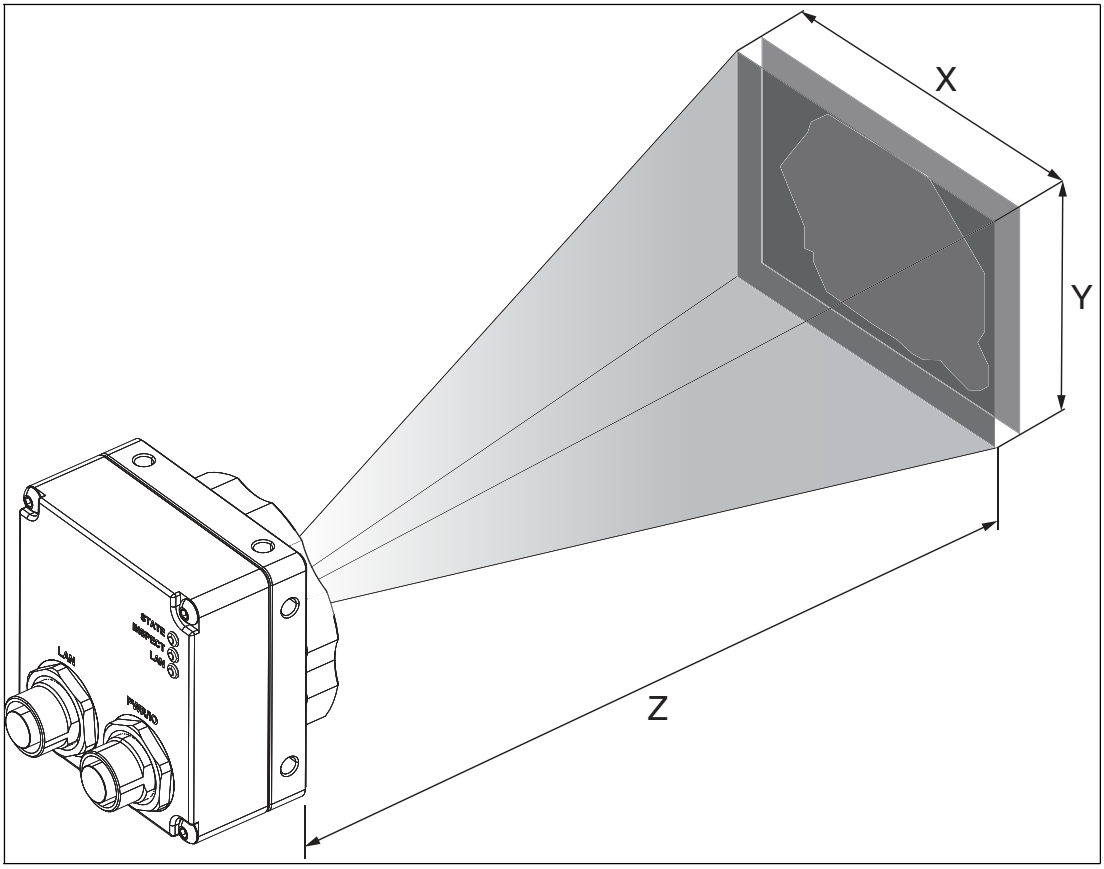
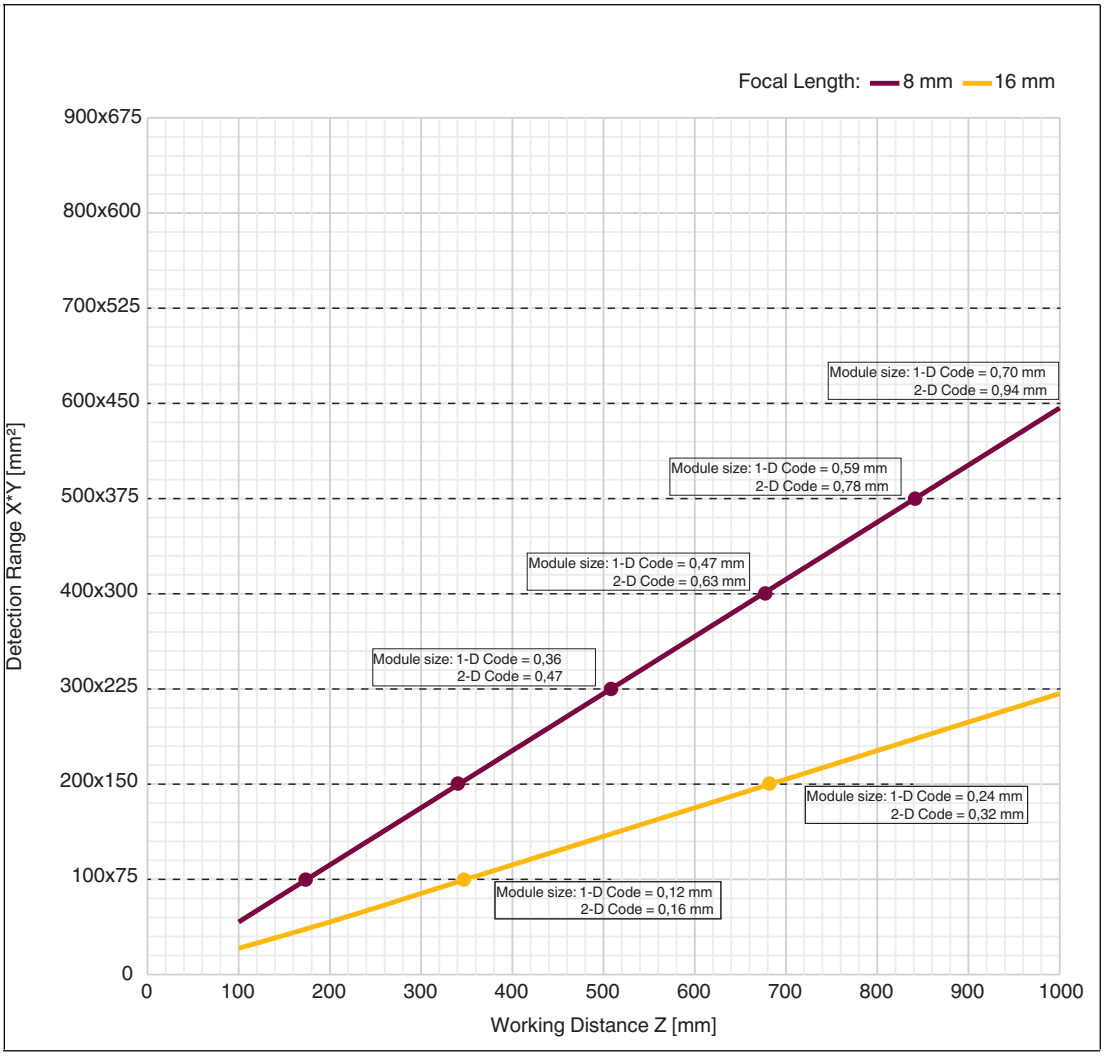


Figure 4.1 Sensing range (schematic diagram)

The following table shows the sensing range at different operating distances. The values in the table below serve as a reference point. The actual values may differ on account of the lens tolerances.

2021-07

VOS2000 Read Field Diagram



Operating distance Z [mm]	VOS2000 sensing range			
	8 mm lens		16 mm lens	
	X [mm]	Y [mm]	X [mm]	Y [mm]
100	55.2	41.4	33.3	24.98
200	115.2	86.4	55.2	41.4
300	175.2	131.4	85.2	63.9
400	235.2	176.4	115.2	86.4
500	295.2	221.4	145.2	108.9
600	355.2	266.4	175.2	131.4
700	415.2	311.4	205.2	153.9
800	475.2	356.4	235.2	176.4
900	535.2	401.4	265.2	198.9
1000	595.2	446.4	295.2	221.4

4.1.2 VOS2000 C-Mount Sensing Range

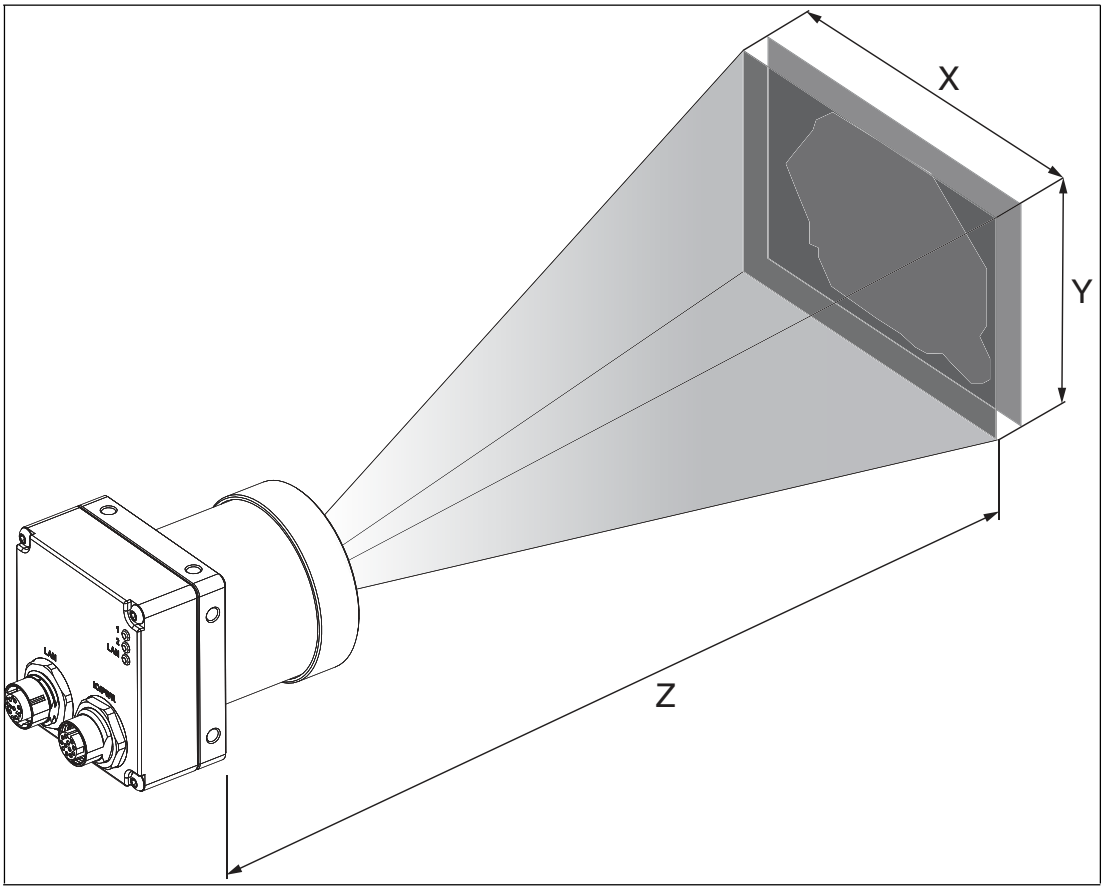
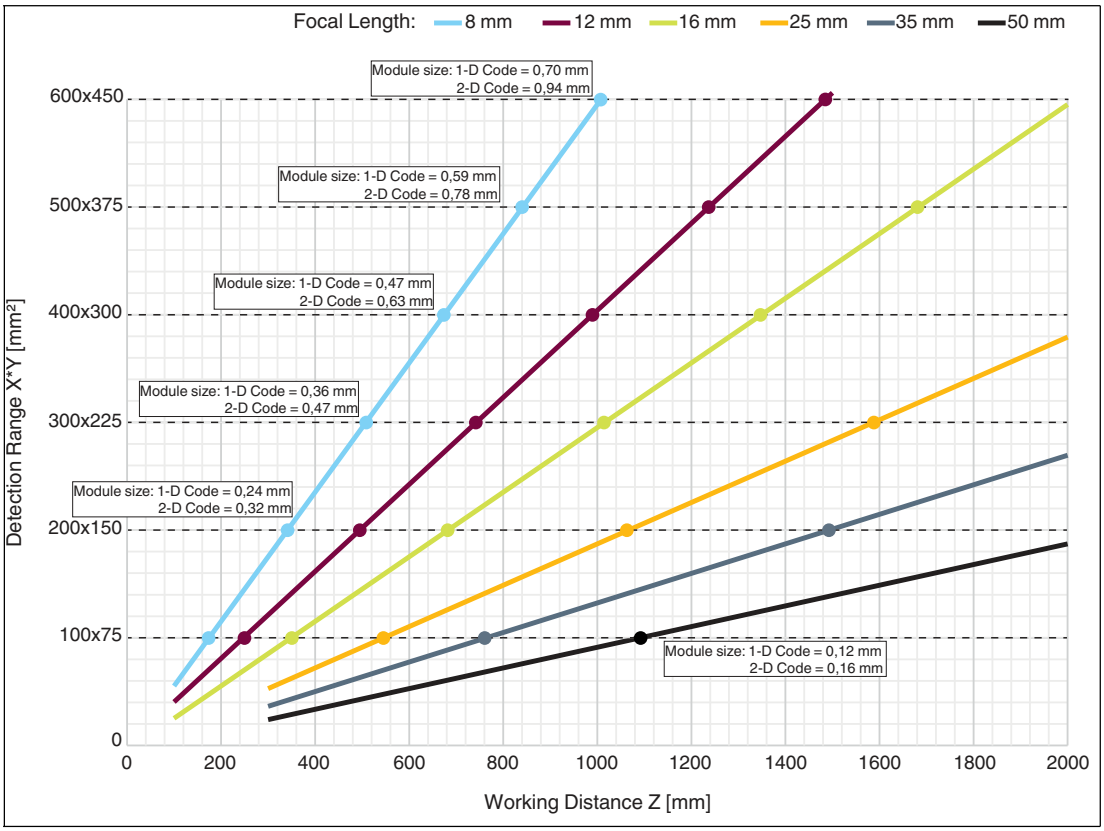


Figure 4.2 Sensing range (schematic diagram)

The following table shows the sensing range at different operating distances for each lens length. The values in the table below serve as a reference point. The actual values may differ on account of the lens tolerances.

VOS2000 C-Mount Read Field Diagram



Operating distance Z [mm]	VOS2000 C-Mount—Sensing Range					
	8 mm lens		12 mm lens		16 mm lens	
	X [mm]	Y [mm]	X [mm]	Y [mm]	X [mm]	Y [mm]
100	55.20	41.40	40.40	30.30	25.2	18.9
200	115.2	86.4	80.81	60.60	55.2	41.4
300	175.2	131.4	121.21	90.91	85.2	63.9
400	235.2	176.4	161.61	121.21	115.2	86.4
500	295.2	221.4	202.01	151.51	145.2	108.9
600	355.2	266.4	242.42	181.81	175.2	131.4
700	415.2	311.4	282.82	212.11	205.2	153.9
800	475.2	356.4	323.22	242.42	235.2	176.4
900	535.2	401.4	363.62	272.72	265.2	198.9
1000	595.2	446.4	404.03	303.02	295.2	221.4
1250	-	-	505.03	378.78	370.2	277.65
1500	-	-	606.04	454.53	445.2	333.9
1750	-	-	-	-	520.2	390.15
2000	-	-	-	-	595.2	446.4

Operating distance Z [mm]	VOS2000 C-Mount—Sensing Range					
	25 mm lens		35 mm lens		50 mm lens	
	X [mm]	Y [mm]	X [mm]	Y [mm]	X [mm]	Y [mm]
100	-	-	-	-	-	-
200	-	-	-	-	-	-
300	52.8	39.6	36.34	27.26	24	18
400	72	54	50.06	37.54	33.6	25.2
500	91.2	68.4	63.77	47.83	43.2	32.4
600	110.4	82.8	77.49	58.11	52.8	39.6
700	129.6	97.2	91.2	68.4	62.4	46.8
800	148.8	111.6	104.91	78.69	72	54
900	168	126	118.63	88.97	81.6	61.2
1000	187.2	140.4	132.34	99.26	91.2	68.4
1250	235.2	176.4	166.63	124.97	115.2	86.4
1500	283.2	212.4	200.91	150.69	139.2	104.4
1750	331.2	248.4	235.2	176.4	163.2	122.4
2000	379.2	284.4	269.49	202.11	187.2	140.4

2021-07

4.1.3 VOS5000 Sensing Range

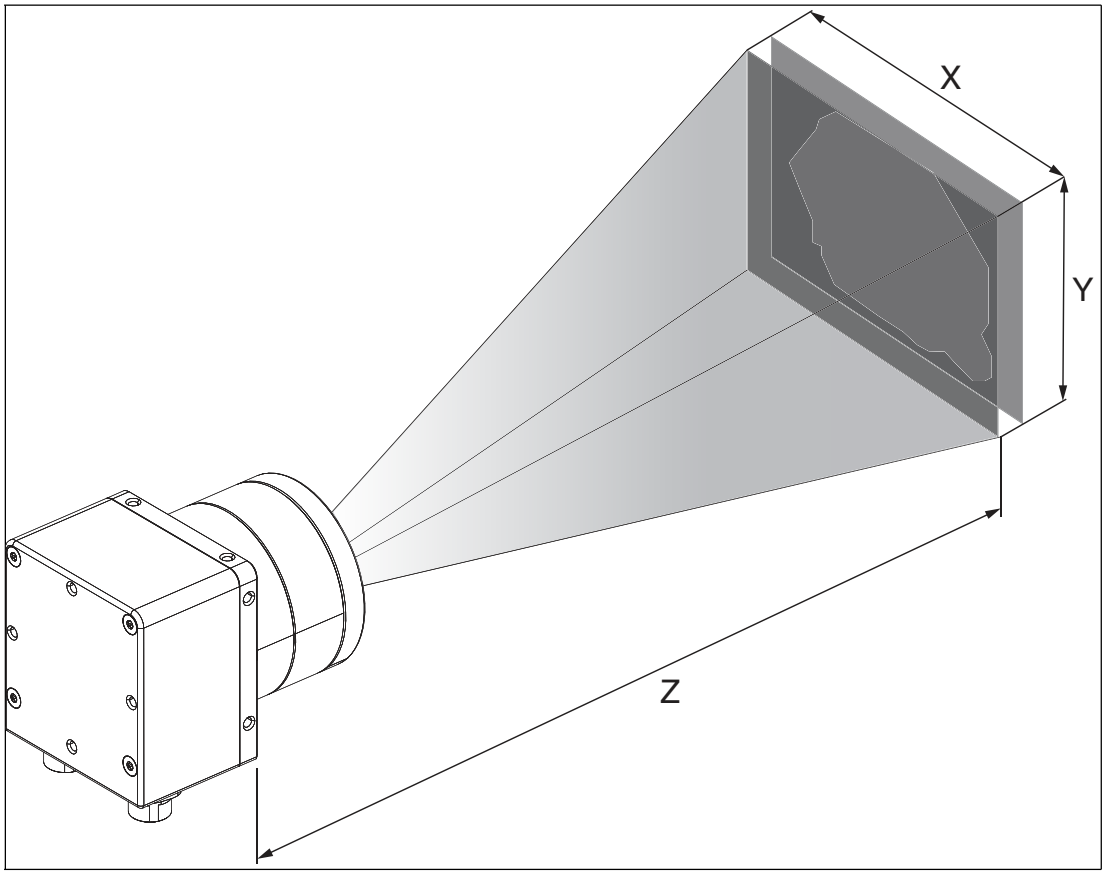
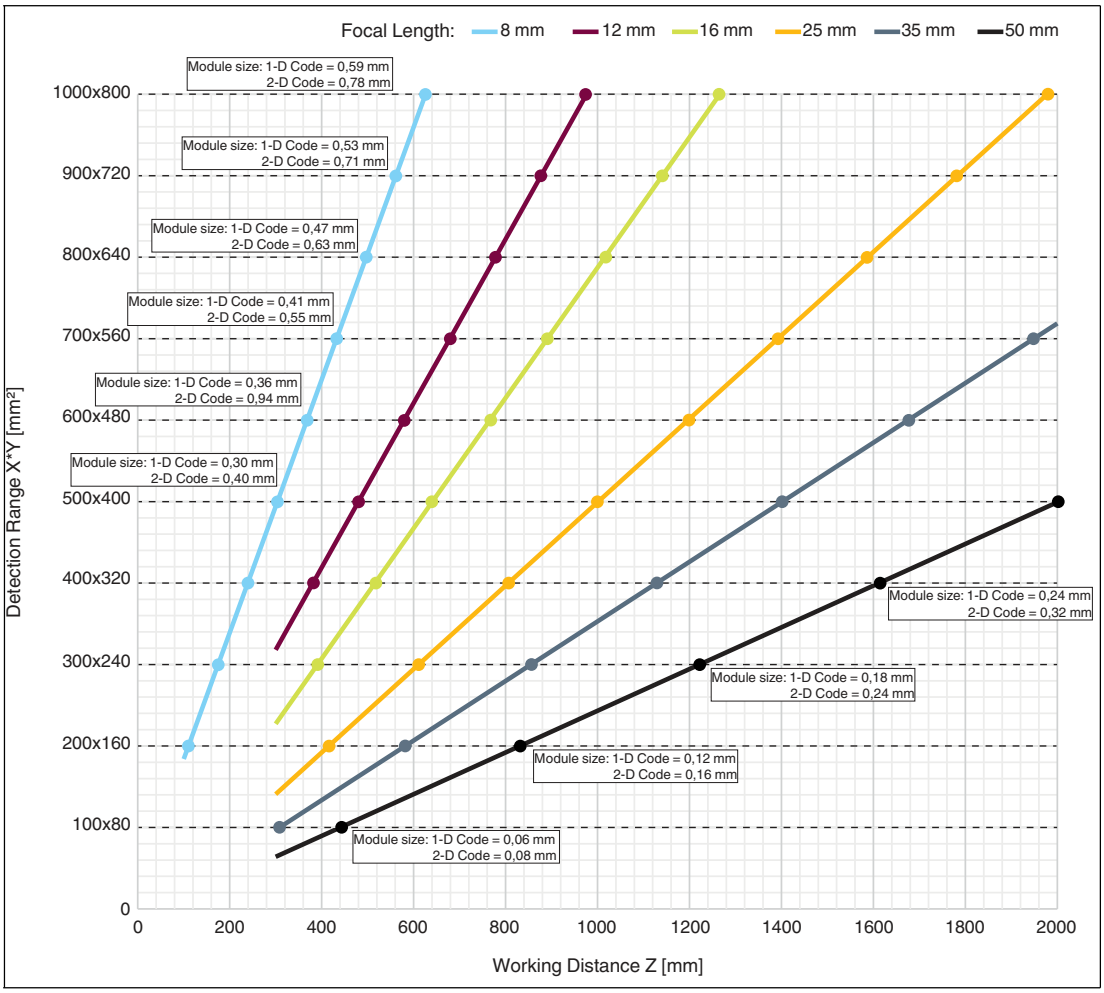


Figure 4.3 Sensing range (schematic diagram)

The following table shows the sensing range at different operating distances for each lens length. The values in the table below serve as a reference point. The actual values may differ on account of the lens tolerances.

VOS5000 Read Field Diagram



Operating distance Z [mm]	VOS500 sensing range					
	8 mm lens		12 mm lens		16 mm lens	
	X [mm]	Y [mm]	X [mm]	Y [mm]	X [mm]	Y [mm]
100	184	138	-	-	-	-
200	339	254	-	-	-	-
300	494	371	318	239	227.2	181.76
400	649	487	419	314	307.2	245.76
500	804	603	520	390	387.2	309.76
600	960	720	621	465	467.2	373.76
700	1115	836	721	541	547.2	437.76
800	1270	952	822	617	627.2	501.76
900	1425	1069	923	692	707.2	565.76
1000	1580	1185	1024	768	787.2	629.76
1250	-	-	-	-	987.2	789.76
1500	2356	1767	1528	1146	1187.2	949.76
1750	-	-	-	-	1387.2	1109.76
2000	3132	2349	2032	1524	1587.2	1269.76

2021-07

Operating distance Z [mm]	VOS500 sensing range					
	25 mm lens		35 mm lens		50 mm lens	
	X [mm]	Y [mm]	X [mm]	Y [mm]	X [mm]	Y [mm]
100	-	-	-	-	-	-
200	-	-	-	-	-	-
300	140.8	112.64	96.91	77.53	64	51.2
400	192	153.6	133.49	106.79	89.6	71.68
500	243.2	194.56	170.06	136.05	115.2	92.16
600	294.4	235.52	206.63	165.3	140.8	112.64
700	345.6	276.48	243.2	194.56	166.4	133.12
800	396.8	317.44	279.77	223.82	192	153.6
900	448	358.4	316.34	253.07	217.6	174.08
1000	499.2	399.36	352.91	282.33	243.2	194.56
1250	627.2	501.76	444.34	355.47	307.2	245.76
1500	755.2	604.16	535.77	428.62	371.2	296.96
1750	883.2	706.56	627.2	501.76	435.2	348.16
2000	1011.2	808.96	718.63	574.9	499.2	399.36

4.1.4 Module Size in Codes

When setting the operating distance between the sensor and the code, refer to the read field diagrams above. When setting the operating distance, the **"module size"** must be known, i.e., the width of the narrowest bar (for 1-D codes) or the side length of the square for Data Matrix codes (2-D codes). If the operating distance selected is too large, reading is either unreliable or not possible.

The values specified in the read field diagrams are maximum values for a specific module size. The operating distance must therefore be less than or equal to the specified distance for a certain module size.

1-D Code

Figure 4.4 Module size = minimum bar width

In a one-dimensional barcode, a module refers to the narrowest element (bar or gap) in a code.

2-D Code

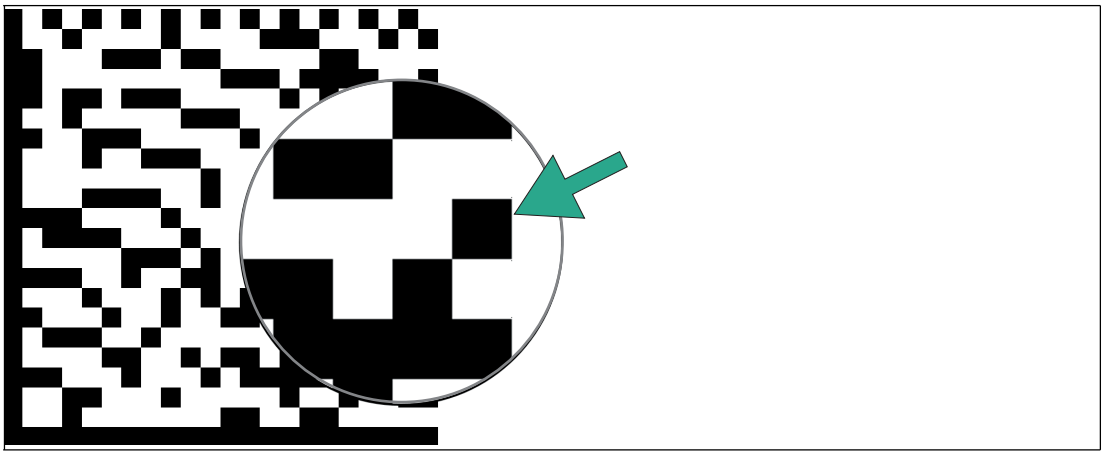


Figure 4.5 Module size = edge length of a square/bit

In Data Matrix codes, a module is a bit (black or white). The module size (also module width) is the edge length of a bit in millimeters or micrometers. Data Matrix codes are scalable, i.e., the module size can be adapted to the application.

4.2 Mounting the Sensor

Please note the following factors when mounting the sensor to ensure a reliable sensing mode:

- Sensing range between sensor and test object:
  - VOS2000: see chapter 4.1.1
  - VOS2000 C-Mount: see chapter 4.1.2
  - VOS5000: see chapter 4.1.3
- Module size in codes, see chapter 4.1.4.
- Environmental influences: If the sensor lens becomes dirty or scratched after setup, the sensor can no longer function reliably. For this reason, select the mounting location such that contamination of any kind is avoided as much as possible, or ensure regular cleaning. The specified ambient conditions on the datasheet must be observed.
- Prevention of ambient light such as sunlight and hall lighting.

Caution!

Damage to the equipment caused by improper mounting!

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

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

Never exceed the maximum permissible tightening speed of the fixing screws.

Note

The final mechanical orientation can only take place after the electrical connections have been made because the captured images can only be displayed on the PC using the configuration software.

Ensure that there is still sufficient space to connect the cable to the sensor once the sensor is mounted.

Securing and Aligning the Sensor

When positioning the sensor, ensure that the camera's field of vision is not obscured by other objects.

If you want to test an object with reflective spots, we recommend mounting the sensor at an angle of 10 to 20 degrees with respect to the vertical axis to prevent reflections.

VOS2000

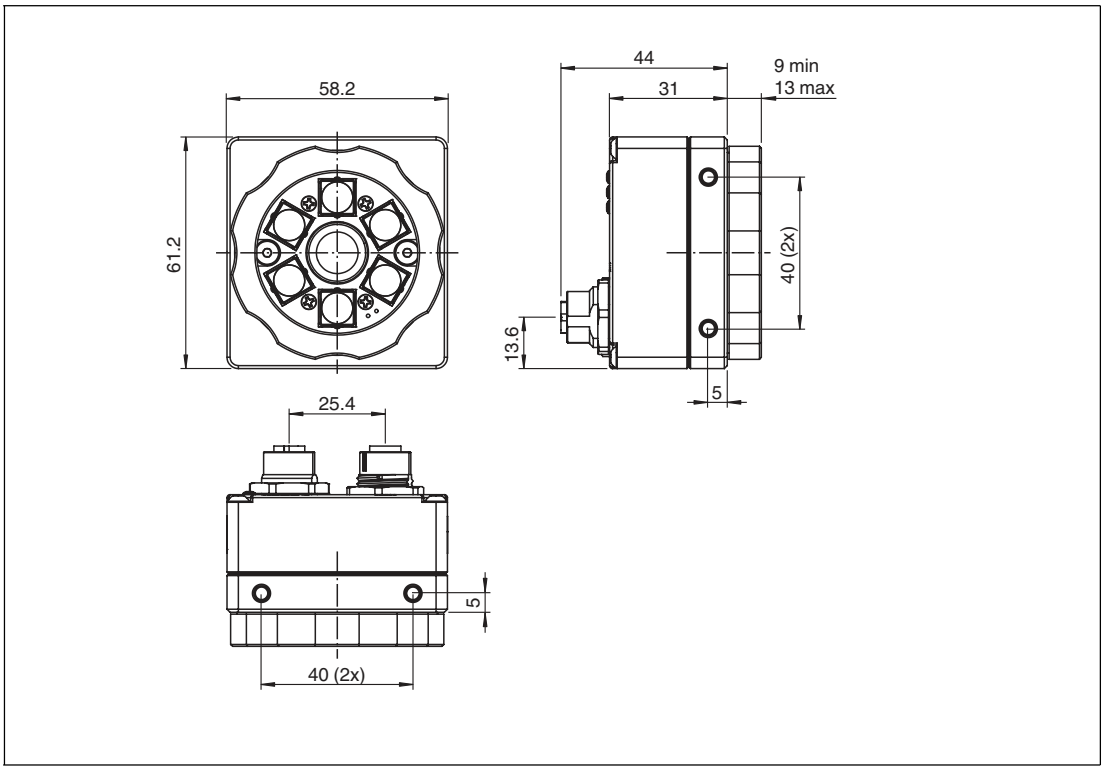
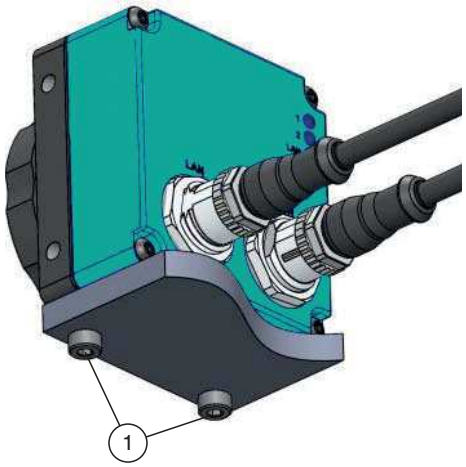
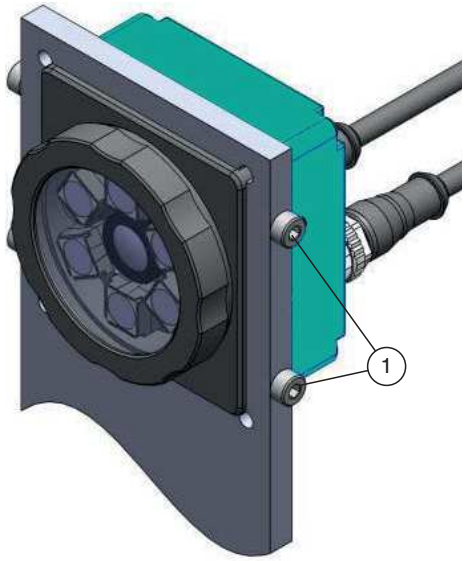


Figure 4.6 Dimensional drawing

Mounting Illustration	Information on Mounting
	Mounting on the underside of the device with M4 screws (1): You can use the two mounting holes with internal thread to mount the housing on the underside of the device. The maximum screw-in depth of the M4 screws is 7 mm.
	Side mounting with M4 screws (1): You can use the two mounting holes with internal thread on both sides of the sensor to mount the housing by its right side, its left side, or by both sides. The maximum screw-in depth of the M4 screws is 7 mm.

VOS5000

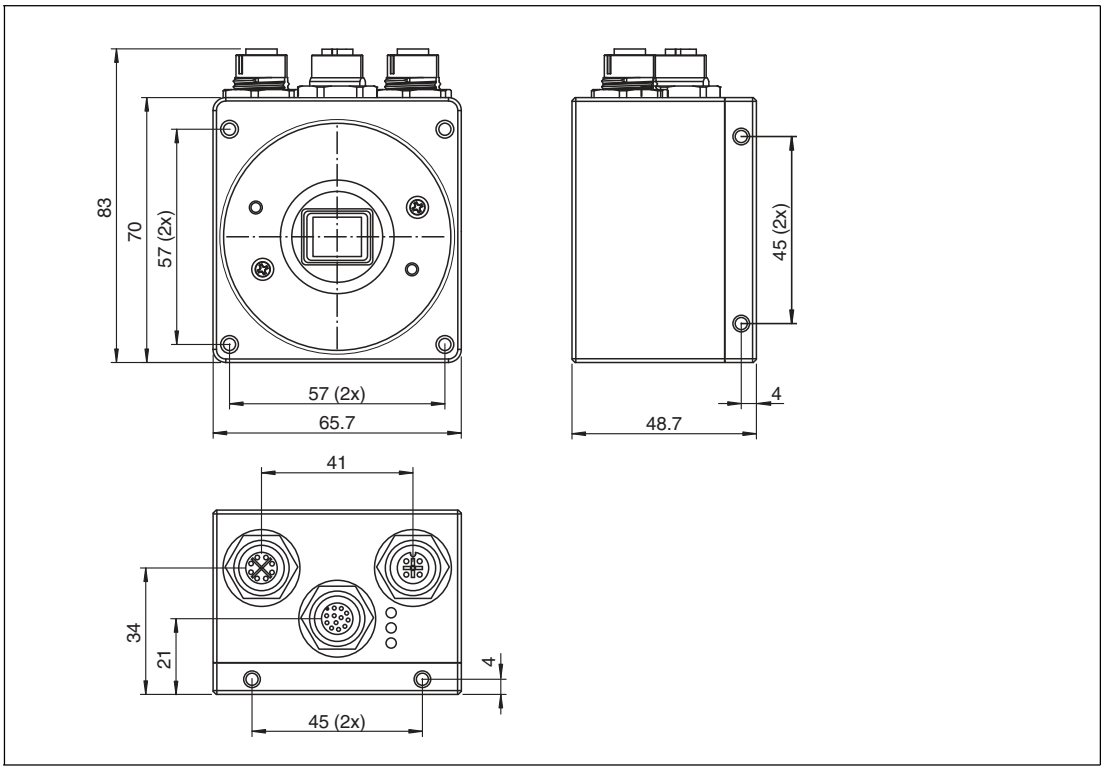
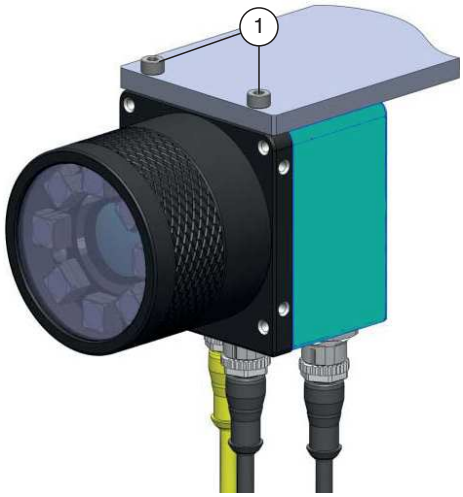
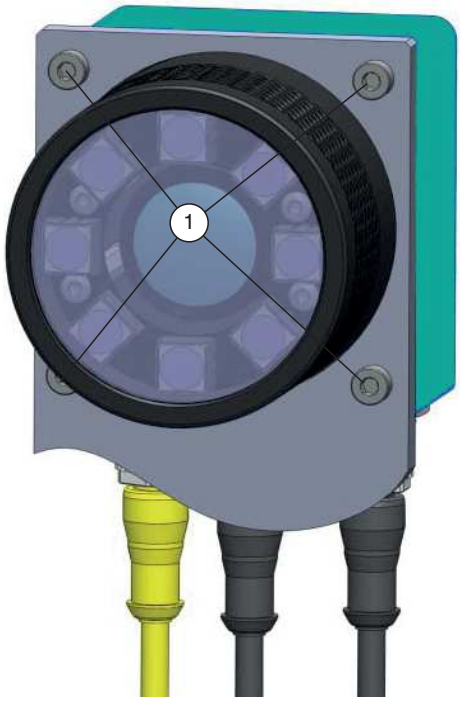


Figure 4.7 Dimensional drawing

Mounting Illustration	Information on Mounting
	Mounting on the top of the device with M4 screws (1): You can use the two mounting holes with internal thread to mount the housing on the top of the device. The maximum screw-in depth of the M4 screws is 7 mm.
	Mounting on the rear of the device with M4 screws (1): You can use the four mounting holes with internal thread to mount the housing on the rear of the device. The maximum screw-in depth of the M4 screws is 7 mm.
	Mounting on the front of the device with M4 screws (1): You can use the four mounting holes with internal thread to mount the housing on the front of the device. The maximum screw-in depth of the M4 screws is 7 mm.

Adjusting Image Sharpness / Focus Ring

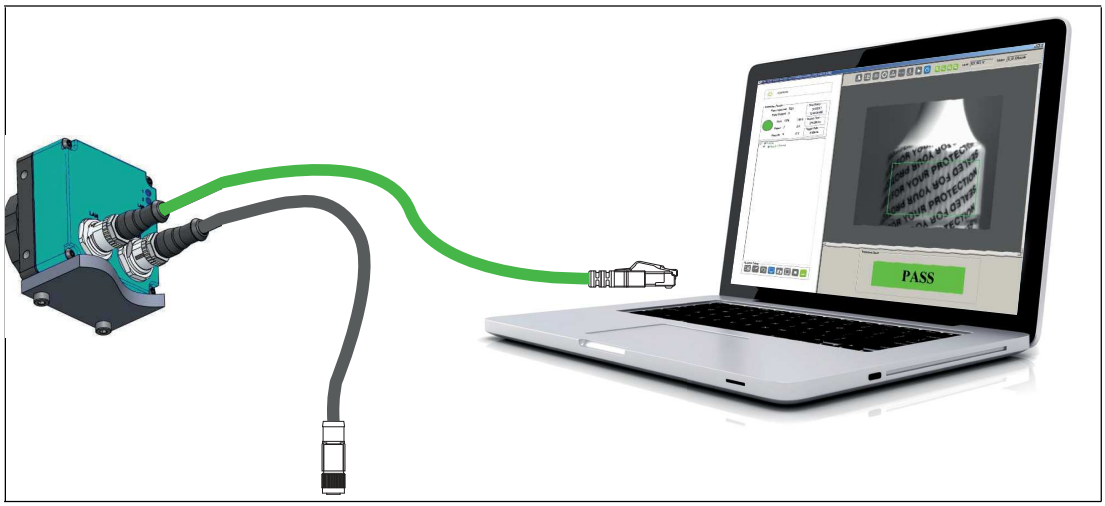
Turn the focus ring on the sensor to adjust the sharpness of the image. Turning the ring clockwise will focus objects that are further away. Turn the ring counterclockwise to bring closer objects into focus.

5 Installation

5.1 Connecting the Sensor

Connecting VOS2000

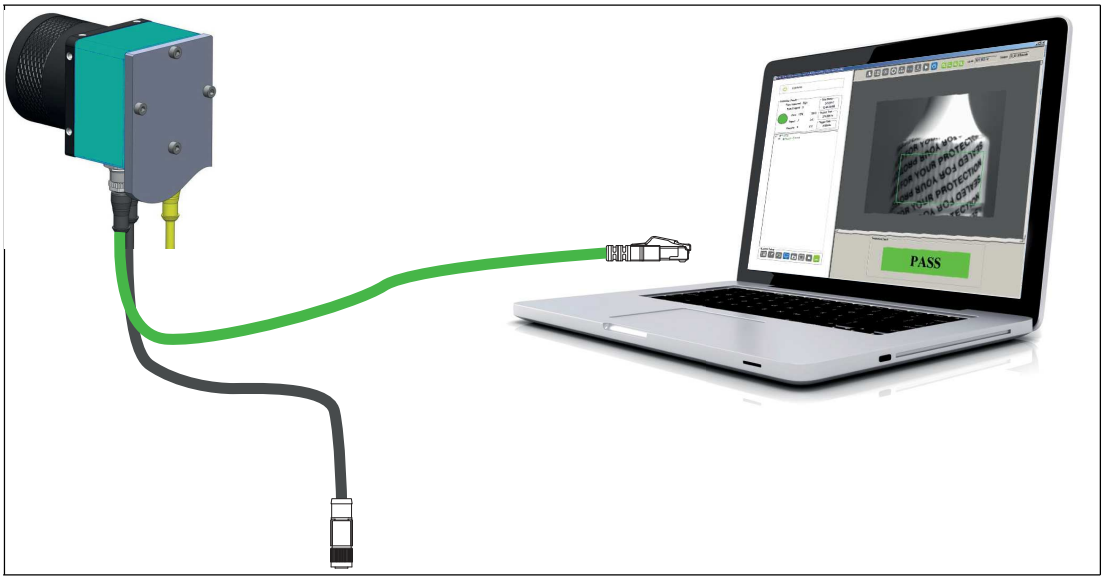
At the back of the sensor there are two connections: LAN and PWR/I/O.
To prepare the sensor for operation, proceed as follows:



1. Connect the Ethernet cable to the LAN connection on the sensor. Connect the other end of the Ethernet cable to the PC, PLC, or factory LAN (the PC is only required for setup).
2. Connect the connection module to the Power I/O connection on the sensor.
3. Connect the power supply in the off state to the connection module.

Connecting VOS5000

There are three connections on the base of the sensor: LAN, PWR/I/O, and LAMP.
To prepare the sensor for operation, proceed as follows:



1. Connect the Ethernet cable to the LAN connection on the sensor. Connect the other end of the Ethernet cable to the PC, PLC, or factory LAN (the PC is only required for setup).
2. Connect the connection module to the Power I/O connection on the sensor.
3. Connect the power supply in the off state to the connection module.

   For more information on the power source, see the following section.

5.2 Electrical Connection

5.2.1 VOS2000

There are two connections and three LEDs on the back of the sensor.

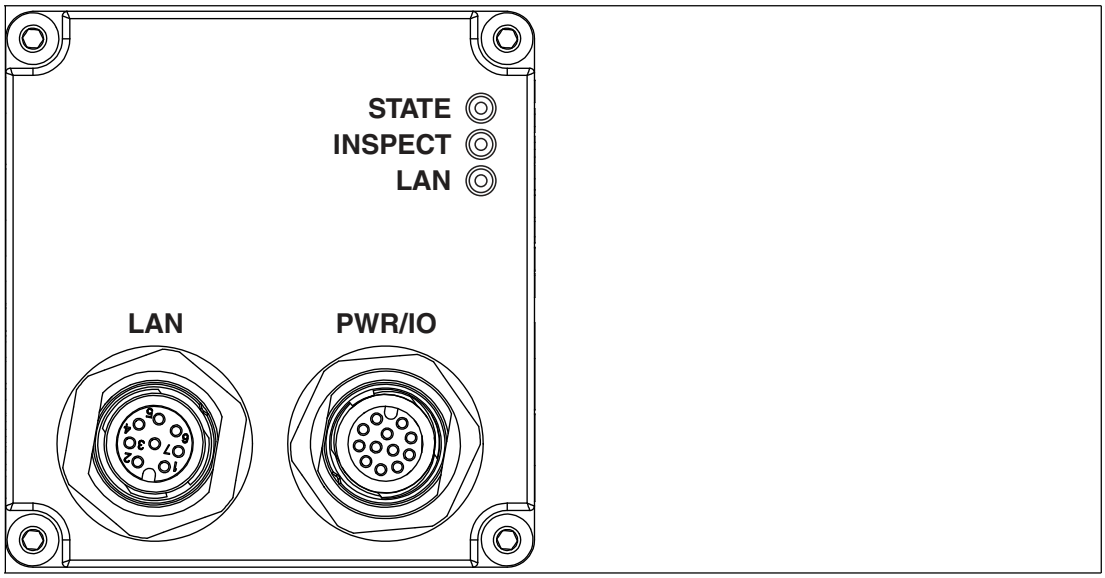


Figure 5.1 Back of the sensor with connections and LEDs

Sensor connections and LEDs

Connection/LED	Designation	Function
Connection	LAN	10/100 BaseT Ethernet connection. Provides the primary interface for configuring the sensor, developing the application, and monitoring the results.
	PWR/IO	Enables access to the sensor I/O: 3 IN, 3 OUT, RS-232. Also provides a PWR input (12 – 30 V). Note: The voltage supply to both connections is at the same potential. The voltage supply can be connected via the PWR/IO or LAN plug connection. Never connect both at the same time!
LED	STATE	• Lights up blue: sensor started, not configured (no job file) • Lights up green: job loaded, ready for execution • Flashes green: job loaded and executed, detection in progress • Lights up red: sensor fault
	INSPECT	• Flashes blue: starts (should stop after 20 seconds) • Lights up green: inspection confirmation • Lights up blue: test cycle (result of runtime decision) • Lights up red: inspection error (result of runtime decision)
	LAN	• Lights up blue: warm restart or restart • Red/green/yellow: network activity

Note

The sensor does not support the IEEE 802.3af Power over Ethernet (PoE) standard and should not be directly connected to a PoE-supported router.

LAN and PWR/IO Connection Assignment

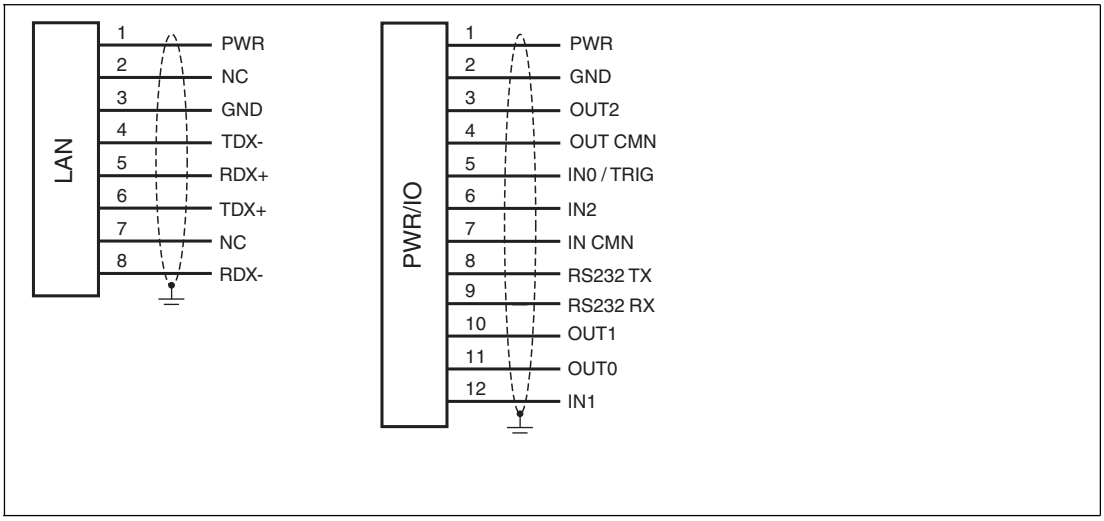
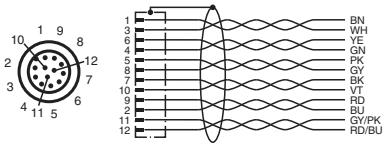
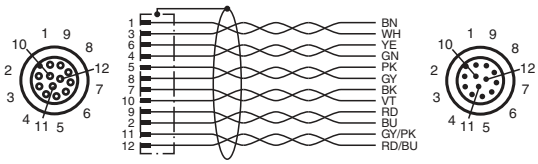
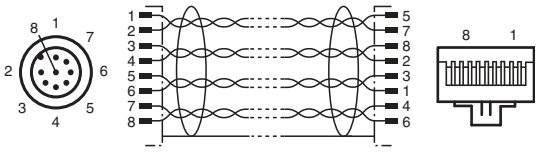


Figure 5.2 Power source diagram

Cable Overview

Cable	Description	Power Source
V112S-G-BK5M-PUR-ABG	Single-ended male cordset with M12 cable A-coded, 12-pin, PUR cable, black, shielded, open end	
V112-G-BK3M-PUR-ABG-V112-G	Cordset, M12 socket to M12 plug, A-coded, 12-pin, PUR cable, black, shielded	
V19S-G-GN5M-PUR-ABG-V45-G	Bus cable, Ethernet, M12 plug, A-coded, 8-pin to RJ45 Ethernet-coded, 8-pin, PUR cable, green, Cat6A, shielded	

5.2.2 VOS5000

There are three connections and three LEDs on the base of the sensor.

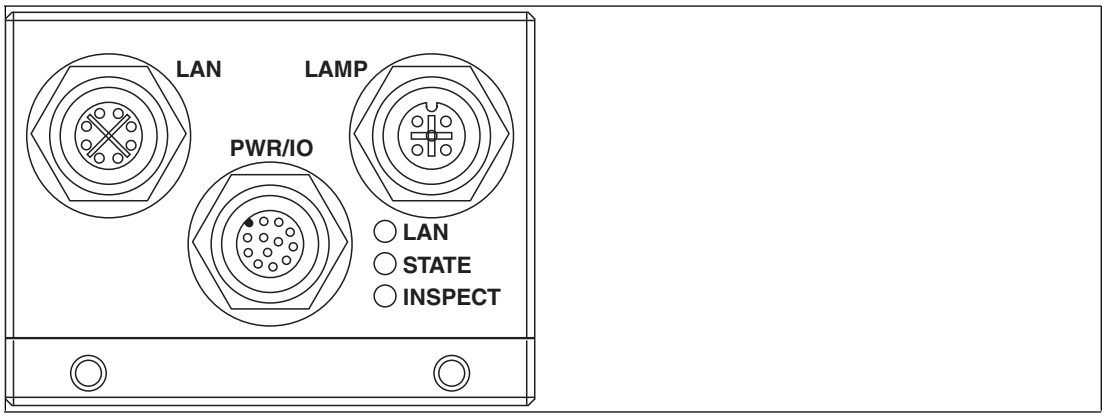


Figure 5.3 Base with connections and LEDs

Sensor connections and LEDs

Connection/LED	Designation	Function
Connection	LAN	10/100 BaseT Ethernet connection. Provides the primary interface for configuring the sensor, developing the application, and monitoring the results.
	PWR/IO	Enables access to the sensor I/O: 3 IN, 3 OUT, RS-232. Also provides a PWR input (12 – 30 V). Note: The voltage supply to both connections is at the same potential. The voltage supply can be connected via the PWR/IO or LAN plug connection. Never connect both at the same time!
	LAMP	Provides PWR and flash control for a local LED light source.
LED	LAN	• Lights up blue: warm restart • Red/green/yellow: network activity
	STATE	• Lights up blue: sensor started, not configured (no solution file) • Lights up green: solution loaded, ready for execution • Flashes green: solution loaded and executed, acquisition in progress • Lights up red: sensor fault
	INSPECT	• Flashes blue: starts (should stop after 20 seconds) • Lights up green: inspection confirmation (result of runtime decision) • Lights up blue: test cycle (result of runtime decision) • Lights up red: inspection error (result of runtime decision)

LAN and PWR/IO Connection Assignment

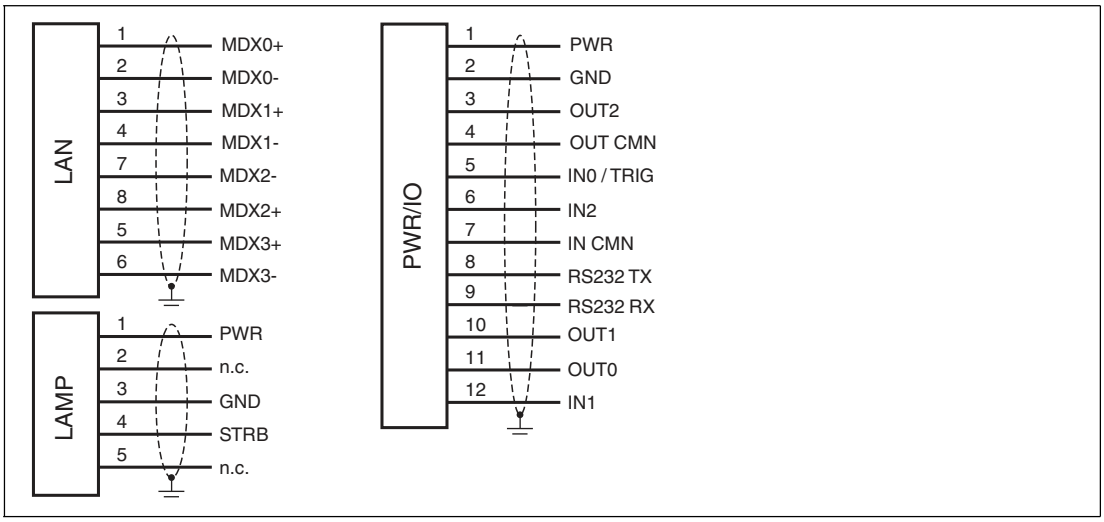
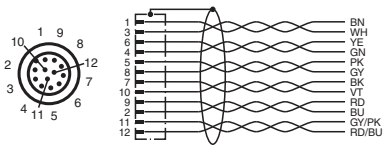
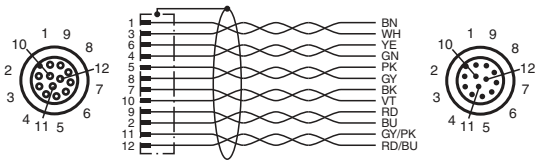
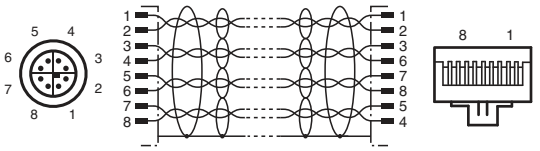


Figure 5.4 Power source diagram

Cable Overview

Cable	Description	Power Source
V112S-G-BK5M-PUR-ABG	Single-ended male cordset, M12, A-coded, 12-pin, PUR cable, black, shielded	
V112-G-BK3M-PUR-ABG-V112-G	Cordset, M12 socket to M12 plug, A-coded, 12-pin, PUR cable, black, shielded	
V19SX-G-GN5M-PUR-ABG-V45-G	Bus cable, Ethernet, M12 plug, X-coded, 8-pin to RJ45 Ethernet-coded, 8-pin, PUR cable, green, Cat6A, shielded	

6 Commissioning

6.1 Downloading and Installing Software

To set up the sensor, you need the Vision Configuration Tool application. The Vision Configuration Tool image processing application provides a range of image processing functions that meet various automated test requirements. There are two ways to download the application. Download directly from the Pepperl+Fuchs homepage or alternatively from the sensor supplied with the integrated Vision Configuration Tool application. The two ways to install the application are described below.

Installing the Vision Configuration Tool

This section describes how to obtain the installation file from the Pepperl+Fuchs homepage and then install it.

1. Go to the Pepperl+Fuchs homepage at http://www.pepperl-fuchs.com and enter the product name or item number in the search function. You can find the Vision Configuration Tool in the **Software** section of the product detail page for the device.
2. Save the installation file locally.
3. Start the exe file.

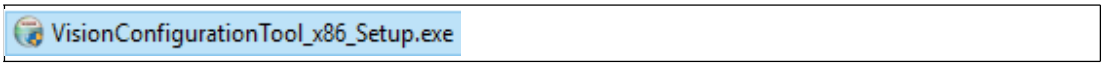


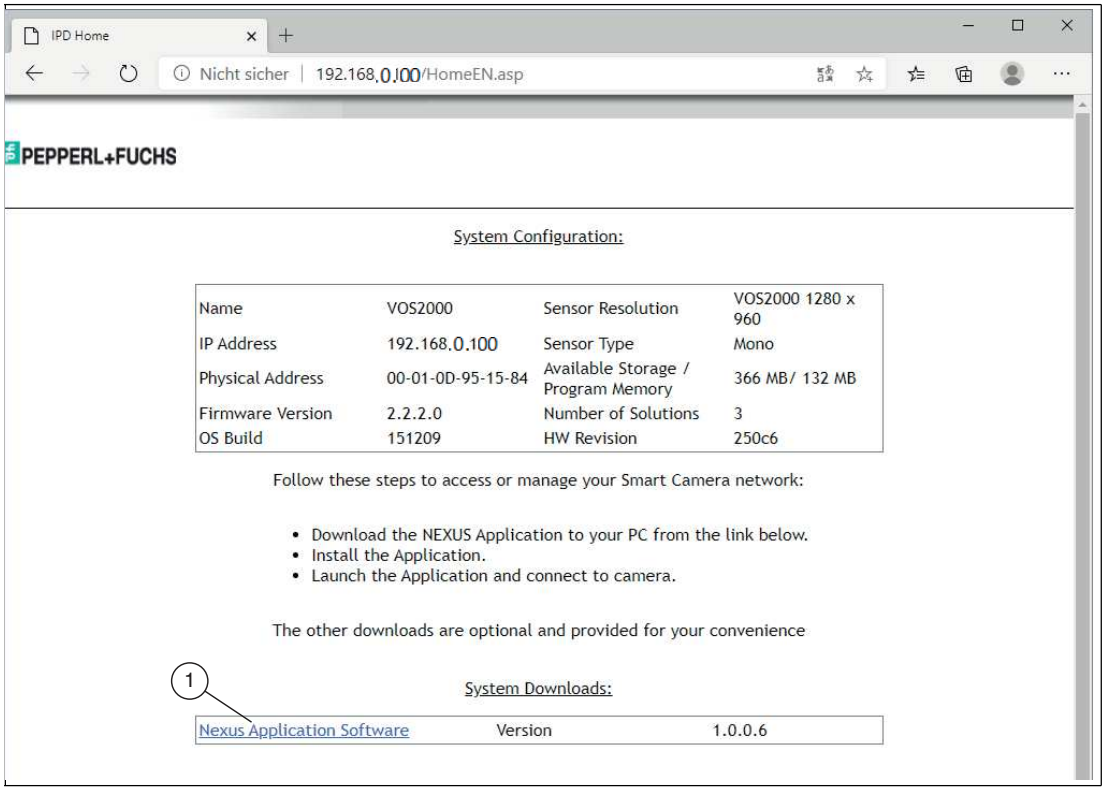
Figure 6.1 exe file

4. Follow the instructions for the installation process.

↳ After installation, several programs for monitoring and parameterizing are available in the Windows Start menu under **"Start > All programs > Pepperl+Fuchs Vision Configuration Tool".**

Downloading and Installing the Vision Configuration Tool from the Sensor

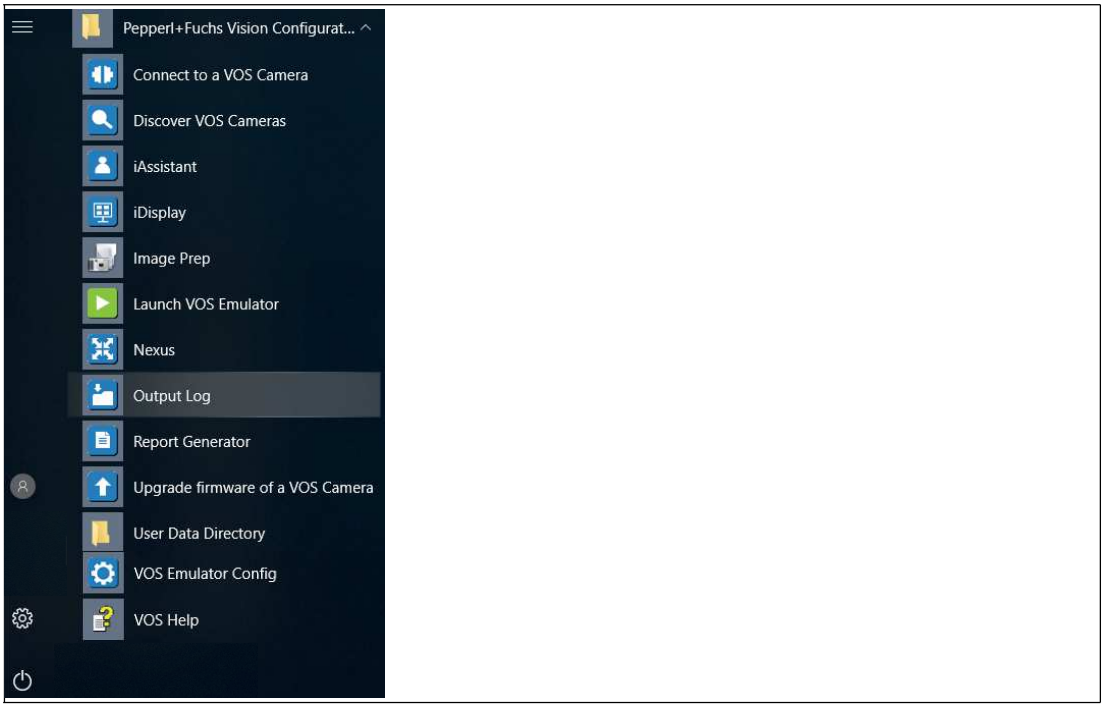
Alternatively, you can download the Vision Configuration Tool image processing application directly from the sensor. The sensor is supplied with the integrated Vision Configuration Tool parameterization software. The application interface is accessed via the Ethernet connection to a PC running Microsoft® Internet Explorer 6 or higher.



1. Open a web browser and type the sensor address 192.168.0.100 in the address bar and press Enter to confirm.

   ↳ The homepage of the sensor web browser is displayed in the browser.
2. Click on the link for "Nexus Application Software" (1) and save the file locally.
3. After the download is complete, you can close the web browser.
4. Open the Downloads folder on your PC.
5. Right-click the exe file and select "Run as administrator".
6. After installation, several programs are available in the Windows Start menu under "**Start > All programs > Pepperl+Fuchs Vision Configuration Tool**".

Programs



Program	Description
Connect to a VOS Camera	Clicking "Connect" opens the main Vision Configuration Tool (see chapter 7) user interface for the selected sensor. This user interface allows you to develop, load, store, or monitor applications (inspections).
Discover VOS Cameras	iDiscover is used to set the IP address of the sensor or to change the device name. You can use this link to launch Discover VOS Cameras or you can use the Nexus application to change the address of a sensor. All sensors are supplied with the same address and name. You must change the address if there is more than one sensor in the network. Changing the name is optional but is recommended to avoid confusion.
iAssistant	iAssistant is used to clean (delete) outdated files on the PC. Nexus can manage multiple versions of the sensor firmware. If you encounter problems, cleaning may be necessary. You can use this link to launch iAssistant or you can use the Nexus application to launch iAssistant.
iDisplay	The iDisplay application supports the display of multiple sensors. You can view multiple active sensors in the Nexus user interface.
Image Prep	Convert JPG to BMP.
Launch VOS Emulator	Launches the emulator
Nexus	Launches Nexus to access the sensor and to manage the sensor. See chapter 6.3.

2021-07

Program	Description
Output Log	This application provides the option to save results from connected sensors to CSV files on the PC.
Report Generator	Generating a solution report
Upgrade firmware of a VOS Camera	This program provides the option to update sensors to a different firmware version.
User Data Directory	Opening the user directory
VOS Emulator Config	Launches the configuration window for the emulator. In the VOS emulator, you can test or demonstrate applications. For this, you need a test photo that you have taken with the sensor. The resolution of the test photo should match the resolution of the selected sensor. You can upload and test the test photo in the emulator. See chapter 6.2.
VOS Help	VOS Online Help

6.2 VOS Emulator

The emulator is a user interface that has the same structure as the standard user interface. The emulator does not require a connected sensor but simulates one instead. This allows you to test the sensor functions on original images and therefore find the optimum settings for the vision tools without having to connect the respective sensor.

Setting the Emulator

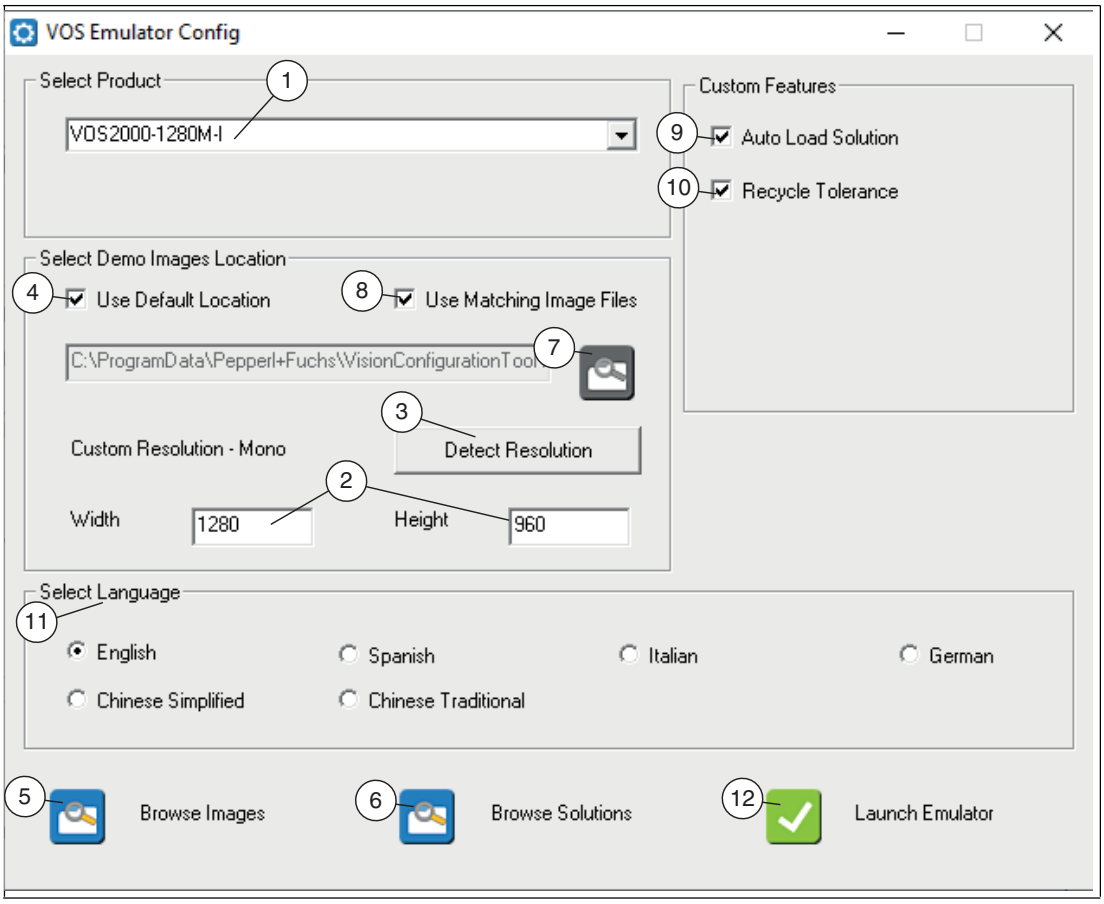


Figure 6.2 Emulator menu

1. Start the Emulator Configuration program from the Windows Start menu under **"Start > All Programs > Pepperl+Fuchs Vision Configuration Tool > VOS Emulator Config."**
2. Select your sensor from the "Select Product" drop-down list (1).

   ↳ The resolution (width and height in pixels) (2) of the selected sensor is displayed under "Custom Resolution." You can determine the resolution of the sensor using "Detect Resolution" (3).
3. Select the image source. The emulator application provides some test images. These are available if the "Use Default Location" (4) check box is checked. The application data directories can be accessed via "Browse Images" (5) or "Browse Solutions" (6). You can use other images by clearing this "Use Default Location" check box (4) and switching to another directory (7).

2021-07

Note

Images must be monochrome bitmaps with the resolution of the selected sensor.

You can use images with a different resolution by clearing the "Use Matching Image Files" check box (8). Use this option with care, since jobs that were created without a matching image resolution are not transferred to the sensor.

4. Set the user-defined functions.

Note

- Auto Load Solution (9): If you check this check box, a default job is loaded at startup and the application test is launched.
- Recycle Tolerance (10): If you select this check box, the test will output the states "Pass", "Recycle", or "Fail", depending on the tolerances set.

5. Select a language under "Select Language" (11).
6. Click "Launch Emulator" (12).

↳ The application test is opened in the Vision Configuration Tool.

6.3 Nexus

Nexus is the main application used to connect, set up, and manage the sensors. The application can be downloaded and installed from the Pepperl+Fuchs homepage or installed with the emulator.

The following functions are supported by Nexus:

- Finding sensors in the network
- Configuring network parameters
- Updating firmware
- Using a mix of sensors and firmware versions
- Connecting sensors for setup, processing, and runtime monitoring

Overview of the User Interface

The user interface of the Nexus application is shown below. When you first launch the application, it automatically detects connected sensors in the network (left image) and shows what it finds in the camera window (right image).

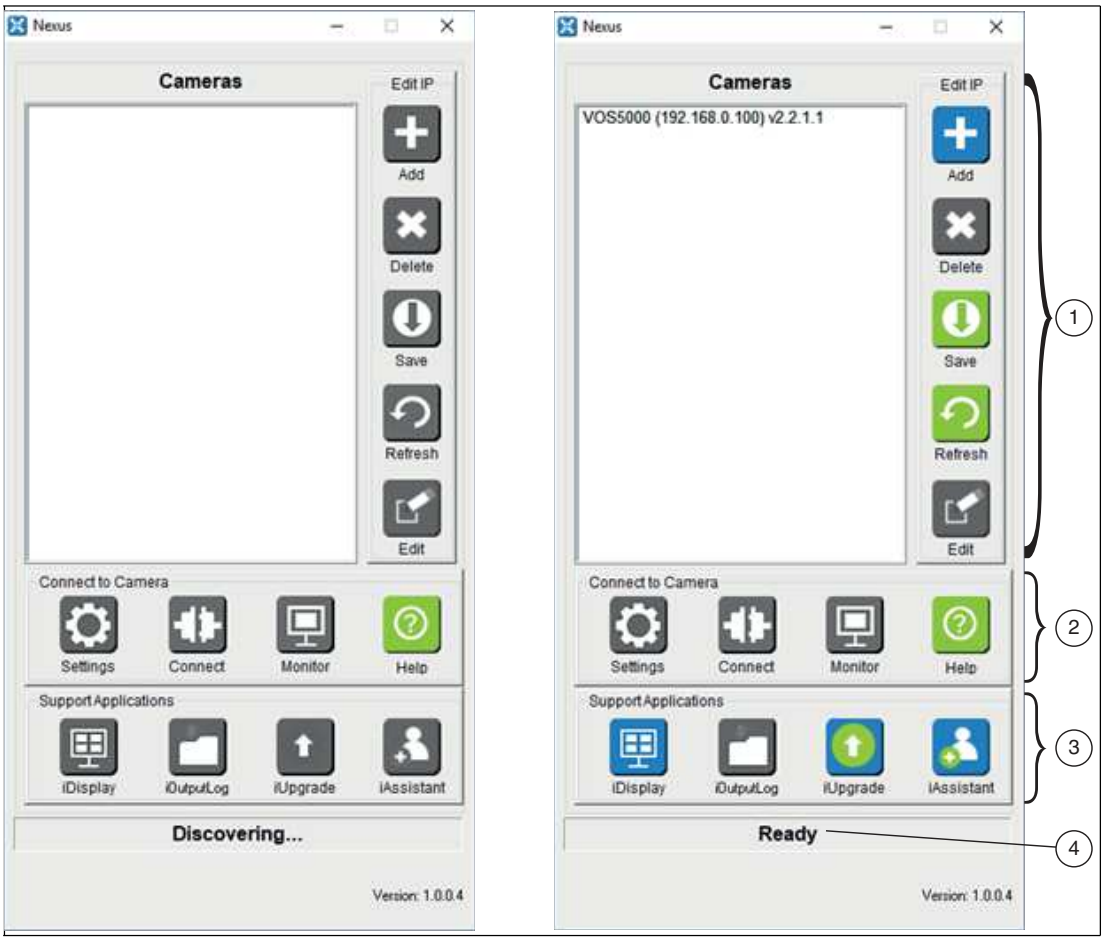


Figure 6.3 Nexus user interface

1 IP settings, configuration controls, and detection controls

2 Sensor settings and connection controls

3 Multisensor display and support applications

4 Activity status

User interface action buttons are gray during sensor detection, and some remain gray until a sensor is selected. Click on a sensor to activate its control buttons. Clicking a green or red button has an immediate effect; clicking a blue button requires further action.

2021-07

Starting the Nexus Application

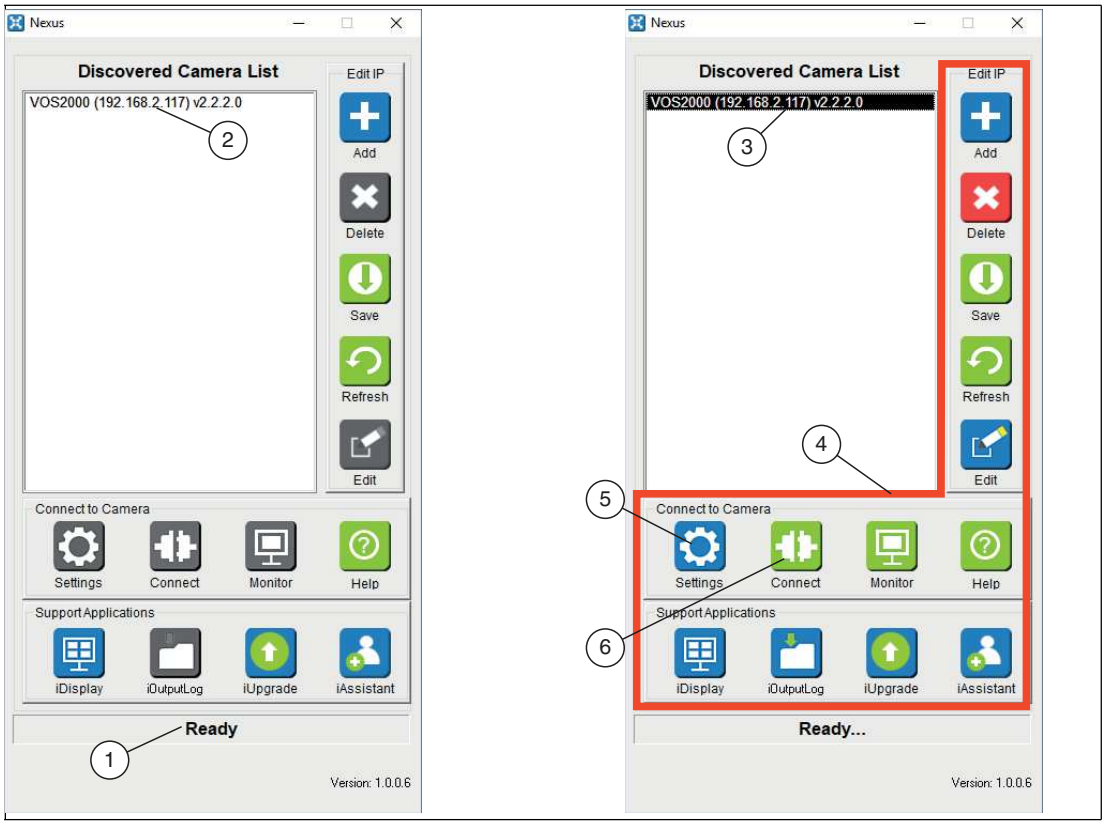


Figure 6.4 Nexus user interface

1. Open **Nexus** on your PC.

   ↳ Nexus scans the network for connected sensors. The status field (1) shows the progress. As soon as the status changes to "Ready", all connected sensors are displayed on the "Discovered Camera List" (2).

   Note

   If an incompatible sensor is detected, all action buttons remain gray until the associated IP settings have been corrected, see "Changing the IP address" on page 35.

2. Click a sensor on the "Discovered Camera List" (3).

   ↳ All buttons (4) on the user interface become active.

3. Click "Settings" (5) to set the camera settings.

4. Click "Connect" (6) to open the **"VOS Configuration Tool"** application and begin job programming.

Overview of IP Settings

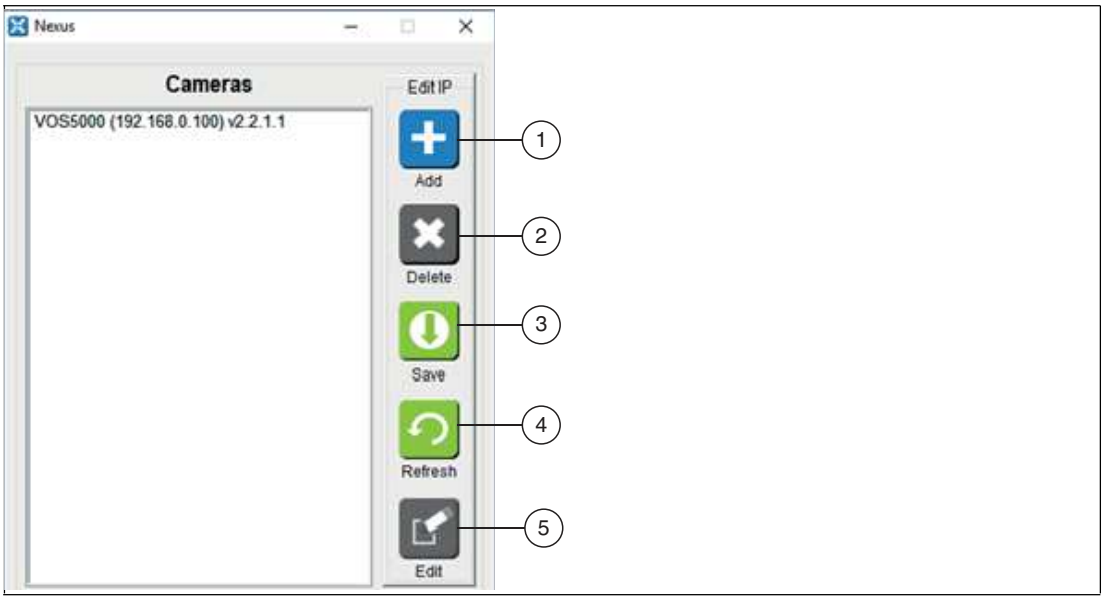


Figure 6.5 Edit IP

Position	Button	Description
1	Add	Manually add sensors that were not found before the automatic search.
2	Delete	Delete the selected sensor. The deleted sensor is not displayed when the configuration is saved.
3	Save	Save the sensor configuration list as shown.
4	Refresh	Perform a manual search.
5	Edit	Edit IP settings to correct network incompatibilities.

Note

Note regarding item 4

The camera window (Cameras) is updated when you manually perform a new search. Your saved configuration file is retained. The next time you start Nexus, it will only show the sensors you want to see.

If you want to delete the saved configuration file, click "Save", and the dialog box will offer an option to delete it. Similarly, the dialog box will allow you to overwrite the saved file.

When you update a sensor in your saved configuration list, you must manually start a new search to update the list. A new configuration file is then saved. If you do not do this, you will have trouble connecting to the sensors if the update version files are not stored on the system.

Changing the IP address

When Nexus is launched, it scans the network connections of the system in search of compatible sensors. If sensors with an incompatible IP address are found, they are highlighted in red (1). Sensors hidden behind a router might not be detected, but they can be added manually if you know their IP addresses. To set the IP address, proceed as follows:

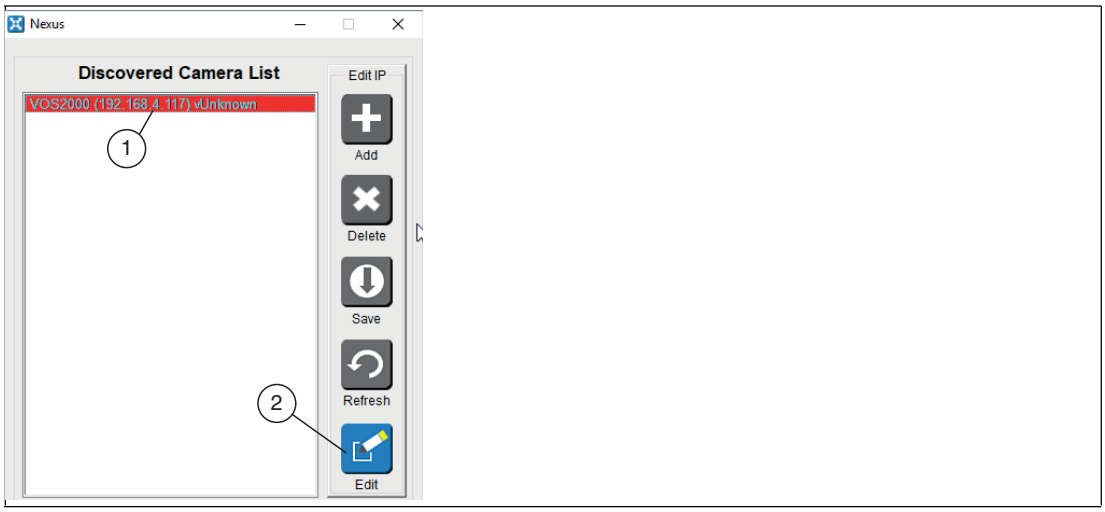


Figure 6.6 Unknown IP address

1. Select a sensor from the "Discovered Camera List" (1) and click "Edit" (2).

Note

If an incompatible sensor is detected, all action buttons remain gray until the associated IP settings have been corrected. When you close the editing window, a search for sensors is started.

↳ The window for the "Resolve Camera Settings" IP settings opens.

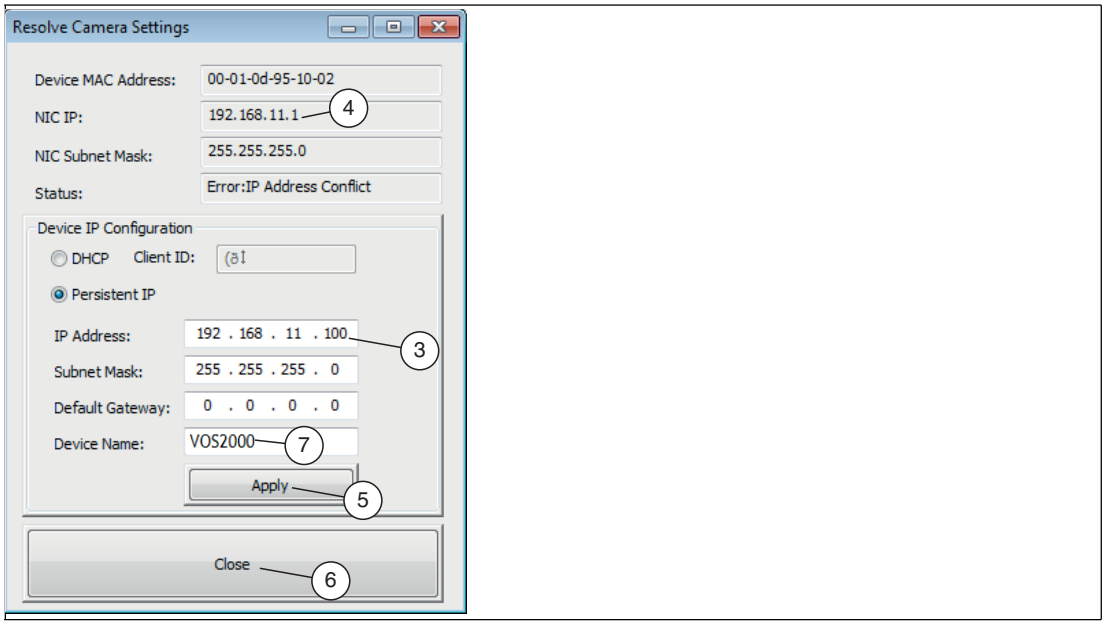


Figure 6.7 Changing the IP address

2. Change the IP address (3) of the sensor to use a free IP on the subnet of your network card (NIC IP) (4). Click "Apply" (5) and "Close" (6) to save the changes.

Note

You can change the standard device name (7) in this settings window.

↳ Nexus scans your network for connected sensors.

Note

Once the sensor is detected, you can configure which sensor should appear in your network when Nexus is launched. This is useful if you have manually added sensors via a router, or if many sensors in your network are directly or indirectly visible to multiple users. Saving a custom configuration prevents the automatic search when Nexus is launched.

Overview of Application Settings

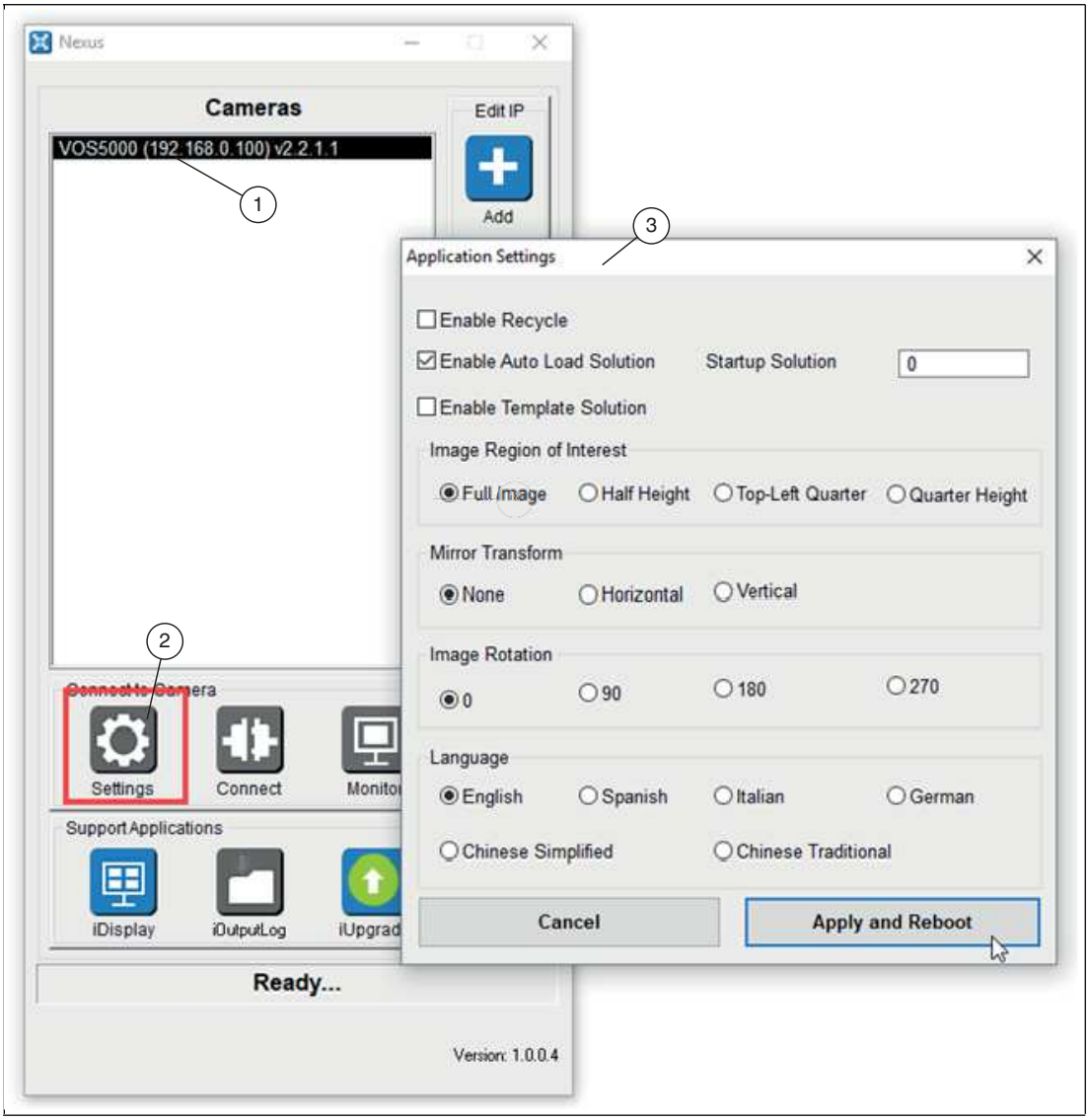


Figure 6.8 Changing the Application Settings

1 List of connected sensors (Cameras)

2 "Settings" Menu

3 "Application Settings" window

Changing Application Settings

1. In the Nexus application window, click your sensor in the "Cameras" list and then click "Settings."

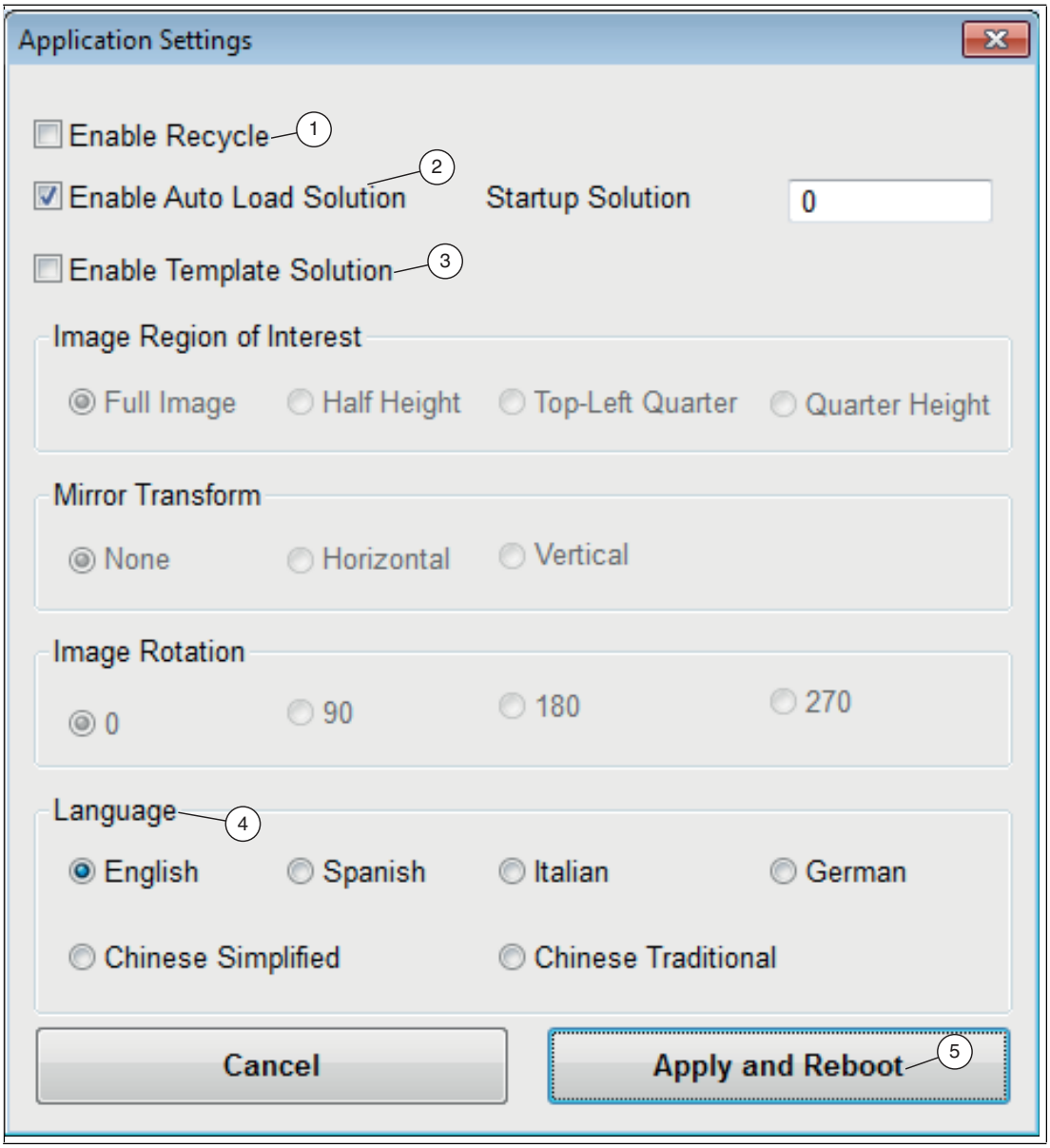


Figure 6.9 Application Settings

↳ The "Application Settings" window opens.

Note

1. You can enable or disable recycle tolerances for the Vision Configuration Tool using the "Enable Recycle" (1) selection check box. The "perfect" and "pass/fail" tolerances remain enabled.
2. You can use the "Enable Auto Load Solution" (2) selection check box to enable a previously programmed job file (Solution) when the sensor is switched on or restarted.
3. You can use the "Enable Template Solution" (3) check box to load a previously saved job file template. The job file acts as a template for your new measurement. All settings of the saved job file (variables, scripts, and tools) are the starting point for the next measurement process.
4. You can change the display language in the Vision Configuration Tool application software.

5. When you have completed your settings, click "Apply and Reboot" (5). The sensor is restarted.

Establishing a sensor connection

To connect to a sensor for setup or monitoring, proceed as follows:

Select your sensor from the "Cameras" list (1) and click either "Connect" (2) or "Monitor" (3).

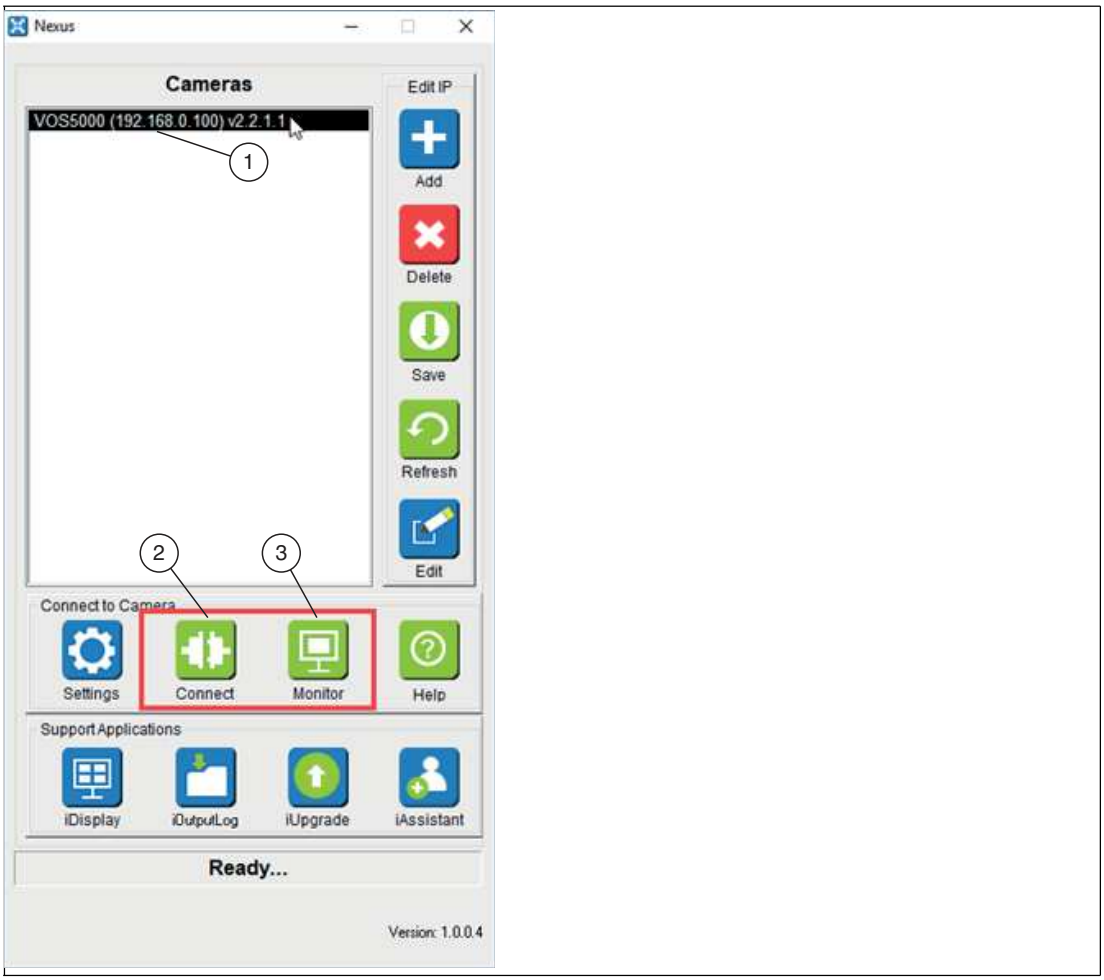


Figure 6.10 Connecting the sensor (Connect to Camera)

↳ The user interface of the selected sensor is started. If there is no application running on the sensor, the monitoring application will not start. When you use multiple sensors, the user interface associated with the selected sensor type is started.

Note

- Clicking "Connect" opens the main Vision Configuration Tool (see chapter 7) user interface for the selected sensor. This user interface allows you to edit, load, save, and monitor your job.
- We recommend that you do not run the main user interface during operation if your application has demanding processing or speed requirements.
- You can open connections to multiple sensors by selecting each sensor from the "Cameras" list and clicking "Connect" or "Monitor" one at a time. Note that this will increase the load on the connection.
- The iDisplay application provides a basic user interface that displays images and results. Its setup is identical to the main user interface except that most control buttons are disabled. You can only change the update rate of the display and choose between text or graphical inspection status.

Multisensor display and support applications

Nexus provides the main application for real-time viewing, logging of inspection results, and updating of sensor firmware.

iDisplay

The iDisplay application supports the display of multiple sensors. The update rate of each sensor depends on the test speed and the processing effort of the running application. Sensors give priority to processing over display or logging.

Selecting Sensors

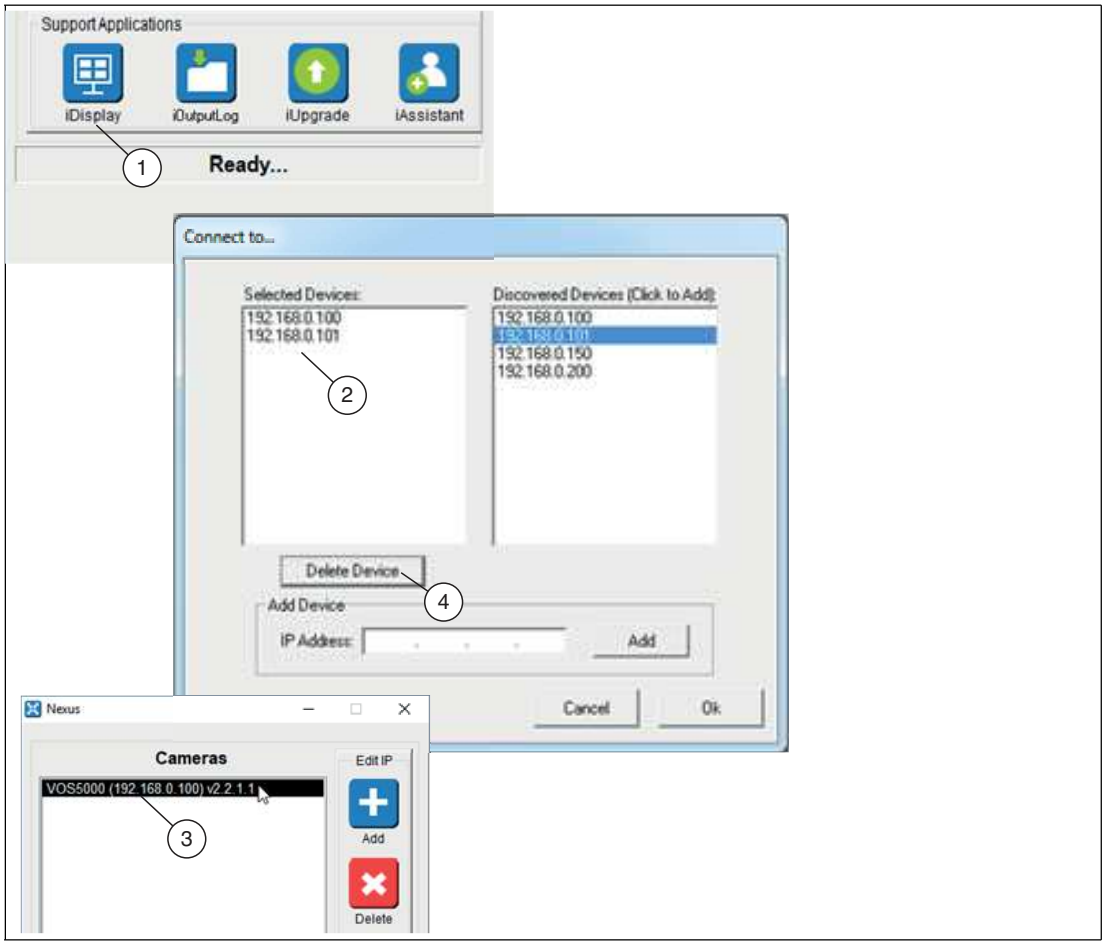


Figure 6.11 iDisplay

1. Click "iDisplay" (1).

   ↳ The application starts. When the application is first launched, a "Connect to..." configuration menu appears in which you must select which sensors "Selected Devices" (2) are to be displayed.

2. Click the required sensor (2) in the selection list. Clicking "Delete Device" (4) will remove sensors from the selection.

   ↳ The sensor is added to the "Cameras" list (3).

Note

The sensor configuration is stored on the respective PC to speed up successive application launches.

iOutputLog

This application provides the option to save results from connected sensors to CSV files on the PC. The CSV files can be viewed using an editor or copied to another system and processed further there.

Creating a CSV File

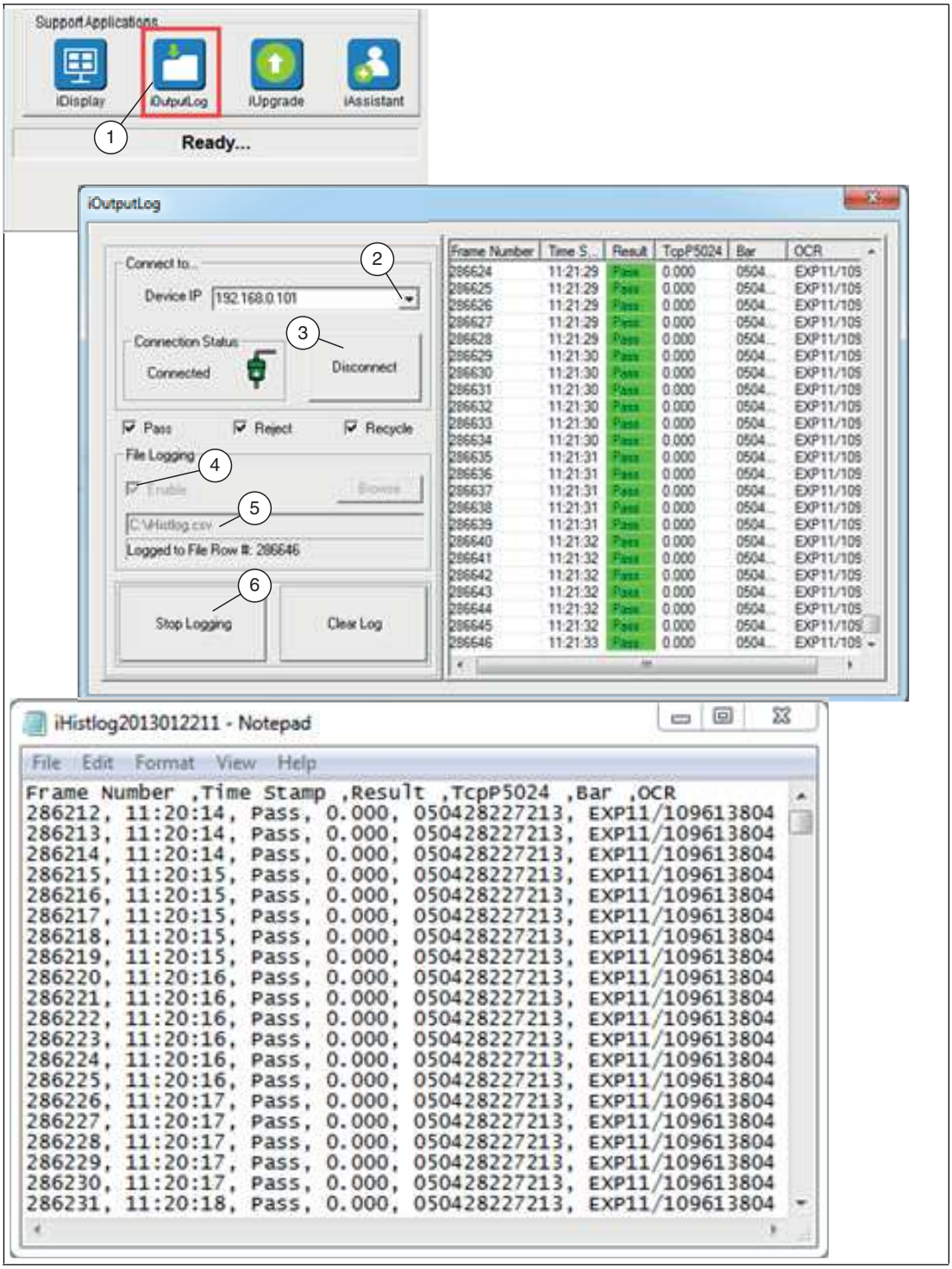


Figure 6.12 iOutputLog

2021-07

1. Click "iOutputLog" (1) to access the interface for setting up logging.
2. Select the "Device IP" (2) from which you want to log the results.
3. Click "Connect" (3).
4. Check the "Enable" selection check box (4) to enable file logging and enter a file name (5). The file name should be specific to the sensor you are connected to (e.g., VOS2000.csv).
5. Click "Start Logging" (6).

   ↳ Data recording is started.

Note

You can set up multiple logging operations one after the other. To do this, follow the steps above to create CSV files.

iUpgrade

This program provides the option to update sensors to a different firmware version. This may be necessary to access new functions, troubleshoot faults, or generally facilitate servicing. However, we do not recommend upgrading if the current firmware meets your expectations. Although we try to ensure solution compatibility, there may be problems upgrading to versions with newer functions.

Note

After each update, make sure that your settings are working properly.

Note

Please execute any safety-critical updates recommended by Pepperl+Fuchs.

Updating the Firmware Version

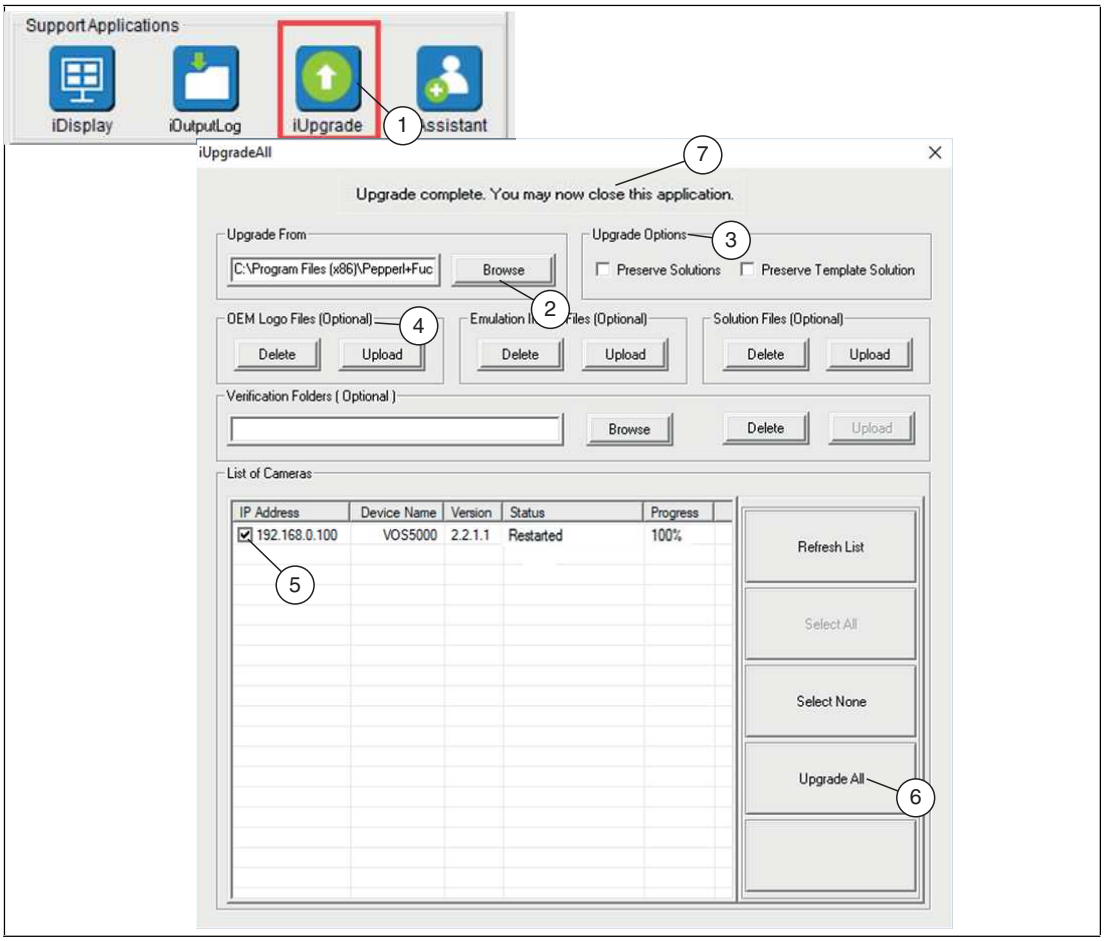


Figure 6.13 iUpgrade

1. Open the upgrade user interface using "iUpgrade" (1).
2. Click "Browse" (2) to access the firmware upgrade file from your PC.
3. Check one of the two check boxes (3) to ensure that existing job files for the sensors being updated are retained.
4. Use "Upload" (4) to install a custom logo on the Vision Configuration Tool screen.

2021-07

Note

The logo must be a 24-bit bitmap file with a size of 200 x 52 pixels.

5. Select the sensors you want to update from the "List of Cameras" selection list (5).
6. Click "Upgrade All" (6) to update all selected sensors.
7. Once successfully updated, close the window by clicking the "X" icon in the top right corner.

Note

The iUpgrade program is not integrated in the Nexus application. As a result, Nexus does not know when an update is complete. When you exit the program, Nexus waits 45 seconds for the updated sensor to complete its restart and then updates the device list.

iAssistant

Nexus manages the software version differences between the sensors connected to the network. In some cases, such as when the network configuration expands over time, Nexus automatically retrieves the files needed to connect to the sensor.

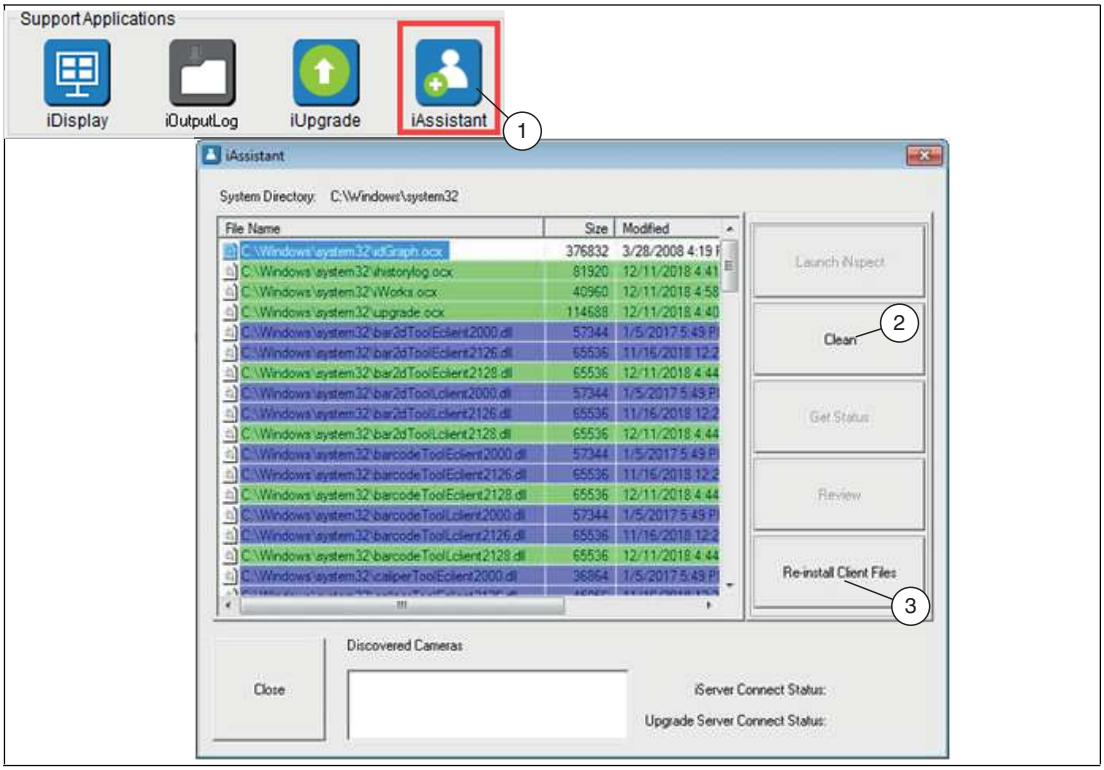


Figure 6.14 iAssistant

iAssistant (1) scans and reports device files on the PC that were installed when sensors were connected. In some cases, device files must be removed if there are incompatibilities between sensor software versions. This would be indicated by an "incompatible version" message when attempting to connect to the sensor. In this case, the system directory can be deleted from the files of the incompatible versions by clicking "Clean" (2) in the iAssistant user interface. New compatible files are installed when you reconnect to the sensor or by clicking "Re-install Client Files" (3).

- Unmarked: Indicates non-critical or unversioned files.
- Highlighted in green: Indicates that the files are the latest version found by Nexus. No action is required. Click "Close."
- Highlighted in blue: Indicates that Nexus has downloaded some files from a previous version to ensure compatibility. These are usually tool-specific files. No action is required. Click "Close."
- Highlighted in red: Indicates that important files are out of date. The Vision Configuration Tool may not work properly. If this is the case, the following steps must be performed:
  - Click "Clean" (2) to remove all files.
  - Click "Re-install Client Files" (3).
  - Wait for the list to populate again before closing it.
  - Click "Close."

Note

If you receive an error message indicating that not all files are deleted, check that your web browser and all sensor-related applications are closed. If you continue to receive errors, restart both your PC and the sensor and run iAssistant again.

7 Vision Configuration Tool

Once you have installed and powered the sensor, you can use the Vision Configuration Tool to access the sensor from a PC via a network connection, allowing you to then program and control the sensor.

Selecting the Sensor and Starting the Vision Configuration Tool

You will need a PC running the Nexus application. The sensor is connected via a local network or by direct connection via an Ethernet cable.

1. Start the Nexus software on your PC.

   ↳ A search is carried out for connected sensors, and these are displayed on the Nexus user interface under "Cameras."

2. Select your sensor from the "Cameras" list and click on the "Connect" button.

   ↳ The Vision Configuration Tool is launched.

Introduction to the Vision Configuration Tool User Interface

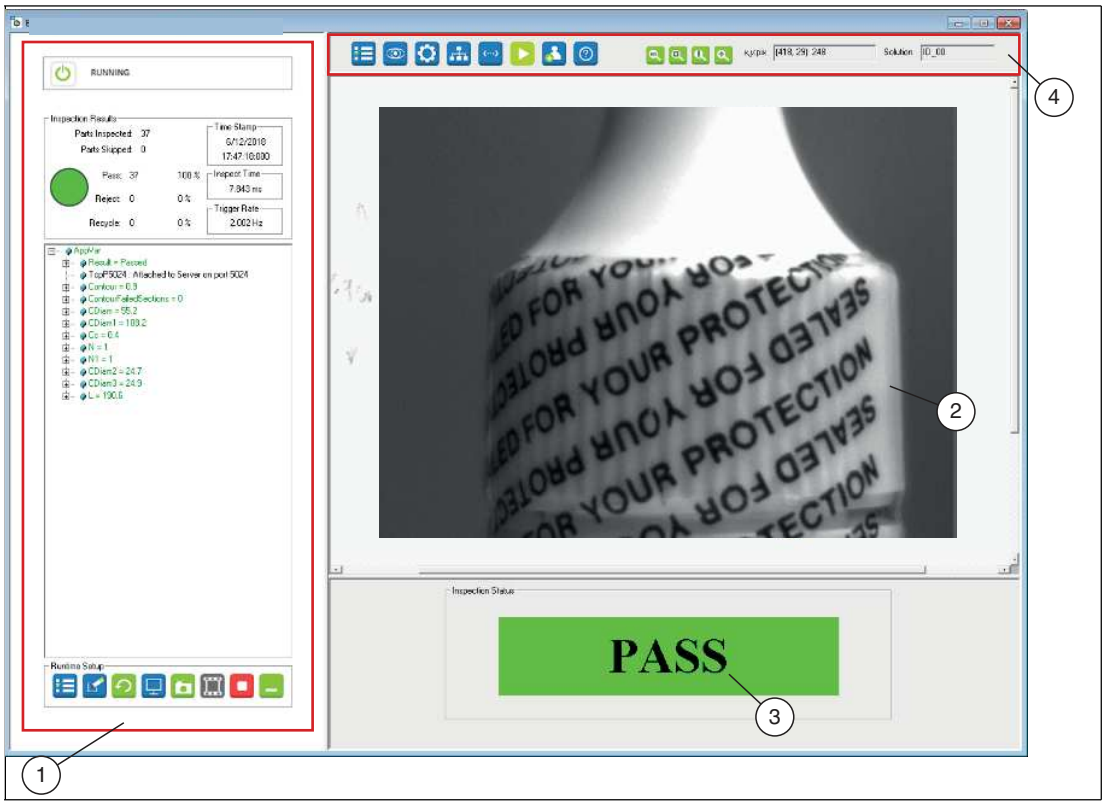


Figure 7.1 Vision Configuration Tool

The user interface of the Vision Configuration Tool consists of four areas. The left-hand side of the window (1) is used for instruction and setup. The large right-hand side of the window (2) is used to display the image and drawing tools. The lower right window range (3) is used for configuration and status display. The upper right window range (4) is the navigation bar. You can change the size of the windows.

Note

When setting up a sensor, the first five functions in the navigation bar—from left to right—are set. The process begins with creating a job, followed by setting the sensor, selecting the appropriate tools, and finally setting up the connection and editing the script. The test sequence can then be started.

Navigation Bar

Button	Description
	**Solution Setup**: Create a new job, load a saved job, import a job. Save, export, or delete a job file.
	**Sensor Setup**: Select the trigger source, enable the flash control, exposure, triggering.
	**Tool Setup**: Create control tools. Set good/bad tolerances for the check.
	**Setup Connections**: Create and configure a PLC connection, TCP/IP or RS-232 data stream settings. Set up I/O and image logging.
	**Edit Script**: Write customized programming scripts to expand solution functionality.
	**Run Solution**: Opens the dialog box for the test sequence. Shows the test image with tools, good/bad status, statistics, and variables.
	**User Administration**: Create and manage user accounts and passwords. Block users from editing functions.
	**Help**: Opens the online help.
	**Zoom Out**: Reduces the size of the image.
	**Stretch to Fit**: Zooms in or out to fill the current image display area.
	**Reset Zoom 1:1**: Restore the original image size.
	**Zoom In**: Increases the size of the image.
	**Mouse pointer (x,y:pix)**: Displays the mouse cursor position in X and Y coordinates and the pixel value.
	**Solution**: Displays the ID number and name of the job currently loaded.
	Closes the Vision Configuration Tool application window. The sensor continues to run.

7.1 Job Settings

Job files are configured and stored on the sensor memory. You can store up to 32 job files in the sensor memory. The actual number of jobs you can store depends on the image size and the application complexity (number of measurements, scripting, communication, etc.). You can export and import job files on your PC.

Eight jobs are pre-installed on the sensor. These cover the basic application of code reading. The default job has the ID 00. This job ensures that 1-D and 2-D codes are automatically read in the image field when the images are in focus.

Overview of Job Settings

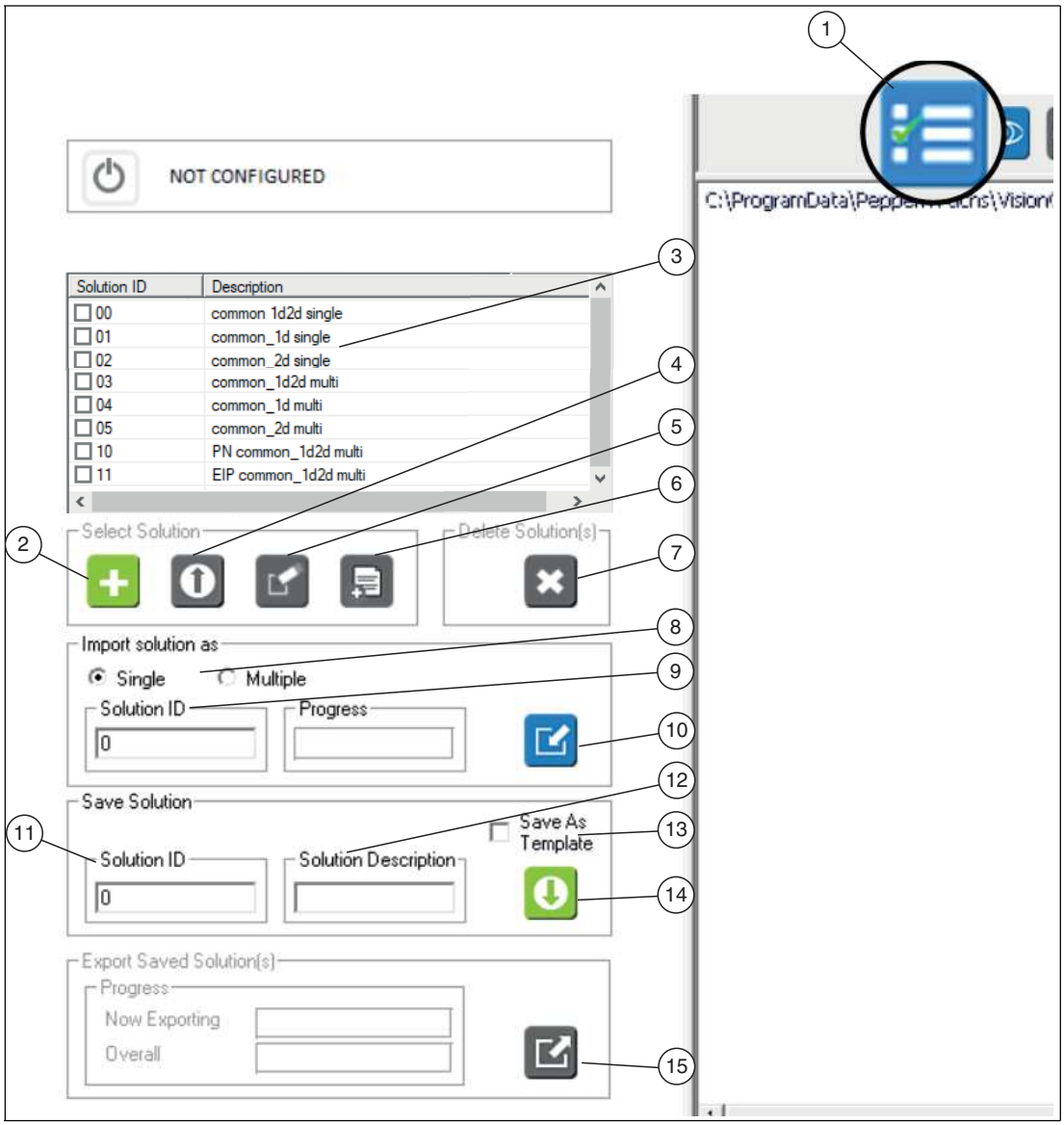


Figure 7.2 Job settings

Position	Function	Description
1	Solution Setup	Job Settings edit menu
2	Start New Solution	Loads a saved sample job or resets the sensor to the default state or "unconfigured" state. All parameters are reset except for the sensor settings. Saved jobs are not deleted from the sensor memory. Note: The "sample job" must be enabled in the "Application Settings" menu, which is launched from the Nexus application (see "Changing Application Settings" on page 37).
3	List of Solutions	A job will appear on this list once created and saved. In the "Solution ID" column, there are check boxes next to the ID number. Use these check boxes to select one or more jobs. Eight jobs are already pre-installed on the sensor. These cover the basic application of code reading: • Job 00: reading 1-D- and 2-D codes • Job 01: reading 1-D codes • Job 02: reading 2-D codes • Job 03: reading multiple 1-D and 2-D codes simultaneously (multi-code reading) • Job 04: reading multiple 1-D codes simultaneously • Job 05: reading multiple 2-D codes simultaneously • Job 10: reading multiple 1-D and 2-D codes simultaneously (multi-code reading) with PROFINET and RS-232 output • Job 11: reading multiple 1-D and 2-D codes simultaneously (multi-code reading) with EtherNet/IP and RS-232 output All other jobs are outputted via TCP/IP and RS-232.
4	Load Solution	Loads the selected job from the list as the current or in-progress job.
5	Load for Edit / Disable Switching	Disables all mechanisms for switching job files (except this operating panel). This is required for editing jobs that use automatic job switching based on variables or inputs. The sensor will not switch a job while you are editing it. You can re-enable switching by saving the edited job file.
6	Generate Report	Generates an HTML file that documents the sensor type, measurement tools and tolerances, variables, and scripting in the one job selected on the list of saved jobs. The job must be saved before the report is generated.
7	Delete Solution	Use the list to select one or more jobs. Click "Delete" to delete all selected jobs.
8	Import solution as	• Single: If you select the "Single" option, you can enter a number to assign to the imported job. • Multiple: If you select the "Multiple" option, you can navigate to a directory and select more than one job file. You cannot assign new job numbers. Only file names with the pattern "Solution##.bin" are displayed. Use the "Control" and "Shift" keys to generate multiple selected files.
9	Solution ID	Use this field to enter a number from 0 to 31 for a single job when the "Single" option is checked. This number becomes an index in the sensor memory. Enter a Job ID number before clicking "Load Solution."

2021-07

Position	Function	Description
10	Import Solution	This function loads previously saved jobs from the connected PC to the sensor memory. A Windows menu for opening files will appear. Click a single job or use the "Control" key to select multiple jobs.
11	Solution ID	Enter a number from 0 to 31 to be the job ID. This becomes an index in the sensor memory.
12	Solution Description	Give each job a description that you will recognize.
13	Save as Template	Saves the current job as "Template.bin." This file is loaded when you click "Start New Solution" if "Template Solution Enable" is enabled in the "Application Settings" menu, which is launched from the Nexus application (see "Changing Application Settings" on page 37).
14	Save Solution	Use "Save Solution" to save your job to the sensor memory.
15	Export	The "Export" function allows you to save one or more jobs from the sensor memory on your PC. You must first save your job to the sensor memory before you can export it. Use the check boxes in the job list to select which jobs are exported to the PC.

Starting a New Job

The following steps show how to create one or more jobs.

1. Click the "Solution Setup" button (1) in the navigation bar.

   ↳ The left side of the editing window displays the controls to select an existing job file or start a new job. If there are no saved job files, only the "Start New Solution" button (2) is enabled.

2. Click "Start New Solution" (2).

   ↳ If an existing job is running, this action deletes or overrides the current job. The software deletes any job that is in progress and launches the Vision Configuration Tool with the default settings. The sensor address, trigger, and exposure settings are not changed.

   ↳ When you click "Start New Solution," some of the buttons in the navigation bar are disabled. You must complete the sensor setup before the other buttons can be used. If you load a previously saved job, all buttons are enabled.

Note

Editing scripts and measurement tools changes the operating mode and disables the job change. Editing scripts can interrupt normal communication with PLCs and peripherals. Save your job and then reload it if you experience problems. Loading a job resets the operating mode unless you have selected the "Load for Edit / Disable Switching" (5) option.

7.2 Sensor Settings

Once you have connected the sensor, you can use the Vision Configuration Tool to access the sensor from a PC via a network connection, allowing you to program and control the sensor.

Overview of Sensor Settings

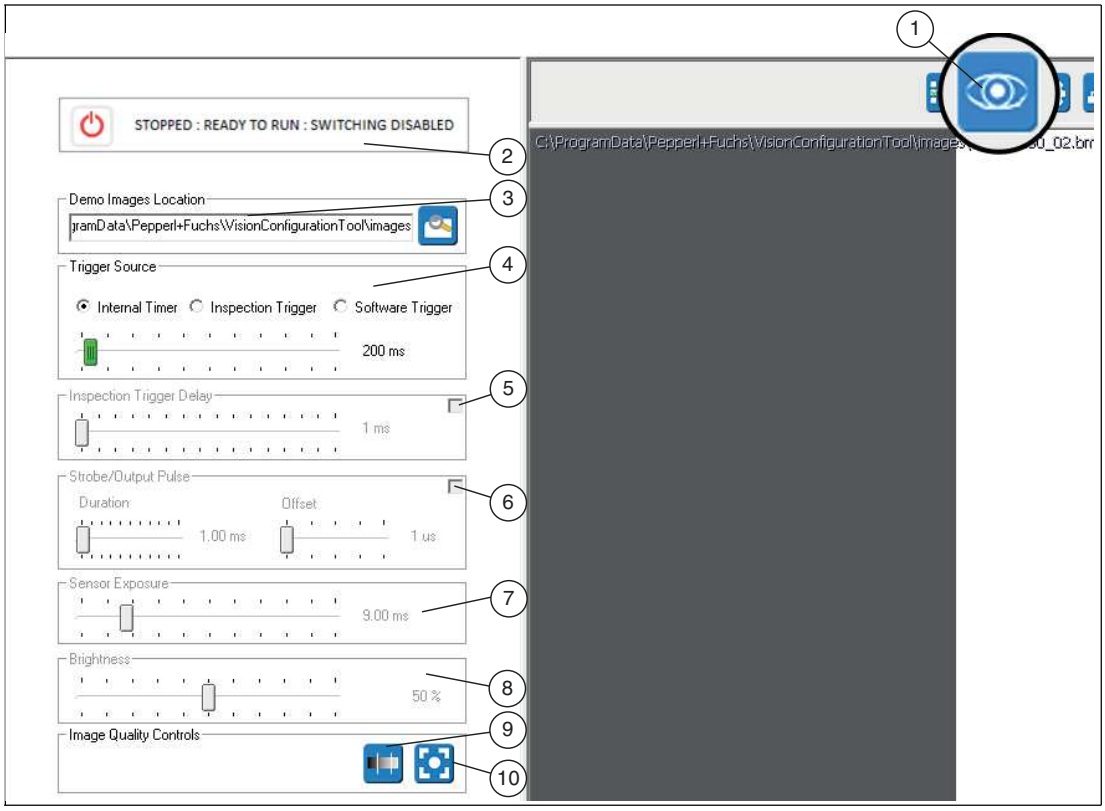
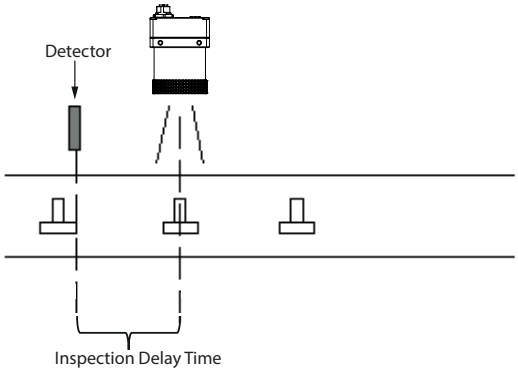
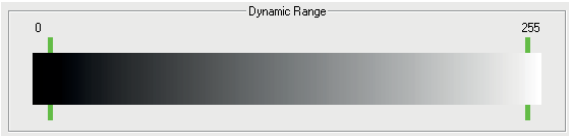
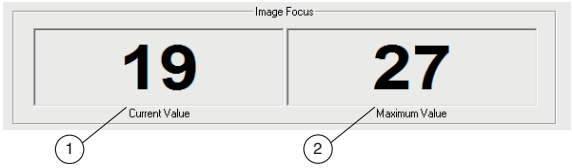


Figure 7.3 Sensor Settings

Position	Function	Description
1	Sensor setup	Sensor Settings editing menu
2	Status bar	Current status or execution mode
3	Demo Image Location	Location of the image file in the emulator
4	Trigger Source	Internal Timer: Use the slider to set the time between image captures.
		Inspection Trigger: The image capture is triggered by an external signal.
		Software Trigger: The image capture is triggered by a software command or a script function.

Position	Function	Description
5	Inspection Trigger Delay	To set the Inspection Trigger Delay, the "Inspection Trigger" trigger source must be selected. Activate the slider by clicking in the box (5). You can use the slider to set a delay between the time the object is under the trigger sensor and the time the object is under the camera. 
6	Strobe/Output Pulse Strobe/output pulse	Enable the slider by clicking in the box (6) to define a pulse output to control a light source. This setting is available if you have connected a sensor:
		Duration: time for how long the light source is on
		Offset: time delay of the light source
7	Sensor Exposure	The value of the sensor exposure can be adjusted using the slider.
8	Brightness	Use the slider to adjust the image brightness. The image brightness is increased if the image is too dark and you cannot increase the exposure time further because the object is moving. This setting is available if you have connected a sensor.

Position	Function		Description
9	Image Quality Controls	Dynamic Range	The dynamic range is expressed in number of exposure levels (light values) and shows the range between the brightest and darkest points of an image.  The green bars indicate the range of pixel values in your image. The goal is to set t0 for the lens opening, brightness, and exposure to give you the widest dynamic range.
10		Image Focus	In the focus quality display window, the left number shows the current focus quality (current value) (1). The right number is the last maximum value achieved (2). Adjust the lens to obtain the highest value. 

Setting up a Sensor

Click "Sensor Setup" (1) in the navigation bar to access the "Set up sensor" window.

↳ The instruction and setup window changes to display the sensor menu. Here you can set the trigger and sensor image settings. If you are using a conveyor belt or other moving parts, adjust the trigger delay, exposure, and brightness with moving parts.

Note

Images that appear dark to the human eye may still contain all necessary image information.

7.3 Vision Tools

This section describes how to use vision tools to solve test tasks. To provide you with the foundations for getting started, sample applications are described in the following sections.

Note

To use the vision tools, you need to have created a new job. Note the description: see chapter 7.1.

The following table provides an overview of the vision tools.

Overview of Vision Tools

Vision Tool	Symbol	Description
1-D code tool (barcode)		This function is used to read 1-D codes. See chapter 7.3.1.
2-D code tool (2-D barcode)		This function is used to read 2-D codes. See chapter 7.3.2.
Count tool		The count tool is used to create locators to detect codes on moving objects. See chapter 7.3.3.

Using Vision Tools

You can use vision tools once you have created a job and adjusted the sensor settings (see chapter 7.2). Note the following description:

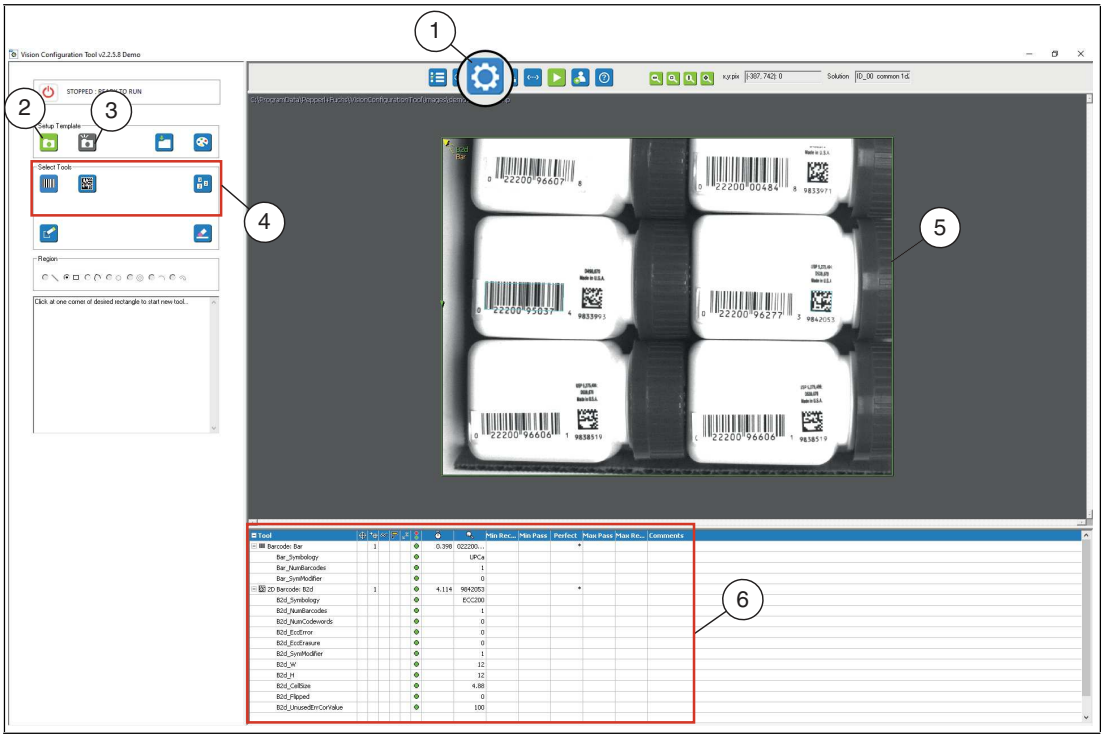


Figure 7.4 Main menu

2021-07

PEPPERL+FUCHS

1. Open the "Vision Tools" menu in the toolbar (1)
2. Take a picture of the object (2) or (3). This will be your reference image of the object being tested.

> **Note**
>
> "Take a photo" (2) captures an instant image. Use this function when working with stationary objects.
>
> "Trigger snapshot" (3) waits for a trigger signal or a software trigger. Use this function when working with moving objects.

3. Select the vision tools required for your test from the tool selection area (4).
4. Draw the search field in the image area (5).

> **Note**
>
> **Moving Objects**
>
> If your test object moves, you must first create a locator. Use the count tool (see chapter 7.3.3) to do this. Refer to the description in the online help of the Vision Configuration Tool.

↳ The selected field is labeled (B2d or Bar). The inner rectangle is the object area. The outer rectangle is the search area. The bottom window area displays the tools in the tool status table (6).

5. Open the Properties menu of the tool by right-clicking the search area outline (7) in the image area. You can right-click the corresponding tool in the tool status table.

Figure 7.5 Properties

↳ You can set the output variable in the Properties menu (8). The Properties menu also displays parameters for refining the tool operation or for enabling additional measurements.

Note

Preprocessing Filters

To improve specific properties of an image, preprocessing filters (9) can be used before a tool is used. Taking one or more preprocessing steps can simplify features that are difficult or impossible to measure. See chapter 7.3.4.

6. Click "OK" (10).

   ↳ The changes are applied and the editing field closes.

7. Click "Cancel" (11).

   ↳ The editing field closes without any changes being made.

8. Click "Apply" (12).

   ↳ The changes are applied without closing the editing field.

9. Click "Run" in the navigation bar when you are finished with the tool settings.

   ↳ You can test how well your tools are functioning and display the required test times.

10. Click "Connections" in the navigation bar to set up communication or inputs and outputs.

11. Click "Scripts" in the navigation bar to add additional instructions or functions to your application ().

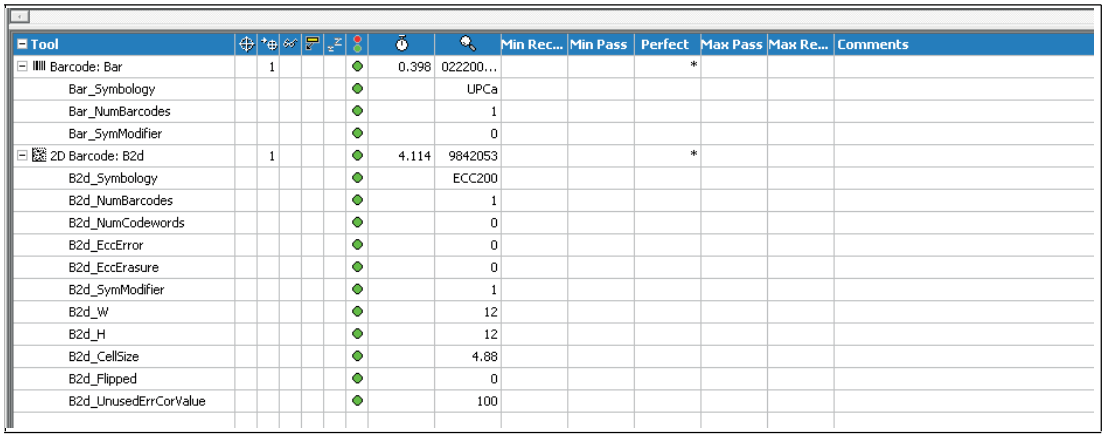
Note

Edge Threshold

The "Global Settings" function opens as soon as you right-click in the image area. Use the "Edge Threshold" slider to adjust the value as of which the edges (black-and-white transition) are highlighted. This function is not required to read codes. If you open this function accidentally, you can close the function by clicking the "X" icon.

Tool Status Table

When you use tools in the image area, these tools appear in the tool list in the configuration and status area. You can right-click a tool's name to open the "Properties" editing menu for that tool. The columns and headings are explained in the online help. You can change the width of the columns by dragging the dividing lines in the table header. The comment box does not appear in other operating windows.



Tool							Min Rec...	Min Pass	Perfect	Max Pass	Max Re...	Comments
Barcode: Bar		1				0.398	022200...			*		
Bar_Symbology							UPCa					
Bar_NumBarcodes							1					
Bar_SymModifier							0					
2D Barcode: B2d		1				4.114	9842053			*		
B2d_Symbology							ECC200					
B2d_NumBarcodes							1					
B2d_NumCodewords							0					
B2d_EccError							0					
B2d_EccErasure							0					
B2d_SymModifier							1					
B2d_W							12					
B2d_H							12					
B2d_CellSize							4.88					
B2d_Flipped							0					
B2d_UnusedErrCorValue							100					

Function	Description
Tool	Controls for tool names: Right-click a tool's name or click the tool's symbol on the list to open the "Properties" edit field for that tool. Click [-] to close the list of dependencies for a tool or [+] to expand it. Click the "Tool" heading to show or hide the list. Header controls: Click the "Tool" heading to reverse the order of the list. Click [-] to collapse the list or [+] to expand it. Right-click the "Tool" heading to open a tool list property field in which you can filter tools by type or name. The text box is case sensitive.
 Locator Definition	Locator definition displays which tools are used to create locators. The tool and the active points are specified. "P" stands for position locator and "A" for rotation locator.
 Locator Assignment	Locator assignment indicates which locator the tool is assigned to.
 Visibility	Hides or shows tools. Click in this column to hide a tool in the image area. An "x" appears, and the tool is grayed out in the list to indicate that the tool is hidden. Click a second time to show/hide the tool again. Click the column header to show or hide all tools. This can be useful if you have tools that overlap and have problems choosing the right tool. In the "Run" operating panel, a hidden tool does not appear in the image area. The measured value appears on the list of values.
 Toggle Labels	Hides or shows tool labels. Click this symbol in the column header to show/hide labels for all tools. An "x" in this column indicates that a label is hidden. This can be useful if you have a lot of tools or a lot of found objects and the labels overlap the display.
 Enable/Disable	Enable or disable tools. Click in this column to disable a tool. The tool is hidden in the image area and grayed out in the tool list, and a small "x" in this column indicates that a tool is disabled. In the "Run" operating panel, a disabled tool is not displayed in the image area, and its measured value is always zero. (You can also use a script to disable or enable a tool).

Function	Description
 Pass/Fail	A red circle indicates that the measurement has failed. A green circle indicates that the measurement has passed.
 Execute time	The estimated execution time, in milliseconds, for the individual tool.
 Measured value	The current measured value.
Min Pass	Minimum tolerance for Recycle/Pass. "Recycle" if the measured value is less than this value. "Pass" if the measured value is greater than or equal to this value.
Perfect	The set "perfect" value.
Max Pass	Maximum tolerance for Pass/Recycle. "Pass" if the measured value is less than or equal to this value. "Recycle" if the measured value is greater than this value.
Comments	You can enter comments here. The comments only appear in the tool list. The comments do not appear on other screens.

Note

General Tool Information (Vision Tools)

- Enlarging the search field increases the test time.
- While drawing a tool, you can cancel by right-clicking. This is useful if you have selected the wrong tool.
- Adding a preprocessing filter may increase the test time. See chapter 7.3.4.

7.3.1 1-D Code (Barcode Tool)

The 1-D code tool is used to read the most common barcodes. For codes that are detected, refer to the tables below in this section.

Multiple codes can be detected in a search field (multiple code reading) with optional output of the code position in the image. Alternatively, you can span multiple search fields in the image to read codes at predefined positions. Enable the grading output to monitor the print quality of the code.

Reading Codes

To enable reliable implementation of your application, the codes must have good image quality, contrast, and the right resolution to be read.

Observe the following instructions for reading codes with the sensor.

- Select the operating distance between the sensor and the code using the read field diagrams above. To do so, the **"module size"** must be known, i.e., the width of the narrowest bar (for 1-D codes) or the side length of the square for Data Matrix codes (2-D codes). If the operating distance selected is too large, reading is either unreliable or not possible.
- The values specified in the read field diagrams are maximum values for a specific module size. For a specific module size, the operating distance must be less than or equal to the specified distance.
- Adjust the operating distance and the focus so that the bars or squares in the image are clearly distinguishable. If the bars or squares blur together, the images cannot be read correctly.
- Ensure that the code to be read is completely within the field of view.
- Ensure uniform lighting of the code to be read. Inhomogeneous code lighting may lead to incorrect readings.
- During parameterization, note the information in the following description for the 1-D code tool.
- Test the settings and make sure that your settings enable process-safe reading. We recommend trying out multiple codes when testing your application. Consider possible changes in lighting conditions along with the position in the test.

Using the 1-D Code Tool

1. Click "1-D Code Tool" in the tool selection area.

   ↳ Move the mouse over the image in the work area.

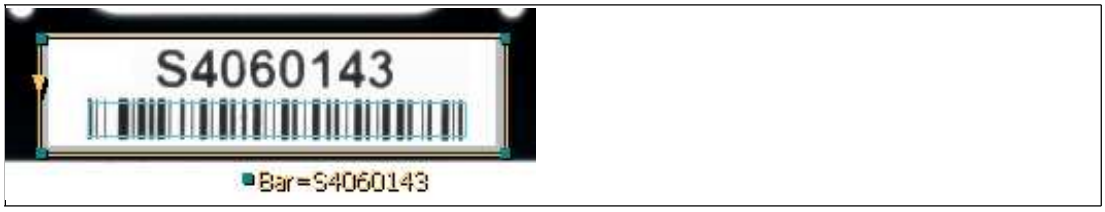


Figure 7.6 1-D code tool

2. Left-click on your starting point. Release the left mouse button and move the mouse to the area that represents your search area. Left-click again to draw a rectangular search area on the image.

   ↳ The selected field is labeled (Bar=S4060143).

3. Resize the search area by clicking and dragging one of the corners of the search area. The maximum size of the search area for 1-D codes is 2048 x 2048 pixels

4. Right-click on an edge of the rectangular search area.

   ↳ The properties window for the 1-D code tool will open.

1-D Code Tool Properties Window

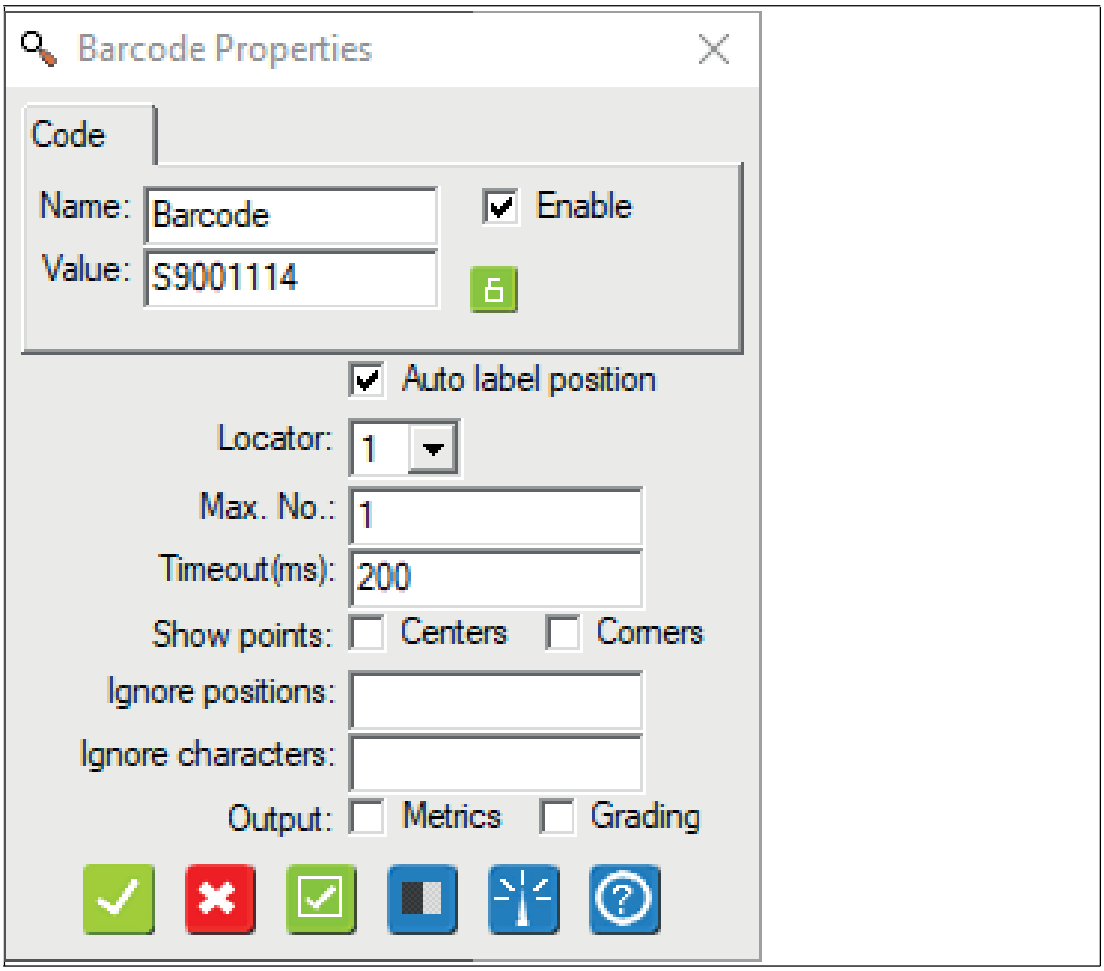


Figure 7.7 Properties window

Output Variable

Tab	Description
Code	The tool will operate differently depending on the mode set. • Read code: The code is read and the content can be output. • Compare code: The code is read and checked to see whether the content matches the specified content. You can enter a fixed value under "Value."

Output Variable Settings

Setting	Description
Name	You can change the variable name of the tool or the measurement. The variable name is displayed in the image, in the tool list, and in the measurement results. The name is used as a variable in scripts.
Enable	Enables the data output of the respective output variable.
Value	In this field, enter the character string to be compared with the content of the current barcode. The result of the tool is "passed" when the code is read and the read character string matches the set character string. Otherwise, the result is "failed." Leave an asterisk in this field if the 1-D code changes. The measurement reports any read value. It only fails if no value can be read. "?" = placeholder for a single character "*" = placeholder for multiple characters. The "padlock" allows you to lock the value so that it will not change while you move the search field. The value should be unlocked when you are changing the parameter settings.

Code Properties

Parameter	Description
Auto label position	Sets the name field next to the area in the display. Clear the check box to move the name field to another position, or drag the name field to the image with the property field closed.
Locator (Position Reference)	This function causes the search field of the tool to follow the test object as it moves. Select a locator (by number) that you want to follow. To detach the tool from the locator or to hold the tool in a fixed position in the image area, select "X." You can define up to four locators.
Max. No.	Number of codes expected in the search field. Increase this number if you have more than one code in your search field. You can define a search field with multiple codes or multiple search fields, each with one code. The maximum number is 255. Larger numbers result in an error message.
Timeout(ms)	This function sets a timeout (in milliseconds) for complicated, incorrect, or non-existent codes. Note: This timeout refers to the decoding or code reading algorithm but does not include the preprocessing time.
Show points	This function displays active points and allows you to use them as point coordinates in the image. • Centers: Centers the center of the boundary box surrounding the code. • Corners: The four corner points of the boundary box that surround the code are placed in the corners.
Ignore positions	This function skips or ignores character positions in the character string. You can ignore unimportant, unrelated, or changing positions. Enter position numbers separated by spaces. Use 1 for the first character position (not 0), use 2 for the second character, etc. Use -1 for the last character, and use -2 for the second-last character.

Parameter	Description
Ignore characters	This function skips or ignores characters that are partially added to some barcodes. Example: If the first position and the $ characters are ignored, a decoded string of 1234 will have a result value of 1234.
Output	Output of the code quality according to ISO. The output enables additional output data. • Metrics: Indicates the code symbology detected and the number of read codes. • Grading: Provides information about the readability and print quality of the codes.

Buttons

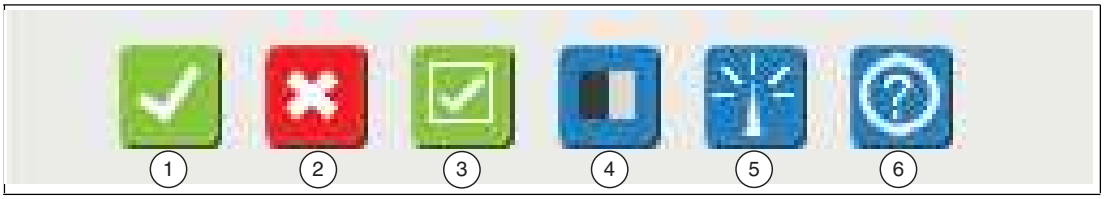


Figure 7.8 Properties window buttons

Position	Button	Description
1	OK	The changes are applied and the editing window is closed.
2	Cancel	The editing window closes without any changes being made.
3	Apply	The changes are applied without closing the editing window.
4	Preprocessing Filter (Preprocess)	This function allows you to add image filtering to the tool. The preprocessing filter is optional. You should add a preprocessing filter before adjusting or locking the other parameters. A preprocessing filter will extend the runtime of the test. See chapter 7.3.4.
5	Advanced	These settings are optional. They allow more control over the object and the runtime. You can zoom in and out on areas and hide them. Hiding areas allows you to cut them from your search area by using a brush to turn these areas red. This function increases the test time.
6	Help	This opens the help window. Here you will find useful operational information.

Editing the 1-D Code Tool

1. Right-click on an edge of the rectangular search area.

   ↳ This opens the editing field for the 1-D code tool.

2. Click on "Preprocess" (see chapter 7.3.4) to add an image filter to the tool.

   Note

   The preprocessing filter is optional. You should add a preprocessing filter before adjusting or locking the other parameters. A preprocessing filter will extend the runtime of the test.

3. Click on the padlock symbol next to the value to store a fixed value.
4. Click "OK".

   ↳ The changes are applied and the editing field closes.

5. Click "Cancel".

   ↳ The editing field closes without any changes being made.

6. Click "Apply".

   ↳ The changes are applied without closing the editing field.

7. Click "Help" (10).

   ↳ This opens the help window. Here you will find useful operational information.

Advanced Settings

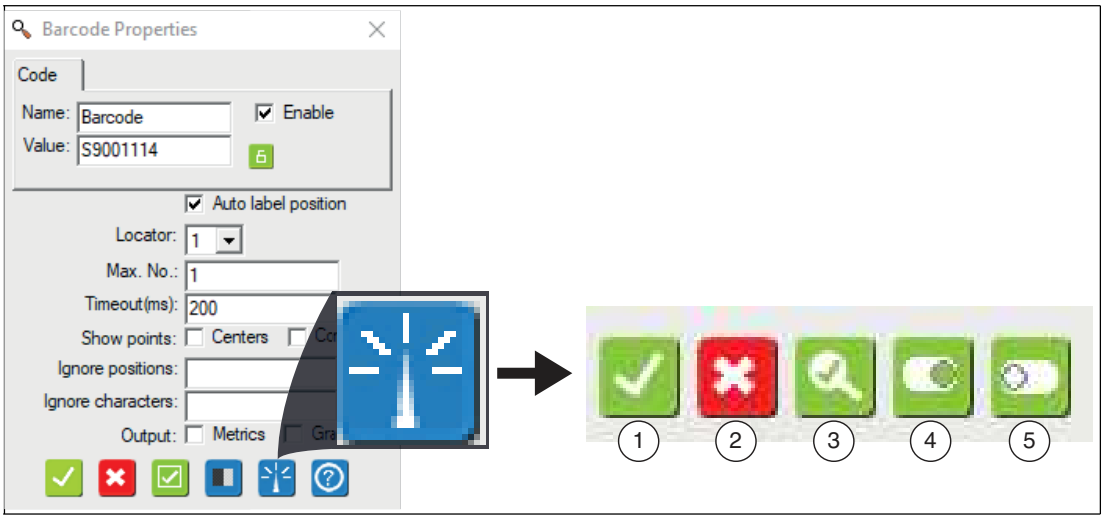


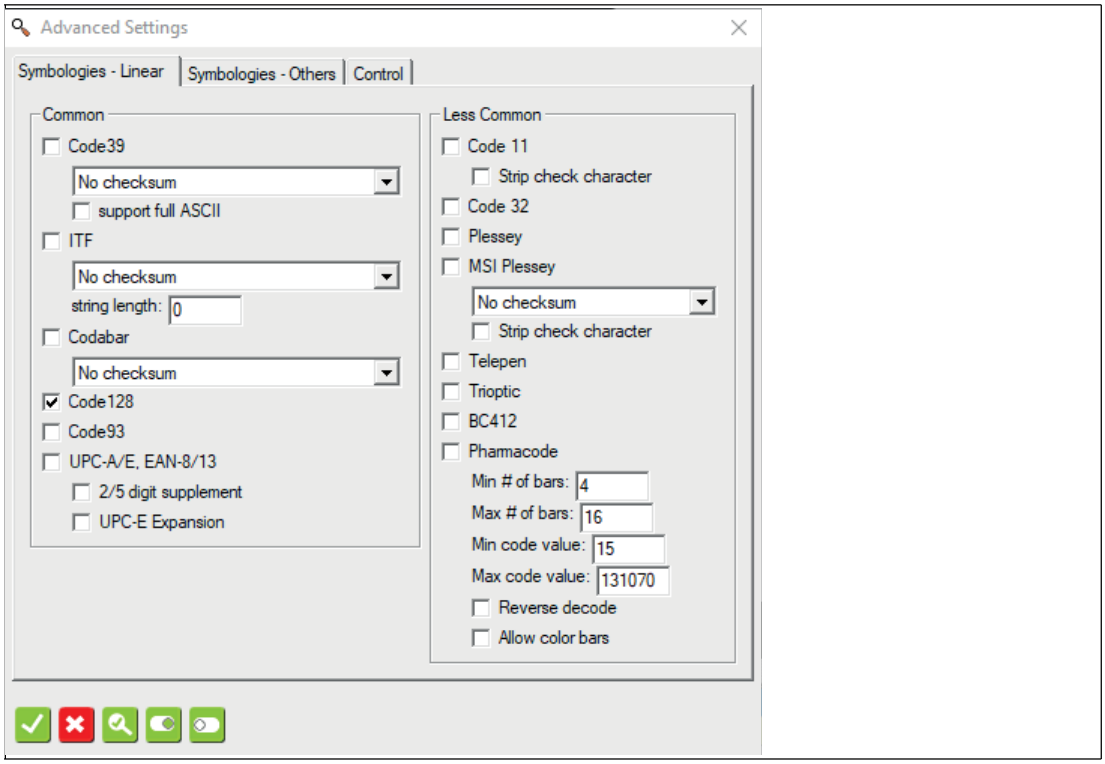
Figure 7.9 Advanced settings in the menu bar

The buttons are the same for all tabs

Position	Button	Description
1	OK	Changes are applied and the editing field is closed.
2	Cancel	The editing window closes without any changes being made.
3	Detect and set symbology	The code type is automatically identified and selected in the search field.
4	Enable all symbologies	All code types or symbologies are enabled on both tabs.
5	Disable all symbologies	All code types are disabled on both tabs.

Tab Description

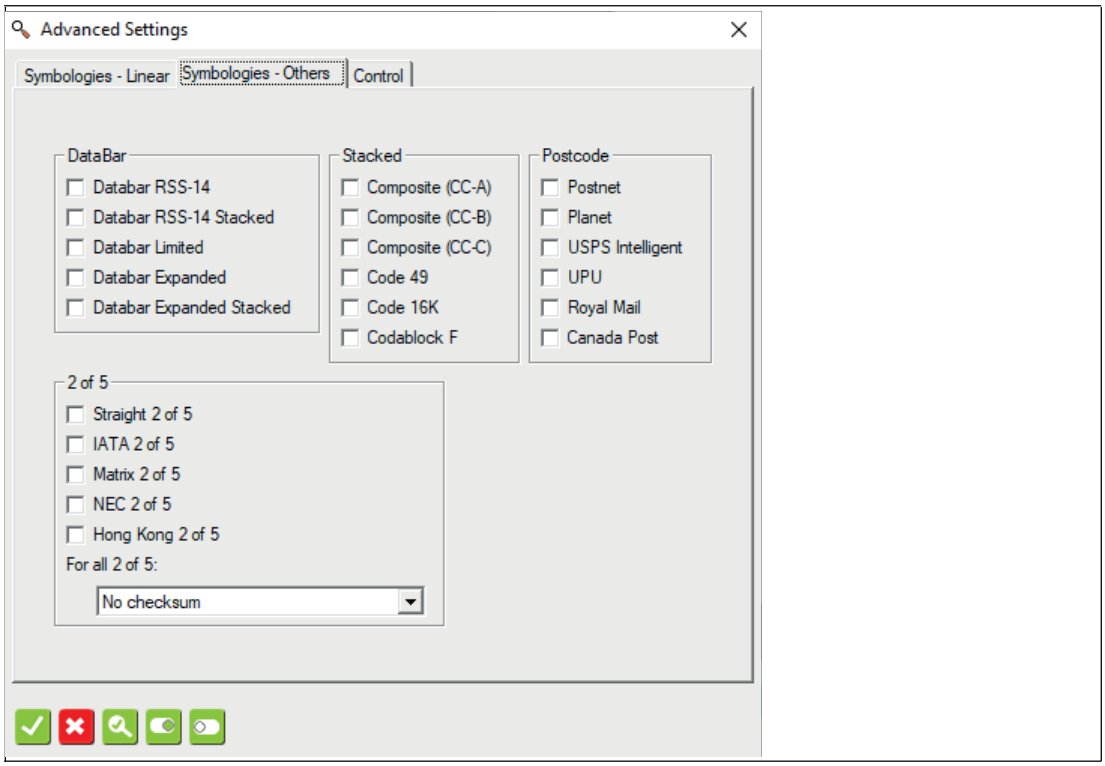
Symbologies - Linear



Menu	Description
Code 39	Enables Code 39 codes to be decoded. The following checksum options are available for selection: • **No checksum**: Disables checksum calculation. • **Use checksum, output check character**: Calculates the checksum and includes the check character in the output of the read character string. • **Use checksum, strip check character**: Calculates the checksum and does **not** include the check character in the output of the read character string. **Support full ASCII**: Enable or disable full ASCII character support for Code 39. If disabled, the output is numeric.
ITF	Enables ITF (Interleaved 2 of 5) to be decoded The following checksum options are available for selection: • **No checksum**: Disables checksum calculation. • **Use checksum, output check character**: Calculates the checksum and includes the check character in the output of the read character string. • **Use checksum, strip check character**: Calculates the checksum and does **not** include the check character in the output of the read character string. **String length**: Sets a minimum string length and controls the check of the quiet zone on ITF. • **0**: The default quiet zone check is used and no check is performed on the code length. • **1**: A short quiet zone is allowed and no length check is performed. • **event number**: The default quiet zone is used and the string to be decoded must be at least N-long to be decoded. • **odd number larger than 1**: A short quiet zone is allowed and the decoding string must be at least N-1 long to be decoded.

Menu	Description
Codabar	Enables Codabar codes to be decoded. The following checksum options are available for selection: • **No checksum**: Disables checksum calculation. • **Use checksum, output check character**: Calculates the checksum and includes the check character in the output of the read character string. • **Use checksum, strip check character**: Calculates the checksum and does **not** include the check character in the output of the read character string.
Code 128	Enables Code 128 codes to be decoded.
Code 93	Enables Code 93 codes to be decoded.
UPC-A/E, EAN-8/13	Enables UPC-A, UPC-E, EAN-8, and EAN-13 codes to be decoded. **2/5 digit supplement**: Enables 2- or 5-digit additional code for UPC and EAN codes. **UPC-E Expansion**: Enables UPC-E expansion.
Code 11	Enables Code 11 codes to be decoded. **Strip check character**: Clears the checksum character from Code 11.
Code 32	Enables Code 32 codes to be decoded.
Plessey	Enables Plessey codes to be decoded.
MSI Plessey	Enables MSI Plessey codes to be decoded. The following checksum options are available for selection: • **No checksum**: Disables checksum calculation. • **Use Mod 10 checksum** • **Use Mod 10/10 checksum** • **Use Mod 11/10 checksum** **Strip check character**: Clears the checksum character from MSI Plessey codes.
Telepen	Enables Telepen codes to be decoded.
Trioptic	Enables Tripotic codes to be decoded.
BC412	Enables BC412 codes to be decoded.
Pharmacode	Enables Pharmacode codes to be decoded. **Min # bars** Sets the minimum number of bars in the Pharma Code. **Max # bars** Sets the maximum number of bars in the Pharma Code. **Min code value** Sets the minimum numeric code value. **Max code value** Sets the maximum numeric code value. **Reverse decode** Enables decoding of reversed images or reverse decoding. **Allow color bars** Allows color codes to be decoded.

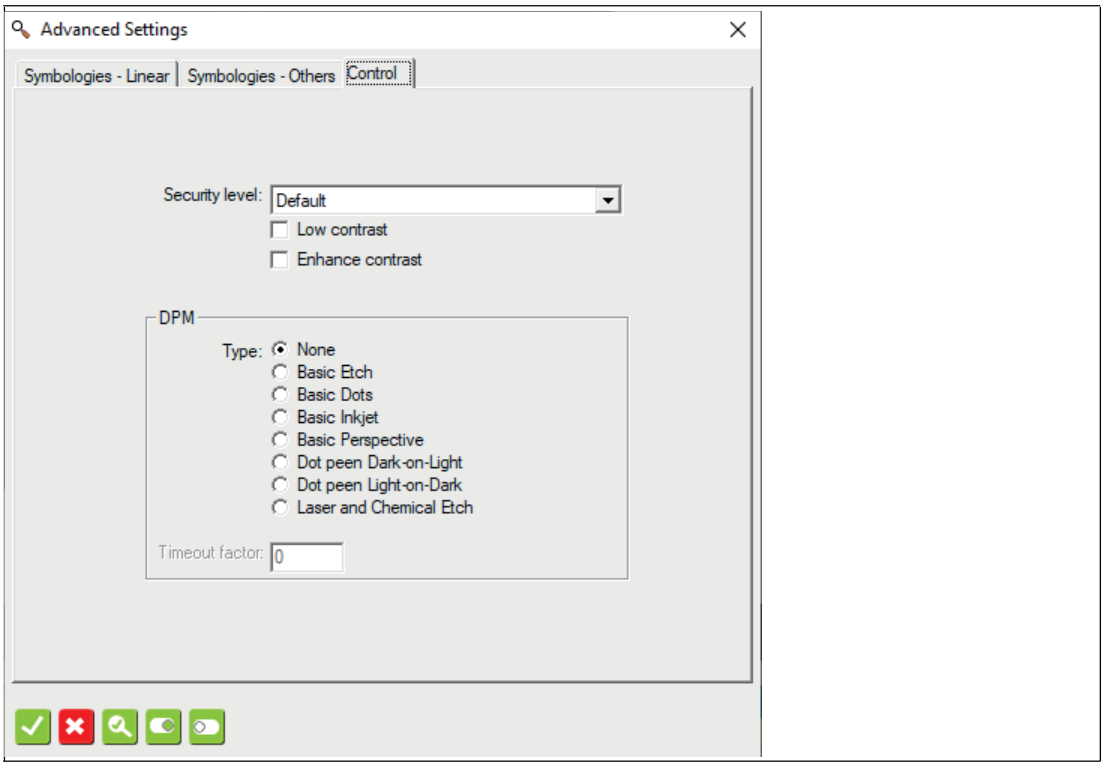
Symbologies - Others



Select the types of codes you want to read or decode. You can make multiple selections if your code types vary. This may lengthen the inspection time.

Menu	Description
DataBar	Select which types of DataBar codes to decode.
Stacked	Select which types of stacked codes to decode.
Postcode	Select which types of postal codes to decode.
2 of 5	Select which types of codes to decode. The following checksum options are available for selection: • **No checksum**: Disables checksum calculation. • **Use checksum, output check character**: Calculates the checksum and includes the check character in the output of the read character string. • **Use checksum, strip check character**: Calculates the checksum and does **not** include the check character in the output of the read character string.

Control



Menu	Description
Security level	Selects the read or decode aggressiveness level to be used. A higher aggressiveness level means "more aggressive read or decode attempts." If the print quality is poor or the resolution is low (code too small), the number of misread or incorrectly decoded codes may increase and the execution time may be lengthened. Reducing the aggressiveness level reduces the number of misreadings or incorrect decodings of poor-quality or low-resolution codes. Applies to: Code 128, Code 39, UPC/EAN, ITF, Codabar, Code 93.
Low contrast	Enables codes with low contrast or reverse contrast to be decoded. This may lengthen the execution time. Applies to: Code 128, Code 39, UPC/EAN, ITF, Codabar, Code 93.

Menu	Description
Enhance contrast	Allows the image to be improved in order to decode poorly printed codes. This may lengthen the execution time. Cannot be used with Dot peen Dark-on-Light, Dot peen Light-on-Dark, or Laser and Chemical Etch.
DPM Type	Selects image conditioning options specifically for direct part marking (DPM). Note: For all DPM options, the maximum search field size is 614,000 pixels. • **None**: Does not allow DPM-specific processing. • **Basic Etch**: Optimized for simple laser or chemical etching. • **Basic Dots**: Point-jet and inkjet images • **Basic Inkjet**: Basic inkjet images and poor-quality inkjet images. • **Basic Perspective**: Reads laser markings and normal codes of good quality that are centered in the image but have strong perspective distortion. This mode is faster but less robust than "Basic Etch" for reading poor-quality codes. • **Dot peen Dark-on-Light**: This is the most robust method for reading dark dot peen codes on a light background. • **Dot peen Light-on-Dark**: This is the most robust method for reading light dot peen codes on a dark background. • **Laser and Chemical Etch**: This is the most robust method for reading laser or chemical etching point codings. **Timeout factor**: The timeout factor sets a timeout for difficult, deteriorated, or non-existent DPM codes. There is no timeout if this field = 0. The actual timeout for the device is ten times the number entered here in milliseconds. Note: This timeout does not apply to None, Basic Etch, or Basic Dots. This timeout applies only to Basic Inkjet, Basic Perspective, Dot peen Dark-on-Light, Dot peen Light-on-Dark, and Laser and Chemical Etch.

7.3.2 2-D Code (2-D Code Tool)

This tool is used to read 2-D codes.

Multiple codes can be detected in a search field (multiple code reading) with optional output of the code position in the image. Alternatively, you can span multiple search fields in the image to read codes at predefined positions. Enable the grading output to monitor the print quality of the code.

Reading Codes

To enable reliable implementation of your application, the codes must have good image quality, contrast, and the right resolution to be read.

Observe the following instructions for reading codes with the sensor.

- Select the operating distance between the sensor and the code using the read field diagrams above. To do so, the **"module size"** must be known, i.e., the width of the narrowest bar (for 1-D codes) or the side length of the square for Data Matrix codes (2-D codes). If the operating distance selected is too large, reading is either unreliable or not possible.
- The values specified in the read field diagrams are maximum values for a specific module size. For a specific module size, the operating distance must be less than or equal to the specified distance.
- Adjust the operating distance and the focus so that the bars or squares in the image are clearly distinguishable. If the bars or squares blur together, the images cannot be read correctly.
- Ensure that the code to be read is completely within the field of view.
- Ensure uniform lighting of the code to be read. Inhomogeneous code lighting may lead to incorrect readings.
- During parameterization, note the information in the following description for the 1-D code tool.
- Test the settings and make sure that your settings enable process-safe reading. We recommend trying out multiple codes when testing your application. Consider possible changes in lighting conditions along with the position in the test.

Using the 2-D Code Tool

1. Click "2-D Code Tool" in the tool selection area.
2. Select the shape of the "Region" from the options that appear below the tool selection window.

Note

- Use a rectangle for codes that are close to 0°, 90°, 180°, and 270° rotations in your template image.
- Use a polygon to draw a rectangle at any angle in the template image.
- Use an arc area (area between two arcs or curves) to read a circle of printed code.

↳ Move the mouse over the image in the work area.

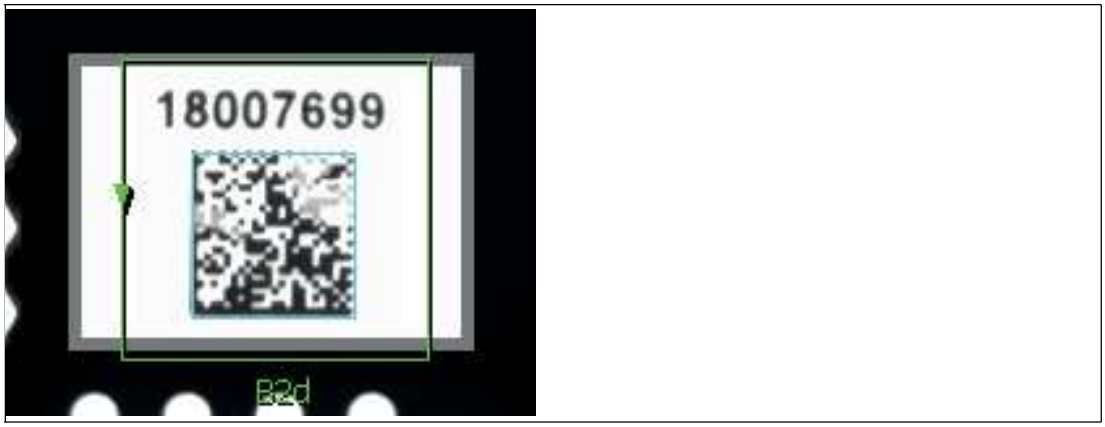


Figure 7.10 2-D code tool

3. Left-click on your starting point. Release the left mouse button and move the mouse to the area that represents your search area. Left-click again to draw a search area on the image.

   ↳ The selected field is labeled (B2d).

4. Resize the search area by clicking and dragging one of the corners of the search area.

5. Right-click on an edge of the rectangular search area.

   ↳ The properties window for the 2-D code tool will open.

2-D Code Tool Properties Window

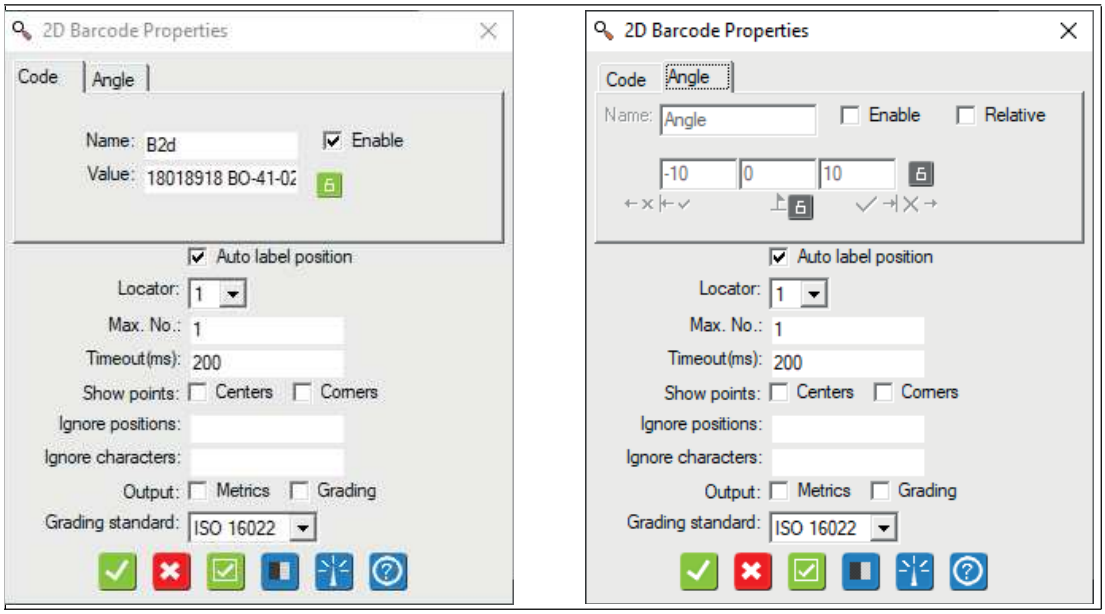
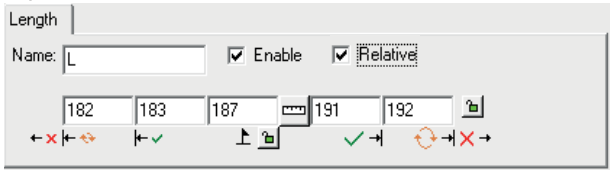


Figure 7.11 Properties window—"Code" and "Angle"

Output Variables

Tab	Description
Code	The tool will operate differently depending on the mode set. • Read code: The code is read and the content can be output. • Compare code: The code is read and checked to see whether the content matches the specified content. You can enter a fixed value under "Value."
Angle	The angle of the code defined by a line drawn from the top left corner to the top right corner in relation to the horizontal or the top edge of the image area. Clockwise is positive.

Output Variable Settings

Setting	Description
Name	You can change the variable name of the tool or the measurement. The variable name is displayed in the image, in the tool list, and in the measurement results. The name is used as a variable in scripts.
Enable	Enables the data output of the respective output variable.
Relative	If you enable this function by checking the box, the difference to the "Perfect" tolerance value is reported and not the actual measured values. For example: The set value of "Perfect" is 187 and the measured value is 189. The difference from "Perfect" is 2. If the "Relative" check box is cleared, the measured value 189 is reported. If the "Relative" check box is selected, 2 is reported. 
Tolerance	See "Setting the Tolerance Values" on page 71.
Value	In this field, enter the character string to be compared with the content of the current code. The result of the tool is "passed" when the code is read and the read character string matches the set character string. Otherwise, the result is "failed." Leave an asterisk in this field if the 2-D code changes. The measurement reports any read value. It only fails if no value can be read. "?" = placeholder for a single character "*" = placeholder for multi-digit characters The "padlock" allows you to lock the value so that it will not change while you move the search field. The value should be unlocked when you are changing the parameter settings.

Code Properties

Parameter	Description
Auto label position	Sets the name field next to the area in the display. Clear the check box to move the name field to another position, or drag the name field to the image with the property field closed.
Locator (Position Reference)	This function causes the search field of the tool to follow the test object as it moves. Select a locator (by number) that you want to follow. To detach the tool from the locator or to hold the tool in a fixed position in the image area, select "X." You can define up to four locators.
Max. No.	Number of codes expected in the search field. Increase this number if you have more than one code in your search field. You can define a search field with multiple codes or multiple search fields, each with one code. The maximum number is 64. Larger numbers result in an error message.
Timeout(ms)	This function sets a timeout (in milliseconds) for complicated, incorrect, or non-existent codes. Note: This timeout refers to the decoding or code reading algorithm but does not include the preprocessing time.
Show points	This function displays active points and allows you to use them as point coordinates in the image. • Centers: Centers the center of the boundary box surrounding the code. • Corners: The four corner points of the boundary box that surround the code are placed in the corners.
Ignore positions	This function skips or ignores character positions in the character string. You can ignore unimportant, unrelated, or changing positions. Enter position numbers separated by spaces. Use 1 for the first character position (not 0), use 2 for the second character, etc. Use -1 for the last character, and use -2 for the second-last character.

2021-07

Parameter	Description
Ignore characters	This function skips or ignores characters that are added to some codes. Example: If the first position and the $ characters are ignored, a decoded string of 1234 will have a result value of 1234.
Output	Output of the code quality according to ISO. The output enables additional output data. • Metrics: Indicates the code symbology detected and the number of read codes. • Grading: Provides information about the readability and print quality of the codes.
Grading standard	The selection list allows you to specify which quality standard is used for the classification calculation. The code quality can be assessed based on the following standards: ISO/IEC 15415, ISO/IEC 16022, AIM DPM, AS 9132. The code is evaluated according to the criteria of the standards with the existing lighting. The evaluation is therefore only based on the standards, since this vision sensor does not have standardized lighting.

Buttons

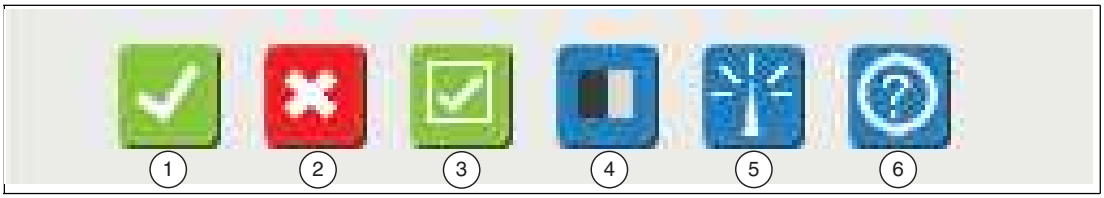


Figure 7.12 Properties window buttons

Position	Button	Description
1	OK	The changes are applied and the editing window is closed.
2	Cancel	The editing window closes without any changes being made.
3	Apply	The changes are applied without closing the editing window.
4	Preprocessing Filter (Preprocess)	This function allows you to add image filtering to the tool. The preprocessing filter is optional. You should add a preprocessing filter before adjusting or locking the other parameters. A preprocessing filter will extend the runtime of the test. See chapter 7.3.4.
5	Advanced	These settings are optional. They allow more control over the object and the runtime. You can zoom in and out on areas and hide them. Hiding areas allows you to cut them from your search area by using a brush to turn these areas red. This function increases the test time.
6	Help	This opens the help window. Here you will find useful operational information.

Setting the Tolerance Values

For each tool, you can set tolerance values for your measurement results. Depending on the measurement result, the tool (Vision Tool) can assume the status "Pass", "Recycle", or "Fail." In the script, this result can be evaluated under the variable #VisionToolVariable.Result.

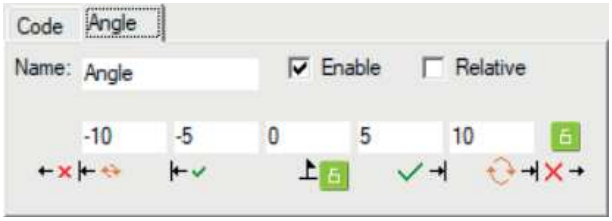
The following tolerance settings are available, depending on the particular tool:

Input Fields

Symbol	Designation	Function
	Minimum Recycle	The minimum recycle tolerance represents the smallest extended measurement tolerance. This value is located between the smallest "Recycle" value and the smallest value at which the measurement is passed (minimum pass). The "Recycle" value is reused in the next measurement. Note: The recycle tolerance is enabled or disabled in the "Application Settings" menu in the Nexus application (see "Changing Application Settings" on page 37).
	Minimum Pass	Smallest value to pass the measurement.
	Perfect	This exact value needs to be measured to pass the measurement.
	Maximum Pass	Largest value to pass the measurement.
	Maximum Recycle	The maximum recycle tolerance represents the largest extended measurement tolerance. This value is located between the largest "Recycle" value and the largest value at which the measurement is passed (maximum pass). The "Recycle" value is reused in the next measurement. Note: The recycle tolerance is enabled or disabled in the "Application Settings" menu in the Nexus application (see "Changing Application Settings" on page 37).
	Fail or Reject too small	Measurement failed, the measured value is too small.
	Fail or Reject too big	Measurement failed, the measured value is too large.
	Locking	The padlock is used to lock or unlock values so that they do not change when you move or resize your tool's search field or customize the tool properties. A red padlock indicates that the values are locked. A green padlock indicates that the values are not locked. Some measurements are locked by default when you apply a vision tool.

Example

For example, a requirement may contain information indicating that the expected measured value is 100 ± 5 and that a test object can be recycled or post-processed if the measured value is 100 ± 7.



2021-07

PEPPERL+FUCHS

Editing the 2-D Code Tool

1. Right-click on an edge of the rectangular search area.

   ↳ This opens the editing field for the 2-D code tool.

2. Click the "Preprocess" button (see chapter 7.3.4) to add an image filter to the tool.

   Note

   The preprocessing filter is optional. You should add a preprocessing filter before adjusting or locking the other parameters. A preprocessing filter will extend the runtime of the test.

3. Click the "padlock" button next to the value to store a fixed value.
4. Click "OK".

   ↳ The changes are applied and the editing field closes.

5. Click "Cancel".

   ↳ The editing field closes without any changes being made.

6. Click "Apply".

   ↳ The changes are applied without closing the editing field.

7. Click "Help".

   ↳ This opens the help window. Here you will find useful operational information.

Advanced Settings

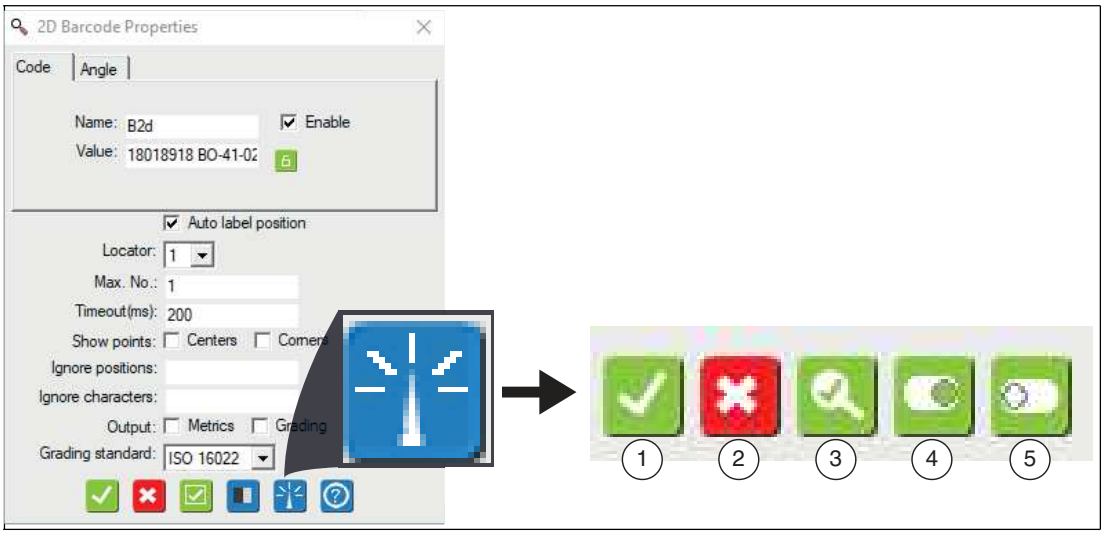


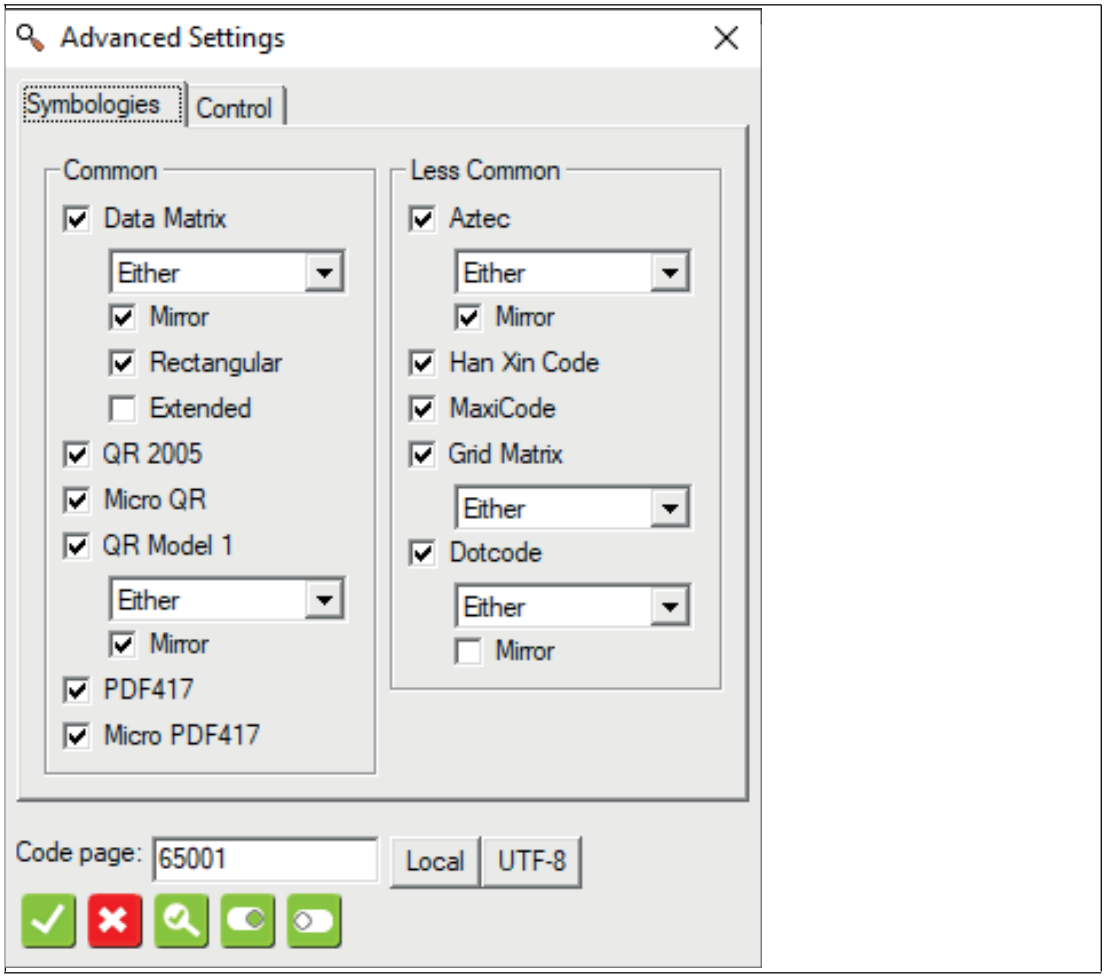
Figure 7.13 Advanced settings in the menu bar

The buttons are the same for all tabs

Position	Button	Description
1	OK	Changes are applied and the editing field is closed.
2	Cancel	The editing window closes without any changes being made.
3	Detect and set symbology	The code type is automatically identified and selected in the search field.
4	Enable all symbologies	All code types or symbologies are enabled on both tabs.
5	Disable all symbologies	All code types are disabled on both tabs.

Tab Description

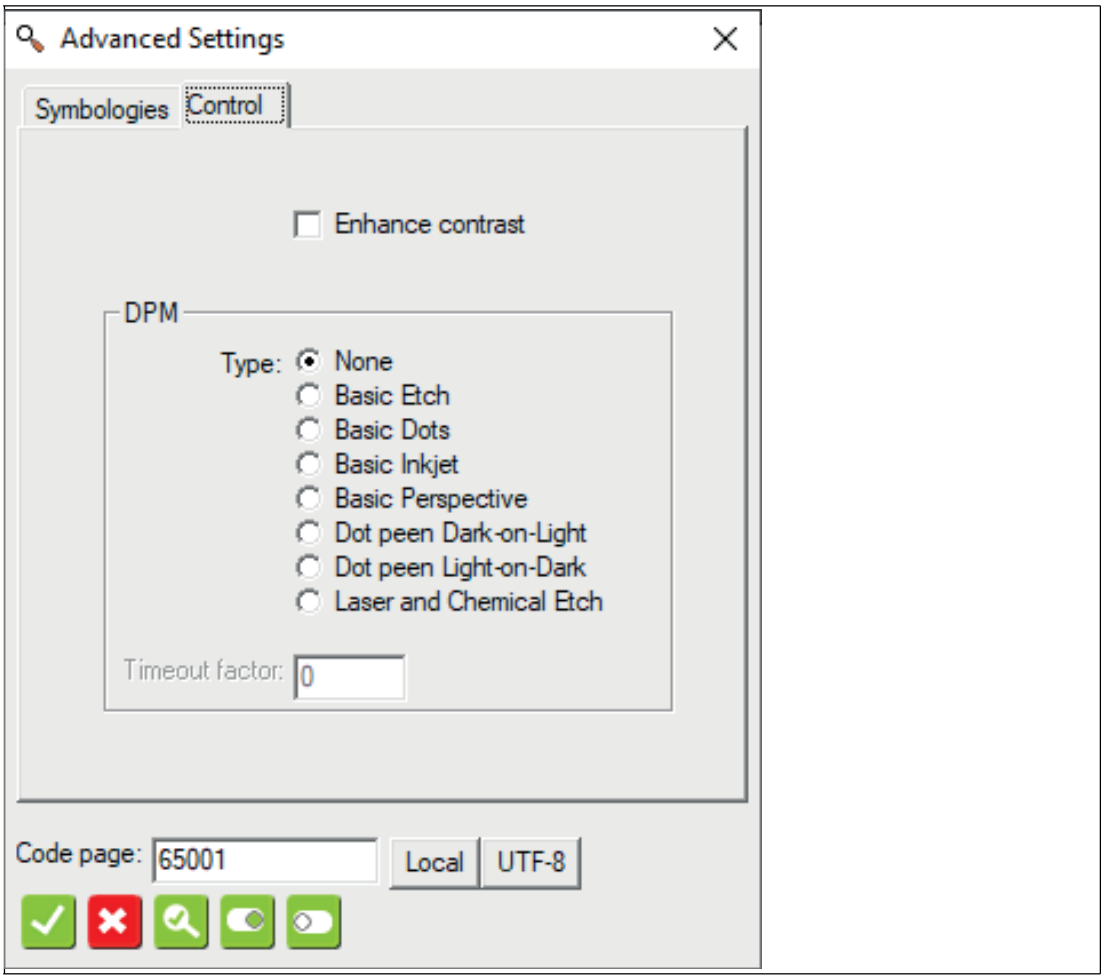
Symbologies



Menu	Description
Data Matrix	Enables decoding of the Data Matrix codes. The polarity options from the drop-down list are Dark-on-Light (dark codes on a light background), Light-on-Dark (light codes on a dark background), or allowing both. • **Mirror**: Enables decoding of both normal and mirrored images of codes. If this box is not checked, only normal images will be decoded. • **Rectangular**: Enables decoding of standard rectangular codes. • **Extended**: Enables decoding of extended rectangular codes.
QR 2005	Enables decoding of the QR 2005 code.
Micro QR	Enables decoding of the QR Model 1 code.
QR Model 1	Enables decoding of the Data Matrix codes. The polarity options from the drop-down list are Dark-on-Light (dark codes on a light background), Light-on-Dark (light codes on a dark background), or allowing both. • **Mirror**: Enables decoding of both normal and mirrored images of codes. If this box is not checked, only normal images will be decoded.
PDF417	Enables decoding of the PDF417 code.
Micro PDF417	Enables decoding of the Micro PDF417 code.
Aztec	Enables decoding of the Aztec code. • **Mirror**: Enables decoding of both normal and mirrored images of codes. If this box is not checked, only normal images will be decoded.

Menu	Description
Han Xin code	Enables decoding of the Han Xin code.
Maxi code	Enables decoding of the Maxi code.
Grid Matrix	Enables decoding of the Grid Matrix code. The polarity options from the drop-down list are Dark-on-Light (dark codes on a light background), Light-on-Dark (light codes on a dark background), or allowing both.
Dot code	Enables decoding of the Dot code. The polarity options from the drop-down list are Dark-on-Light (dark codes on a light background), Light-on-Dark (light codes on a dark background), or allowing both. • **Mirror**: Enables decoding of both normal and mirrored images of codes. If this box is not checked, only normal images will be decoded.
Code page	Displays the character set table for character decoding. Unicode is widely used, but some codes still contain older characters in the character set table. The two buttons indicate the most common values. You can enter a different character set number in this field. • **Local** sets the character set table to the same local language setting of the PC. • **UTF-8** sets the character set table to the Unicode standard (Unicode Transformation Format 8).

Control



2021-07

PEPPERL+FUCHS

Menu	Description
Enhance contrast	Allows the image to be improved in order to decode poorly printed codes. This may lengthen the execution time. Cannot be used with Dot peen Dark-on-Light, Dot peen Light-on-Dark, or Laser and Chemical Etch.
DPM Type	Selects image conditioning options specifically for direct part marking (DPM). Note: For all DPM options, the maximum search field size is 614,000 pixels. • **None**: Does not allow DPM-specific processing. • **Basic Etch**: Optimized for simple laser or chemical etching. • **Basic Dots**: Point-jet and inkjet images • **Basic Inkjet**: Basic inkjet images and poor-quality inkjet images. • **Basic Perspective**: Reads laser markings and normal codes of good quality that are centered in the image but have strong perspective distortion. This mode is faster but less robust than "Basic Etch" for reading poor-quality codes. • **Dot peen Dark-on-Light**: This is the most robust method for reading dark dot peen codes on a light background. • **Dot peen Light-on-Dark**: This is the most robust method for reading light dot peen codes on a dark background. • **Laser and Chemical Etch**: This is the most robust method for reading laser or chemical etching point codings. **Timeout factor**: The timeout factor sets a timeout for difficult, deteriorated, or non-existent DPM codes. There is no timeout if this field = 0. The actual timeout for the device is ten times the number entered here in milliseconds. Note: This timeout does not apply to None, Basic Etch, or Basic Dots. This timeout applies only to Basic Inkjet, Basic Perspective, Dot peen Dark-on-Light, Dot peen Light-on-Dark, and Laser and Chemical Etch.
Code page	Displays the character set table for character decoding. Unicode is widely used, but some codes still contain older characters in the character set table. The two buttons indicate the most common values. You can enter a different character set number in this field. • **Local** sets the character set table to the same local language setting of the PC. • **UTF-8** sets the character set table to the Unicode standard (Unicode Transformation Format 8).

7.3.3 Creating a Locator Using Count Tool

The 1-D and 2-D code tools are suitable tools for finding codes in all orientations and movements. However, if the codes in the image move in a larger area, you would have to select a very large search field. The resulting larger search field would increase the processing time proportionally and therefore decrease the read rate. If you follow the search field using a locator, you can allow a smaller search field, thereby increasing the read rate per second.

The Count Tool searches for light objects on a dark background or dark objects on a light background. The tool can locate the center and corner points of areas used to create the locator. You can match the center of an object to the "blob" (= a group of adjacent contiguous pixels of the same or similar brightness or color value that have a closed contour) and use it as the locator point. A locator alone is sufficient to track horizontal and vertical movements, but it cannot track rotation. You need two locators to track the rotation.

The following describes how to create a locator using the Count Tool.

Creating Position and Rotation Locators

1. Click the "Count Tool" button in the tool selection area.

   ↳ Move the mouse over the image in the work area.

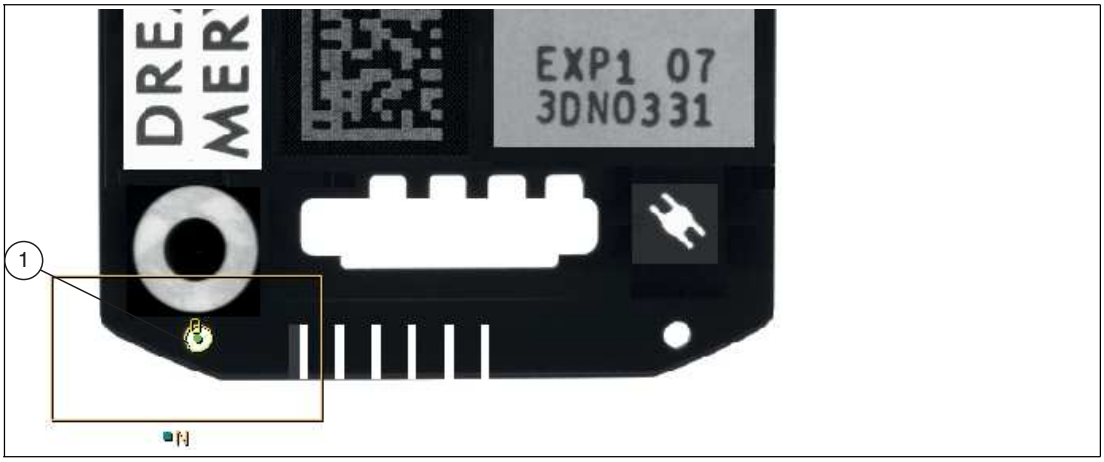


Figure 7.14 First locator

2. Draw a search field around the first locator (1). Set the search area to the largest expected movement range of the locator.

   ↳ The selected field is labeled "N." Inside the search area, the area found is outlined in yellow and the center point is marked with a green dot.

3. Right-click on an edge of the rectangular search field.

   ↳ The Count Tool editing window will open.

4. Next to the "Show points" function, select "center."

5. Enable the "Reject touch" function.

   ↳ Only "blobs" (included areas) are counted.

6. Select the appropriate background under "Object type." "Bright" has been chosen in this example.

Note

Bright means that light objects on a dark background are identified and counted.

2021-07

PEPPERL+FUCHS

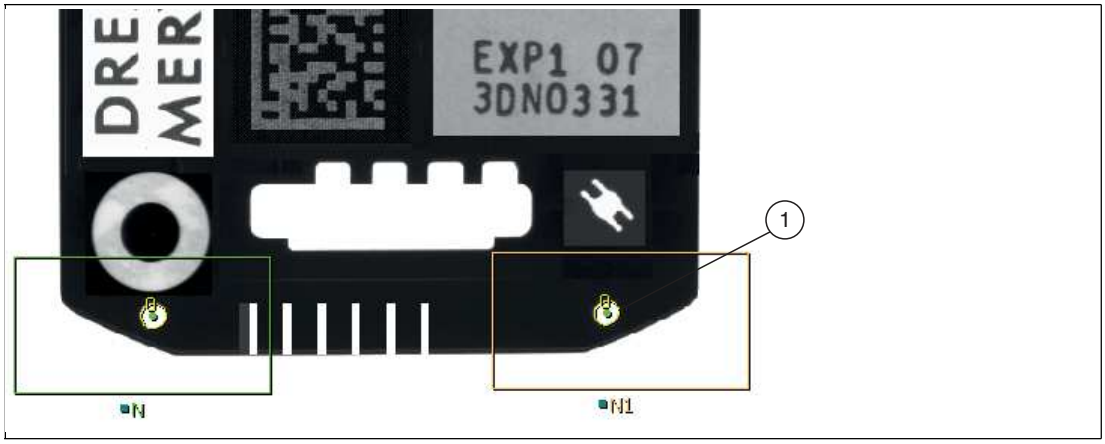


Figure 7.15 Second locator

7. Create a second locator (1) in the image area. Set the search area to the largest expected movement range of the locator.

Tip

You can copy the first Count Tool.

↳ The selected field is labeled "N1." Inside the search area, the area found is outlined in yellow and the center point is marked with a green dot.

8. Right-click an edge of the rectangular search field of the second locator.

↳ The Count Tool editing window will open.

9. Next to the "Show points" function, select "center."

10. Enable the "Reject touch" function.

↳ Only "blobs" (included areas) are counted.

11. Select the appropriate background under "Object type." "Bright" has been chosen in this example.

Note

Bright means that light objects on a dark background are identified and counted.

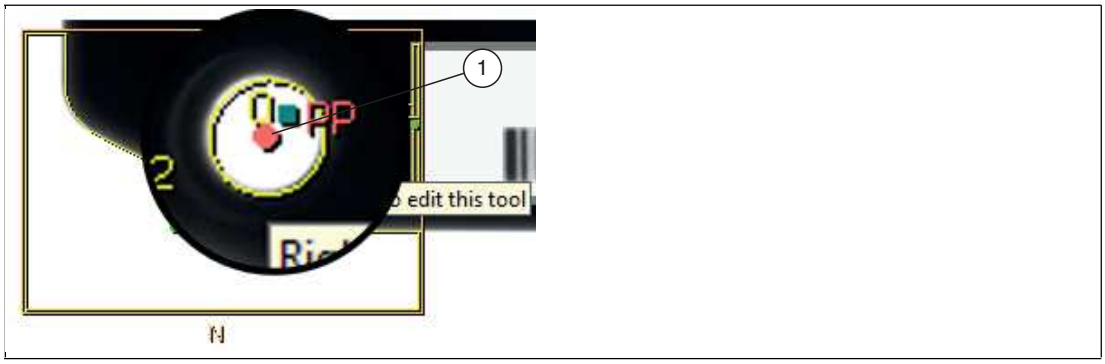


Figure 7.16 Search area center

12. Move the mouse over the center "PP" (1) of the **first** search area. Right-click the point when it turns red.

↳ The properties window for the "Point Properties" point will open. In the "Act as Locator" area, you can define the position locator.

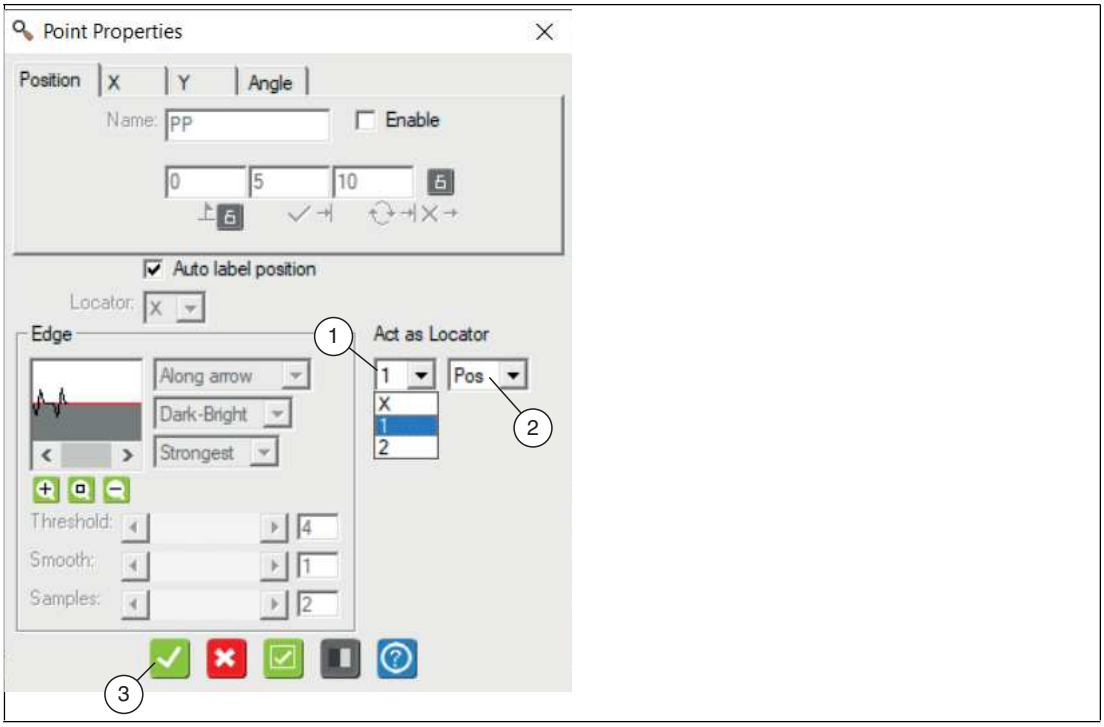


Figure 7.17 Point Properties—Select Position Locator

13. In the "Act as Locator" section of the Point Properties window, select "1" from the first drop-down list (1) and "Pos" from the second drop-down list (2). Click "Apply" (3).

Figure 7.18 Position and rotation locators

1 Crosshair = position locator

2 Flag symbol = rotation locator

↳ The properties window will close. A crosshair (1) is displayed on the point in the image area. This indicates that the point is defined as a position locator.

Note

The "Position" locator alone can track horizontal and vertical movements but not rotations. The position locator alone may be sufficient if your part is mechanically limited so that it does not rotate, e.g., a square or rectangular part that slides between two close-fitting rails. However, if your part rotates—even slightly—you will need a rotation locator.

2021-07

PEPPERL+FUCHS

Tip

To obtain the best results, the two locator points (for position and rotation) should be as far apart as possible and must not leave the camera's field of view.

14. Move the mouse over the center of the **second** search area. Right-click the point when it turns red.

    ↳ The properties window for the "Point Properties" point will open. In the "Act as Locator" area, you can define the rotation locator.

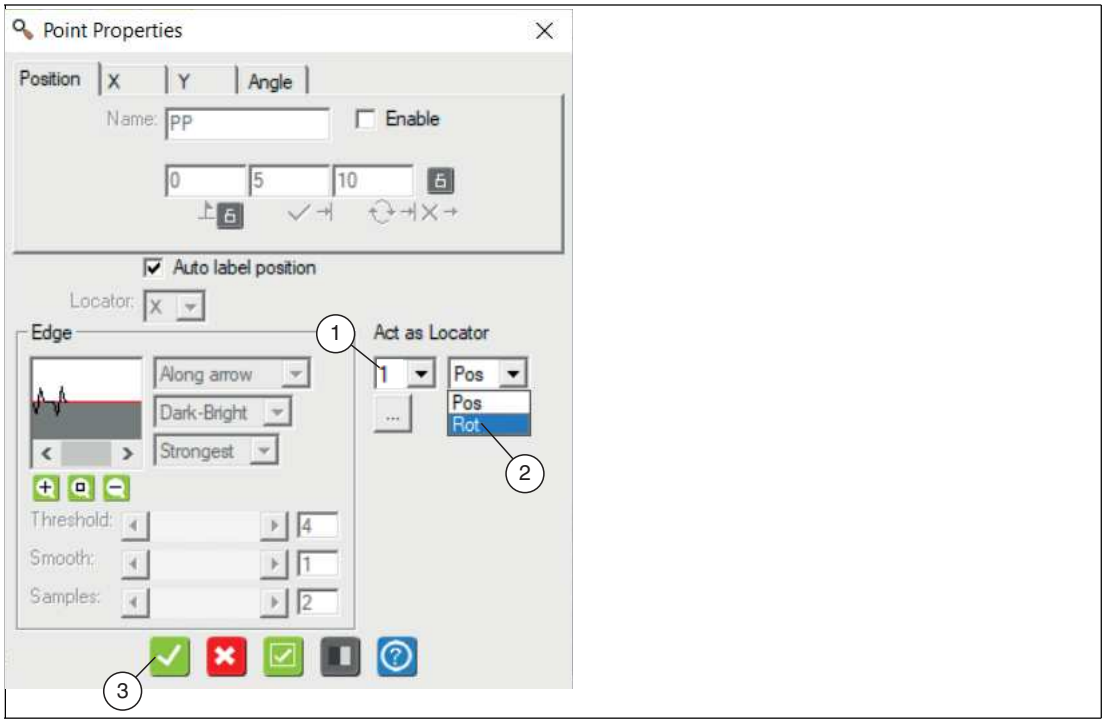


Figure 7.19 Point Properties—Select Rotation Locator

15. In the "Act as Locator" section of the Point Properties window, select "1" from the first drop-down list (1) and "Red" from the second drop-down list (2). Click "Apply" (3).

    ↳ The properties window will close. In the image area, a flag symbol is displayed on the point. This indicates that the point is defined as a rotation locator.

16. Go to the "Run solution" operating panel (see chapter 7.6) to test your locator. Check whether the locator follows the part or whether you need to make adjustments.

7.3.4 Preprocessing Filter (Preprocess)

To improve specific properties of an image, preprocessing filters can be used before a tool is used. Codes that are difficult to read can be simplified by taking one or more preprocessing steps.

Use the selection lists to select up to five different preprocessing filters for the selected search field. Multiple preprocessing filters are applied in sequence, from left to right. Some preprocessing filters have parameters that you can change. The fields below the selection list have names and default values. The fields remain blank if there are no modifiable parameters.

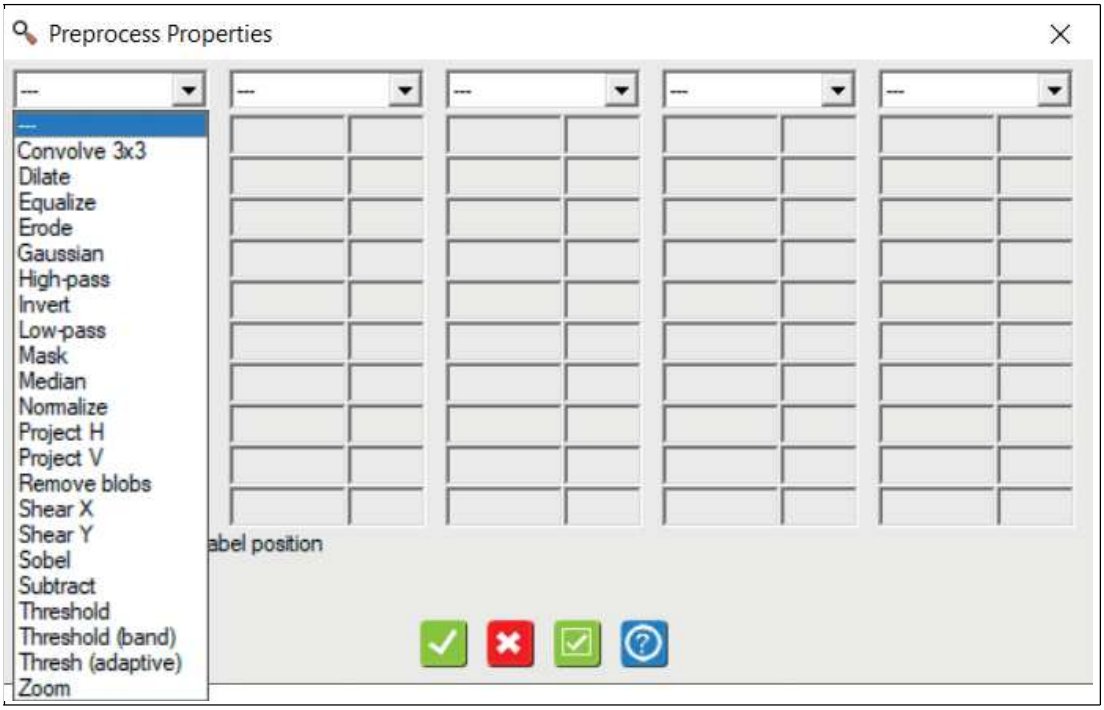


Figure 7.20 Screenshot—Preprocessing filter

Note

Avoid excessive use of preprocessing filters. First try adjusting the image using the lighting and the optics (lens). Preprocessing modifies the original image data. It also extends the test time.

Note

The VOS-I sensor contains all preprocessing filters of the VOS universal vision sensor. This section describes only the preprocessing filters relevant to the VOS-I.

Dilate Function

The Dilate operator expands the lighter pixels. It is useful for combining small elements that are lighter than their surroundings.

Figure 7.21 Dilate function

1. Original image
2. Dilate Preprocessing Filter

Equalize

This function extends the image intensity to cover the entire range from 0 to 255. The darkest pixels in the area are set to 0. The lightest pixels in the area are set to 255. Other pixels are set based on statistical standardization of the intensity histogram.

Erode

Erode is the opposite of the Dilate function. It expands the darker pixels. It is useful for resolving features that are connected by a relatively thin element.

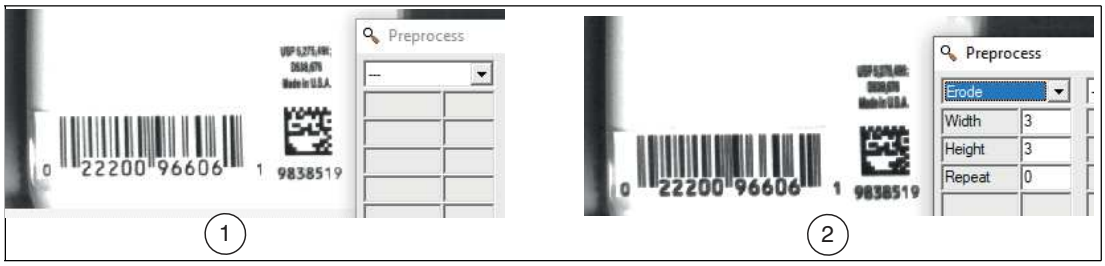


Figure 7.22 Eroding

1. Original image
2. Eroded image

After an eroding operation, a dilation operation can be performed to obtain the size of the features:

Median

The median setting replaces a pixel with the median value of its neighbors. The median setting is similar to the low-pass function in that it smoothes an image. The median setting is better suited for removing peak noise than the low-pass function.

Original image

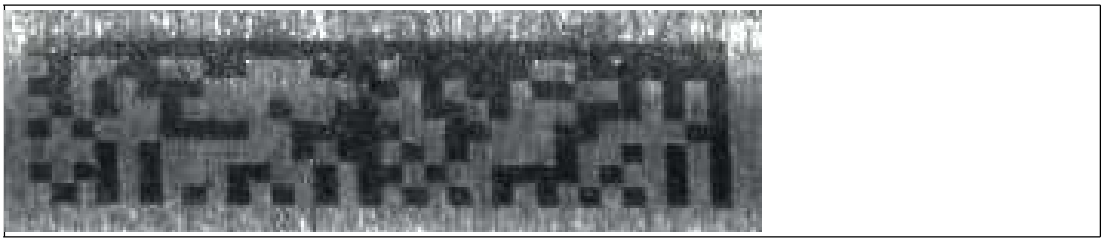


Figure 7.23 The original image

Low pass

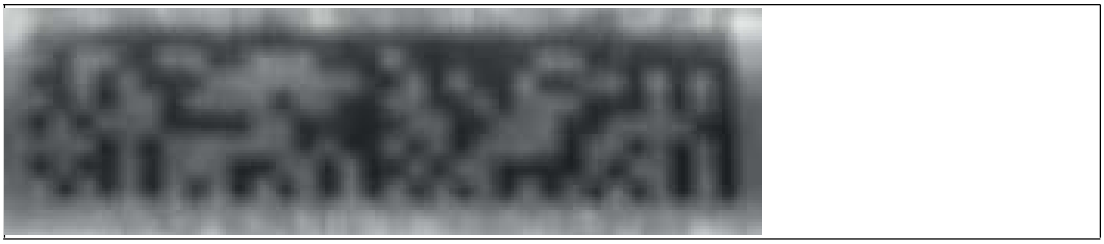


Figure 7.24 Low pass (the background texture is suppressed, but the edges become blurred)

Median

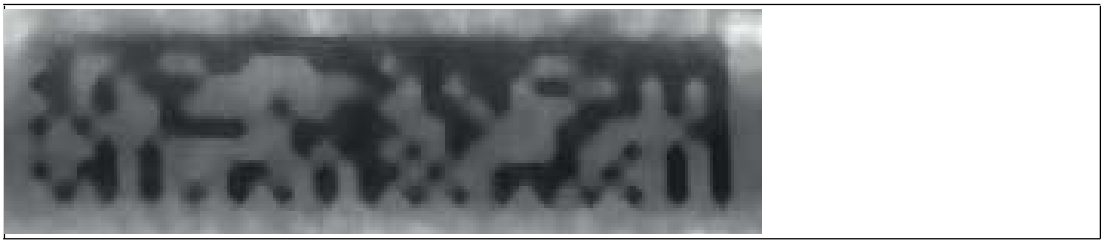


Figure 7.25 Median (the edges are sharper)

Normalize

This function is used to increase the contrast such as black-and-white contrasts of codes.

Figure 7.26 Normalizing

1. Original image
2. Edited image

Threshold

The Threshold filter converts a grayscale image to a digital image. Pixels above a threshold value are shown as white and pixels below the threshold value are shown as black.

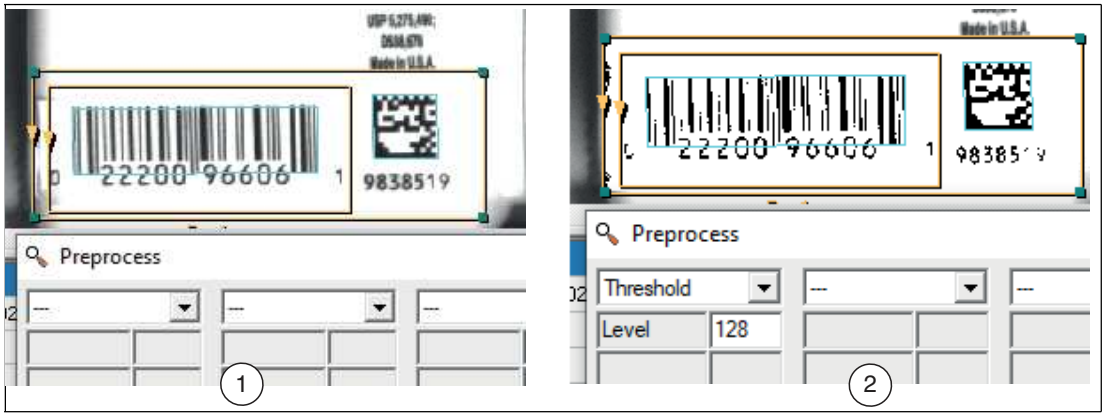


Figure 7.27 Converting a grayscale image into a digital image

1. Original image
2. Threshold image (weak deviations are removed)

Example

Combining filters

In this example, a Median filter was applied first and then a Dilate filter. In addition, the "Dot peen Light-on-Dark" function has been enabled under the advanced settings.

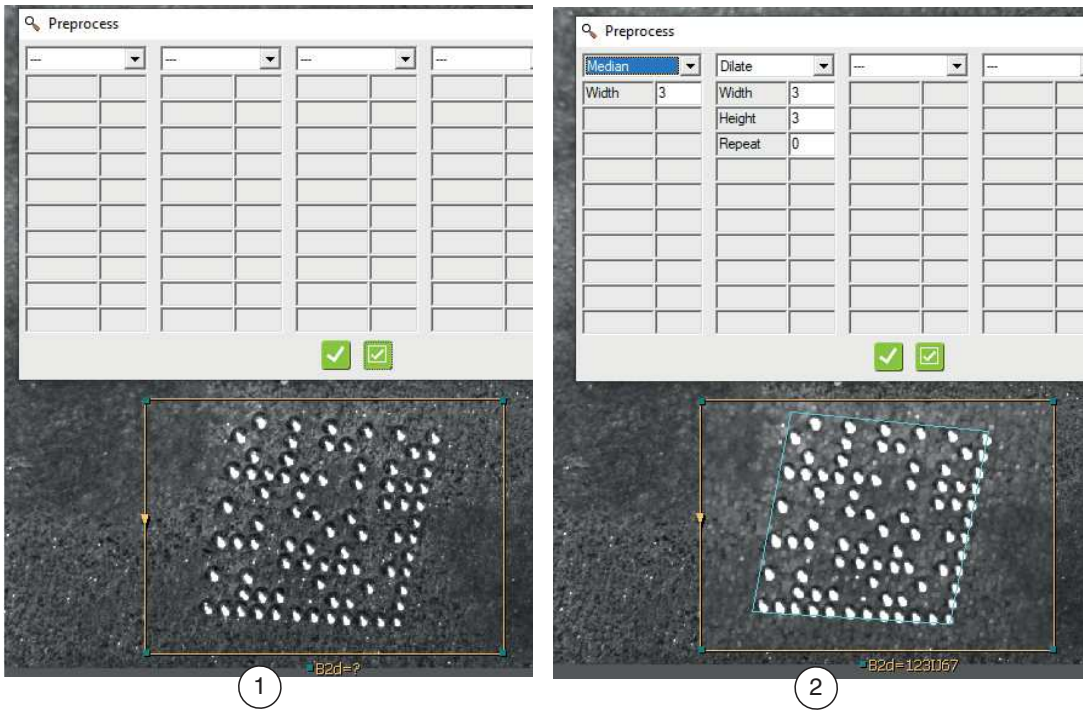


Figure 7.28 Combining filters

1. Original image
2. Median and Dilate preprocessing filters

7.4 Interface Configuration

This section describes how to configure the sensor to communicate with other devices.

Overview of Interface Menu

The interface menu is the place to set up the connection to external devices (e.g., PLC or storage devices) for control/status communication.

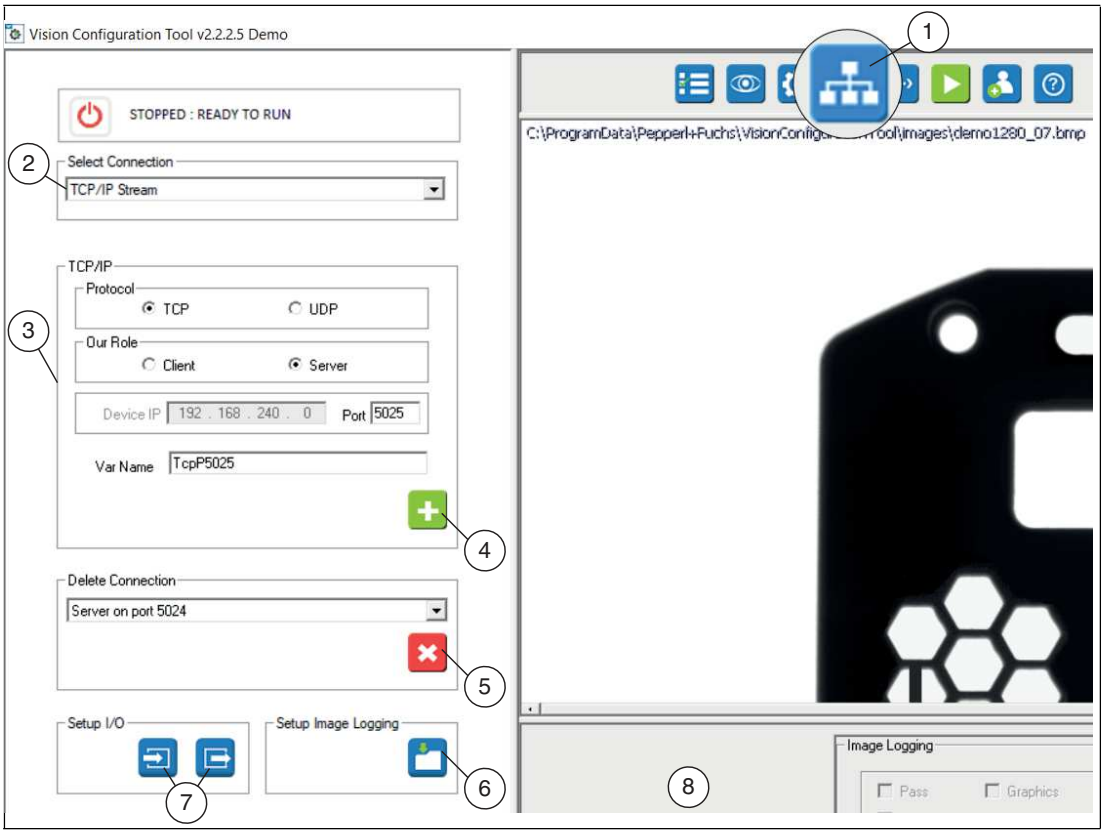


Figure 7.29 Example of interface menu for TCP/IP Stream

Item	Menu	Description
1	Set up communication	The "Set up communication" button takes you to the main menu for interface configuration.
2	Select connection method	Use the drop-down window to select your interface type.
3	Set up connection method	Use this section of the menu to set up the connection of the selected interface type. The menu changes depending on the PLC type selected. Various functions are available for selection.
4	Add configuration	Add new configuration
5	Delete configuration	Delete the selected configuration
6	Set up image logging	Enable/disable history log and image logging
7	Configuration I/O	You can use these two buttons to set up the inputs and/or outputs.
8	Configuration and status window	The settings of the selected menu are displayed in the configuration and status window.

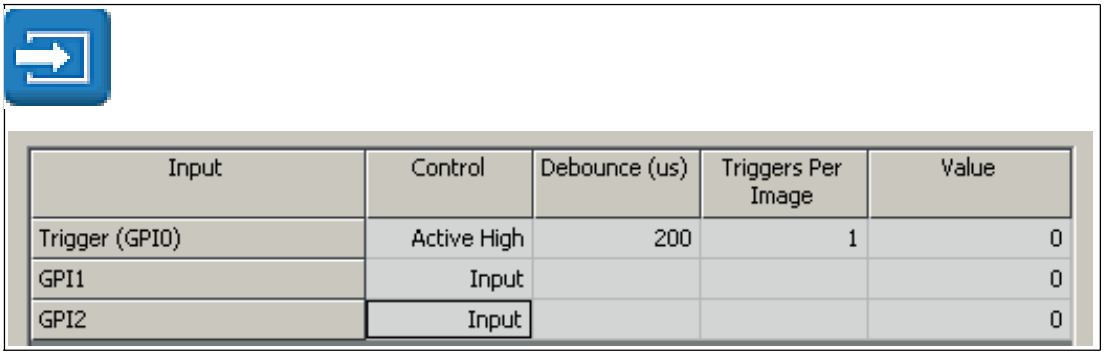
2021-07

Sensor Name and Address

Sensors are supplied with a default name "VOSxxxx" and an address (192.168.0.100). If you are connecting more than one sensor in the same network or on the same PLC, change the IP address so that each sensor has its own address. You should also change the name so that each sensor has its own name. This is important and even critical for some PLCs and network configurations to avoid confusion. Use the Nexus application to change the name and address of the sensor (see "Changing the IP address" on page 35).

7.4.1 Inputs

When you click the "Inputs" button, the settings menu for inputs opens in the configuration and status window. In this menu, you can adjust the following settings:



Input	Control	Debounce (us)	Triggers Per Image	Value
Trigger (GPI0)	Active High	200	1	0
GPI1	Input			0
GPI2	Input			0

Figure 7.30 Input menu

Function	Description
Trigger (GPI0)	You can set the polarity (Active High or Active Low) in the "Control" column, enter a debounce period, and set a trigger divider "Triggers Per Image" for the sensor trigger input.
GPI1 & GPI2	You can use a drop-down list to select the function of the two remaining inputs. The "Value" column shows the current status of all inputs (1 for high or 0 for low).

Example of Job Change via Inputs

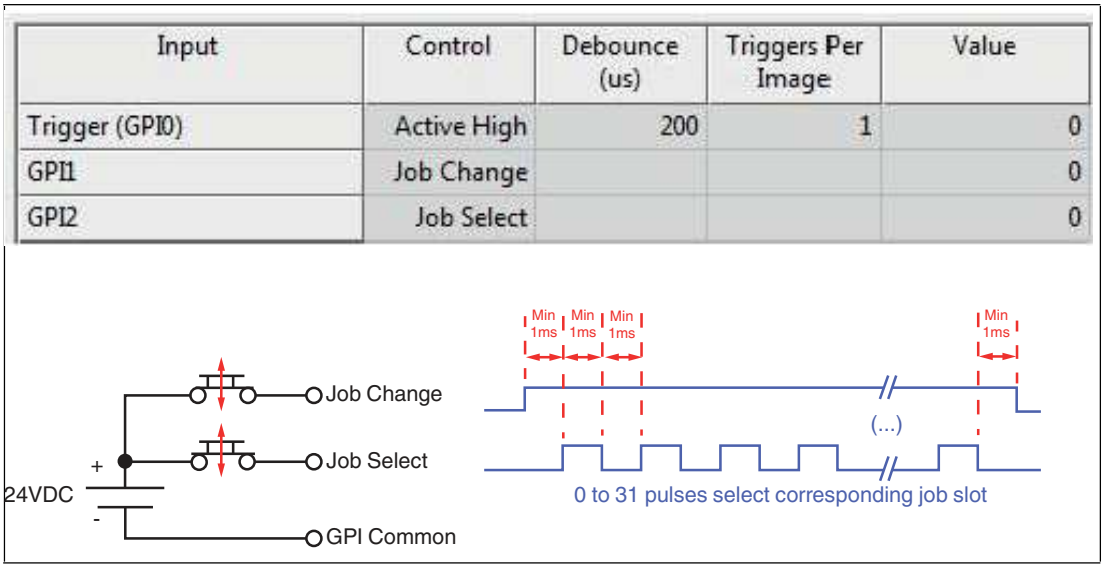


Figure 7.31 Sample input menu

You can save up to 32 job files to your VOS device that can be loaded as and when needed. The job files can be changed over in different ways. Via the "Select Solution" menu selection, via the "ChangeSolution(#)" script function, or via the inputs as described below.

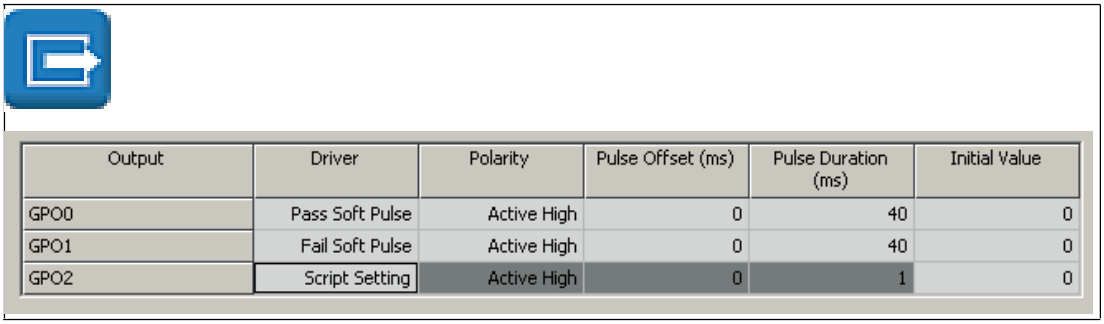
To use the sensor inputs for the job change, select the "Job Change" and "Job Select" functions in the configuration window of the input control panel as shown in the figure.

2021-07

A PLC sets a HIGH signal on the "Job Change" line GPI1 and sends a series of pulses on the "Job Select" line GPI2 corresponding to the job number; e.g., 12 pulses for job file 12. The "Job Select" pulse sequence requires a setup and delay time of one millisecond between the edge transitions. If a job change is detected, the sensor cancels the current test process (if it is running) and switches over immediately.

7.4.2 Outputs

When you click the "Outputs" button, the settings menu for outputs opens in the configuration and status window. In this menu, you can adjust the following settings:



Output	Driver	Polarity	Pulse Offset (ms)	Pulse Duration (ms)	Initial Value
GPO0	Pass Soft Pulse	Active High	0	40	0
GPO1	Fail Soft Pulse	Active High	0	40	0
GPO2	Script Setting	Active High	0	1	0

Figure 7.32 Output menu

Function	Description	
Driver	Use the drop-down list to select what is used to control the output:	
	Script Setting	The output is controlled by equations or instructions in the script. This value prevents the other fields in the row from being displayed.
	Pass Pulse	An "active pulse" is output if the result is "Pass."
	Recycle Pulse	An "active pulse" is output if the result is "Recycle."
	Fail Pulse	An "active pulse" is output if the result is "Fail." Skipped parts are counted as "Fail."
	NOTE: The "Pass" pulse, the "Recycle" pulse, and the "Fail" pulse are hardware-controlled pulses. The offset and the clock start with the image capture trigger. This is a "deterministic" time measure. If a decision result is not available when the pulse shift time is reached, a "Fail" pulse is output by default. Minimum pulse offset = exposure time + capture time + inspection time Runtime overrange (e.g., displaying or saving images) can also affect the request for the minimum pulse offset. You should investigate or calibrate this time based on your specific system usage and use of other devices in the same network.	
	Strobe Pulse	A pulse is output after a trigger input. The pulse uses the settings from the sensor settings menu
	Pass Soft Pulse	An active (polarity value) pulse occurs when the result of the job is "Pass." This is a software-controlled output. The offset and time begin with the availability of the decision result. This is a "non-deterministic" time sequence.
	Fail Soft Pulse	An active (polarity value) pulse occurs when the result of the job is "Fail." This is a software-controlled output. The offset and time begin with the availability of the decision result. This is a "non-deterministic" time sequence. Skipped parts are also counted as "Fail."
Polarity	Use the drop-down list to select "Active High" or "Active Low" for the output pulse.	
Pulse Offset	Enter a value for the output pulse delay to synchronize it with external devices.	
Pulse Duration	Enter a value for the output pulse width.	
Initial Value	Enter a value for the state or value of the output when the job is loaded.	

2021-07

7.4.3 Image Logging

When you click "Setup Image Logging," the settings menu for image logging opens in the configuration and status window. You can use the "Image Logging" function to store a limited number of images and data in the sensor memory. The "Image File Logging" function can be used to save sensor images to the PC.

Setting the History and Image Logging

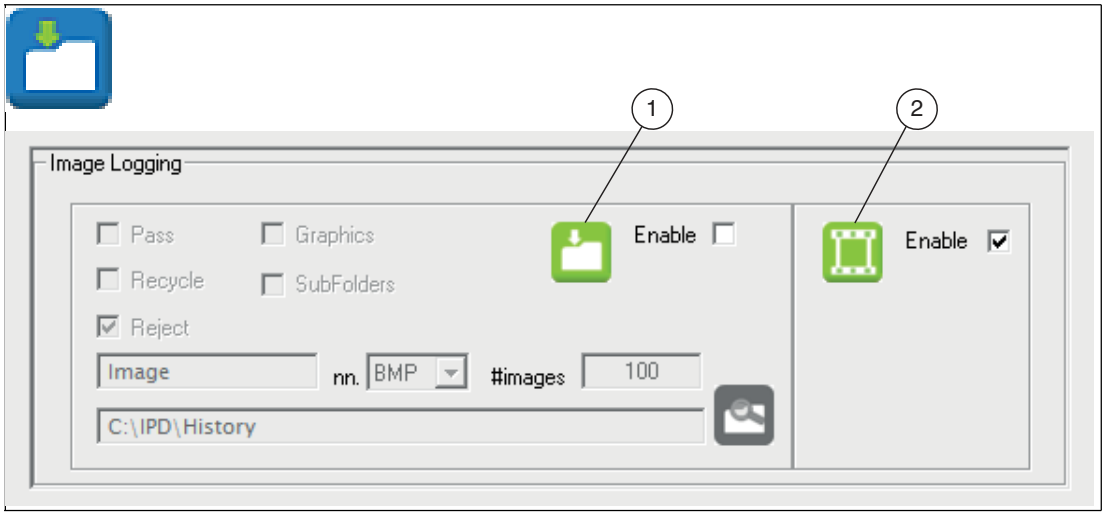


Figure 7.33 Image Logging menu

Saving Sensor Images to the PC (Image File Logging)

The number field (#images) is the maximum number of images. Images are stored under file names with consecutive numbers (image0.bmp, image1.bmp). The counter continues to run on this number and will overwrite older images until you disable image logging.

1. Select the "Enable" check box (1).

Note

To stop saving images, return to this operating panel, clear the check box next to "Enable" (1), and close the operating panel.

2. Enter the maximum number of images to be saved in "#images."
3. Choose a category: "Pass", "Recycle", or "Reject."
4. If you select more than one category, select "SubFolders" to create subdirectories for "Pass", "Recycle", and "Reject."
5. Check the "Graphics" check box to save measurement graphics. If the box is cleared, only camera images will be saved.
6. Change the file name if necessary (the default name is "Image").
7. Select the image format: BMP or JPG.
8. Specify the target directory.

Saving Images and Results in the Sensor

Select the "Enable" check box (2) for the "Image Logging" function to save images and/or data in the sensor.

Note

There are three categories: Pass, Recycle, and Fail. Each category saves the last 20 images at a resolution of 640 x 480 pixels, two or four images at a resolution of 1280 x 960 pixels. Clear the "Enable" check box to prevent images or data from being saved in the sensor. This will shorten the test periods. The history log appears in the "Run solution" window when you click "History Recall" in the execution section (see chapter 7.6).

Image Logging Via FTP

The sensor supports image logging via FTP as required. An FTP server pointing to a shared folder on the host must be enabled for this function to work.

Below you will find a description on how to set up image logging with the sensor:

Setting Up Image Logging

1. Start the FTP server on the host.
2. Use the "LU" command to send the user name, password, and host name of the FTP server. The format is username:password@host, i.e., "LUvos:vospwd@GV400."

   ↳ The FTP server specifies the folder on the host where the images are stored.
3. Use the "LI" command to send the file image names you want to save on the host, e.g., "LIMyimage."
4. Use the "LE" command to enable the image logging function on the sensor, e.g., "LE1."

   ↳ Once enabled, the sensor logs images (each time they are taken) in BMP format in the host's shared folder. The files are stored in a ring buffer with a ten-image capacity, meaning that the eleventh image will overwrite the first image.

Example

Sample contents of the shared host folder:

Myimage1.bmp

Myimage2.bmp

Myimage3.bmp

.

Myimage10.bmp

7.4.4 RS-232

The serial connection or "RS-232 stream" is used for general communication such as messages and data.

Figure 7.34 RS-232

RS-232 Example

You can create data strings using the Free Edit editor and the String Formatting editor (). You can use the following functions in script instructions to read and write on the serial interface:

PutPortString GetPortChar GetPortString WriteFormatString

The "PutPortString" function sends the exact string, without evaluating variables. The "WriteFormatString" function evaluates variables.

Serial I/O Functions

Function	Description
GetPortChar( )	Returns a new character input from the COM serial interface, if any, otherwise returns a value of zero.
GetPortString(endingChar)	Returns a new data string entered by the COM serial interface, if any, otherwise an empty data string is returned immediately. **endingChar**: Specifies the received character to indicate the end of a received character string. This character is not included in the returned character string.
PutPortString(string)	Sends a string to the COM serial interface. Does not perform evaluations for embedded variables.

7.4.5 TCP/IP Stream

The TCP/IP stream is used for general communication with devices (host computer), e.g., for outputting messages or data. Communication between the host computer and the sensor is as follows:

Inputs (host computer to sensor): control commands

Outputs (sensor to host computer): results, images via FTP

Communication from the sensor to the host computer takes place via scripts. A script contains a set of equations or instructions that affect variables to control events. The scripts are included in four functions on the sensor. Three of these functions are predefined and are run in the following order:

- **Solution Initialize**—This function is accessed immediately after a job is loaded. It is used to initialize the job variables to their expected state.
- **Pre Image Process**—This function is accessed immediately after receiving a new image, before processing begins. It can be used to handshake with other devices or to control external input/output.
- **Post Image Process**—This function is accessed immediately after processing and is usually used to formulate results and communicate with external devices. Example:
  - Send decoded strings (with or without symbology header)
  - Send result status (good, bad, match, no match, etc.)
  - Compare decoded strings with saved match strings

A fourth function is the periodic function (**periodic: 200 ms**). This function is accessed in a fixed time interval, which is preset in the sensor to 200 ms. The periodic function is used to query external inputs, such as serial RS-232, TCP/IP streams, or PLC tabs, to change the sensor properties or to trigger actions. Example:

- Setting the sensing modes such as Trigger mode, match string activation, string formatting, image logging, etc.
- Triggering a capture (when using software triggers)
- Modifying job files (when switching is enabled)
- Changing the sensor operating status (online/offline)
- Sending match strings to save in the local array
- Sending the FTP user name, password, host name, and file name of the image

Note

Each instruction (line) in a script is sent to the server (host) when a script is opened for editing. If scripts are long, there are delays in opening them. This is not noticeable on the emulator, since the client and server are on the same machine. This is not the case if you perform editing directly on the sensor. Reduce delays by reducing the number of lines in the script:

- Eliminate additional lines for comments (i.e., add comments to the same line as instructions).
- Optimize recurring instructions with arrays wherever possible.

Note

Windows Firewall should be switched off on systems that communicate via the network or should be set accordingly. See chapter 8.1.

The following example shows the basic run-through of image capture, tool evaluation, and subsequent further processing by the script. The "Post Image Process" is used in this example. In the script, it is important for the result variable—in this case B2d—to be written to TCP/IP port 5025 ("TCPP5025") for the read result of the 2-D tool. This port must also be entered for the TCP/IP connection. In this example, a substring is outputted. This can be displayed in the monitor program shown below if the specified TCP/IP port (i.e., in this case 5025) is selected again. How to use a monitor program is described in section (see "Displaying Interface Commands" on page 98).

2021-07

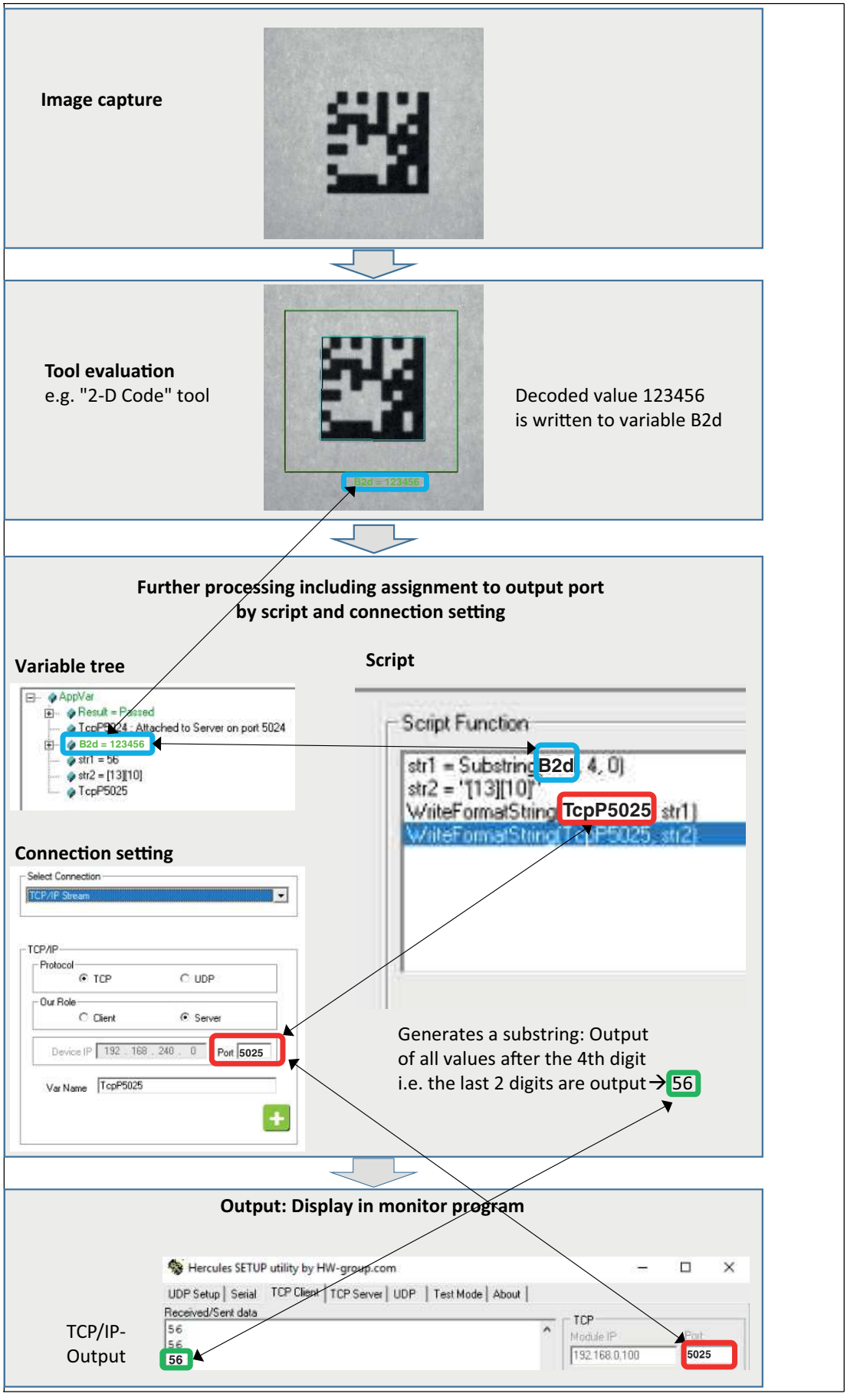


Image capture
Tool evaluation
e.g. "2-D Code" tool
B2d = 123456
Decoded value 123456
is written to variable B2d
Further processing including assignment to output port
by script and connection setting
Variable tree
AppVar
Result = Passed
B2d = 123456
str1 = 56
str2 = [13][10]
TcpP5025
Script
Script Function
B2d
TcpP5025
Connection setting
Select Connection
TCP/IP Stream
TCP/IP
Protocol
TCP
UDP
Our Role
Client
Server
Port 5025
Var Name TcpP5025
Generates a substring: Output
of all values after the 4th digit
i.e. the last 2 digits are output → 56
Output: Display in monitor program
TCP/IP-
Output
Hercules SETUP utility by HW-group.com
UDP Setup | Serial | TCP Client | TCP Server | UDP | Test Mode | About
Received/Sent data
56
TCP
192.168.0.100
5025

2021-07

Note

Simple and convenient TCP/IP communication is realized by the predefined jobs 00 to 05 on the sensor. To do this, one of these jobs must be selected. A description of the jobs can be found in section (see chapter 7.1).

The following interface example describes the individual settings in more detail.

TCP/IP Interface Example

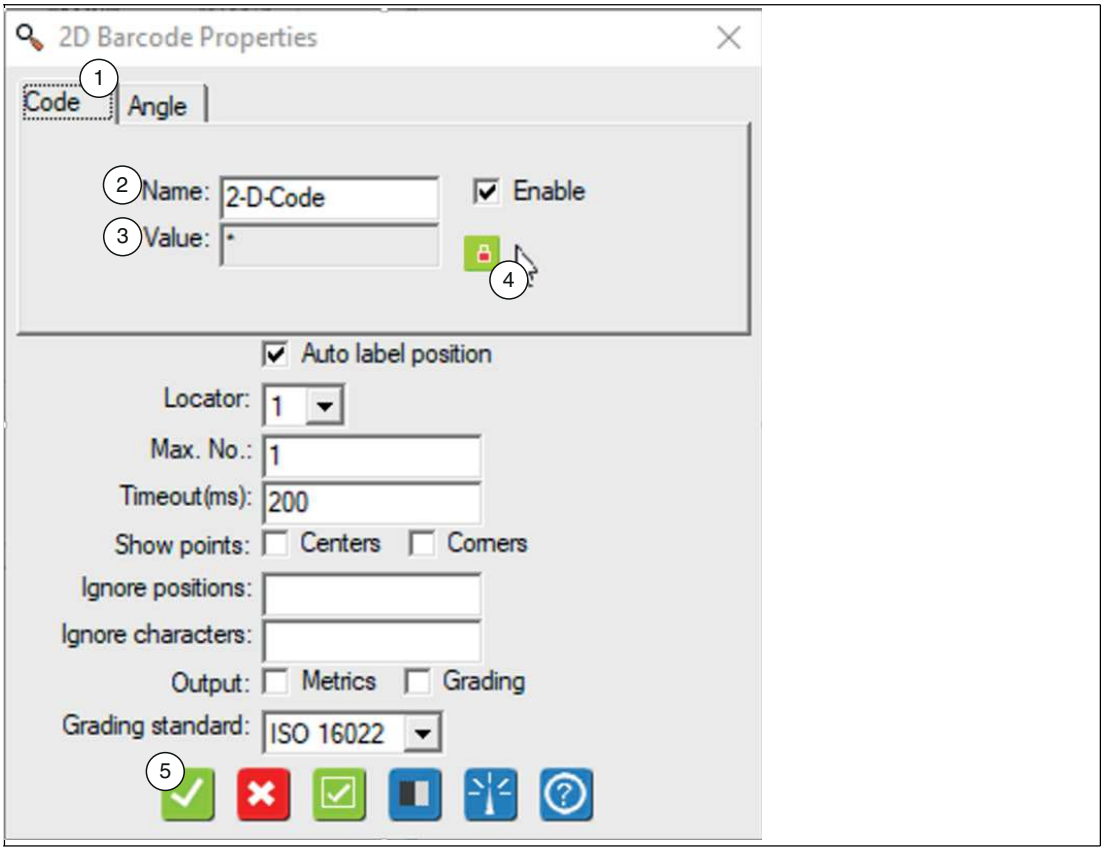


Figure 7.35 Properties window

1. Create a new job. See chapter 7.1.
2. Set the output variable "Name" (2) (example: 2-D code tool) in the properties window under the "Code" tab.
3. Place a "*" symbol as a placeholder for multi-digit characters in the input field of the output variable "Value" (3).
4. Click the padlock symbol (4) to lock the variable.
5. Click "OK" (5).

↳ The changes are applied and the properties window closes.

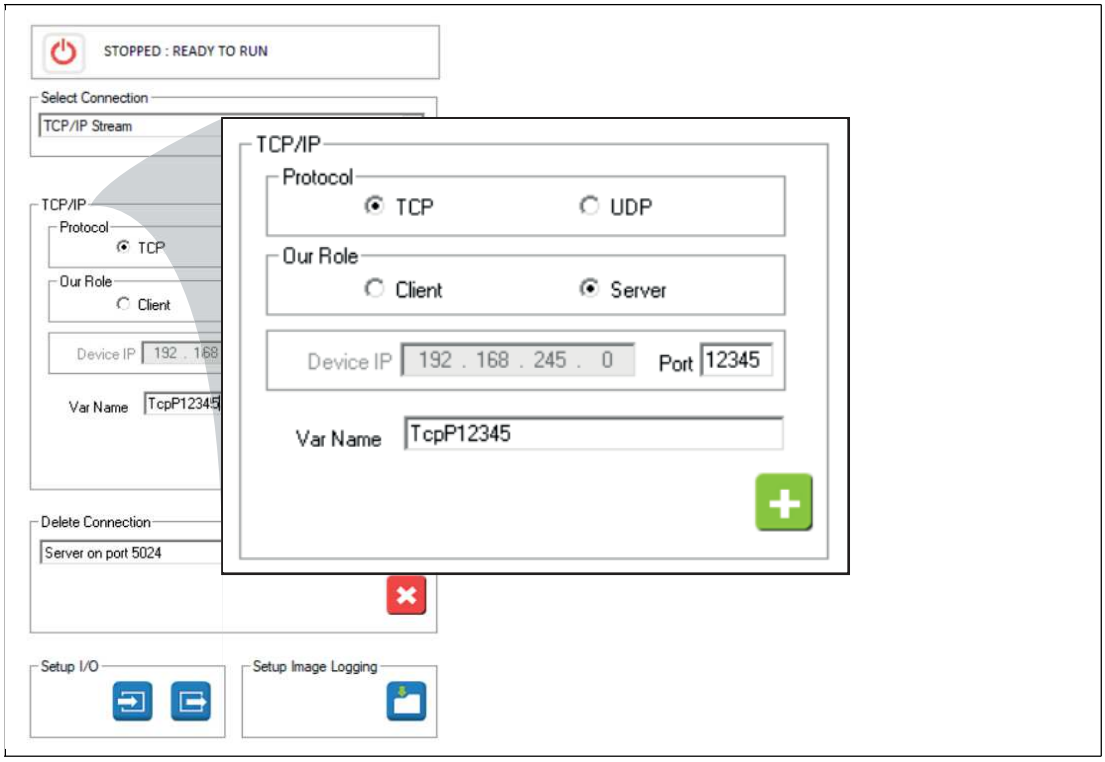


Figure 7.36 Interface parameters and a variable name

6. Set the interface parameters and a variable name for the interface. See table "Parameter Description" on page 98.

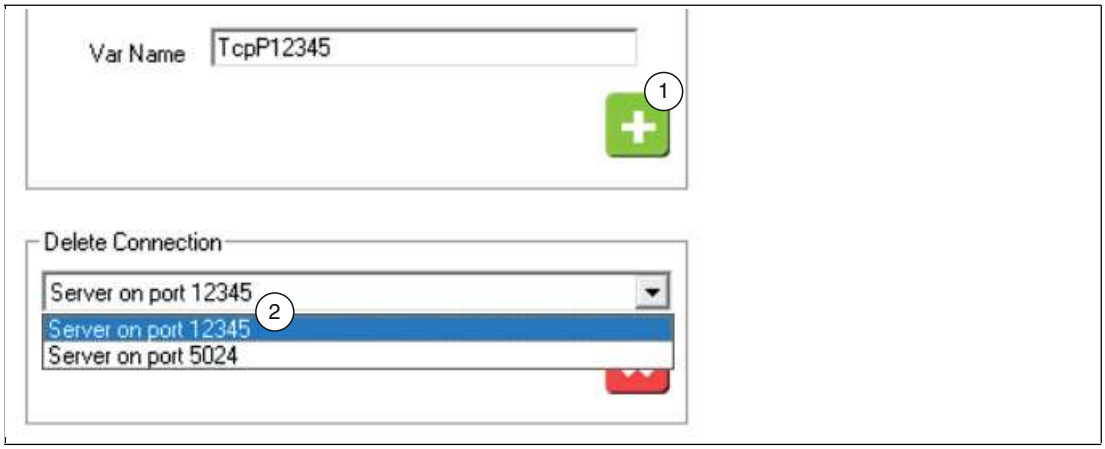


Figure 7.37 Variable Name

↳ The TCP/IP interface is available in the variable tree once it has been added using the "+" symbol (1).

7. To set a script to output the code via the TCP/IP interface, open the script editor () in the navigation bar.

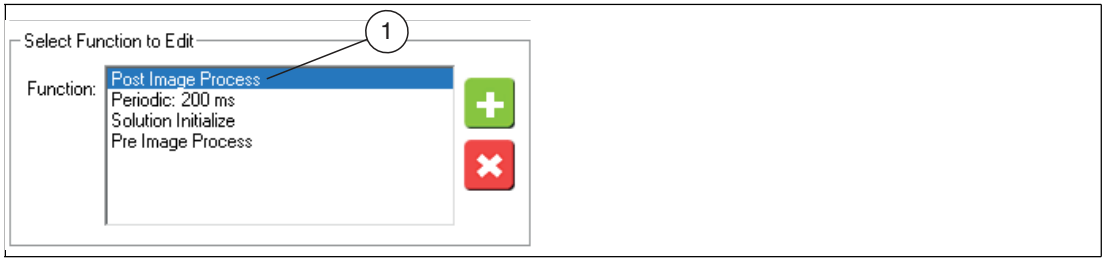


Figure 7.38 Selecting function blocks

8. Select the "Post Image Process" function block from the list (1).

   ↳ The selected function block is accessed after processing the captured image.

9. Create a script to write the output variable of the vision tool to the interface variable (in this example: TcpP12345). To do this, enter your script function (1) in the input window.

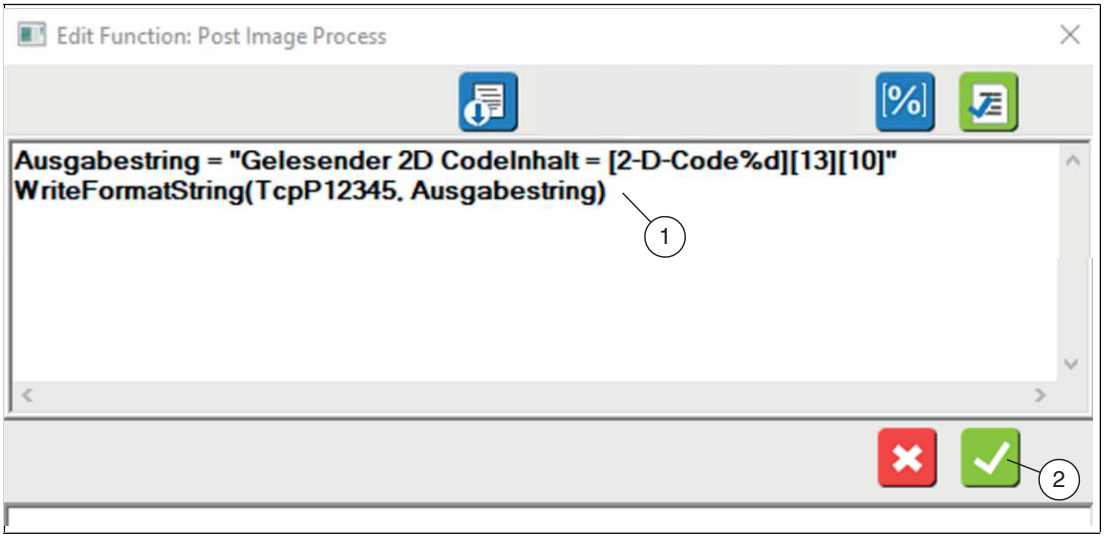


Figure 7.39 Script function

Example

```
ResultString = "Read 2-D CodeContent = [2-D-Code%d][13][10]"
WriteFormatString(TcpP12345, ResultString)
// WriteFormatString() function.
// Control character: [13] = Carriage Return, [10] Linefeed
```

10. Click "OK" (2) to confirm your entry.

Figure 7.40 Application test

11. Test your application by clicking "Run Solution" in the navigation bar to start an "application test" (1). See chapter 7.6.

Parameter Description

Parameter	Description
Protocol	Select TCP or UDP protocol
Client	The VOS software sends data to a network server.
Server	Other computers request data from the VOS software. The requests have no significant impact on the inspection throughput.
Device IP	The IP address of the server to which the data is sent.
Port	If VOS is a server, assign a port number. If VOS acts as client, enter the port number used by the server. Port 5024 is the default assignment and is available to assign your own format and your own conditions. The following port numbers are reserved and cannot be used: all port numbers lower than and including 1024, 5005 to 5023. You should use port number 5024 or higher.
VarName	This field displays a default variable name (e.g., "TCPIP5024"). You can use the suggested name or change it to a more meaningful name.

Displaying Interface Commands

To display interface commands, you need a port monitor. The following example uses the "Hercules SETUP utility" port monitor from HW Group.

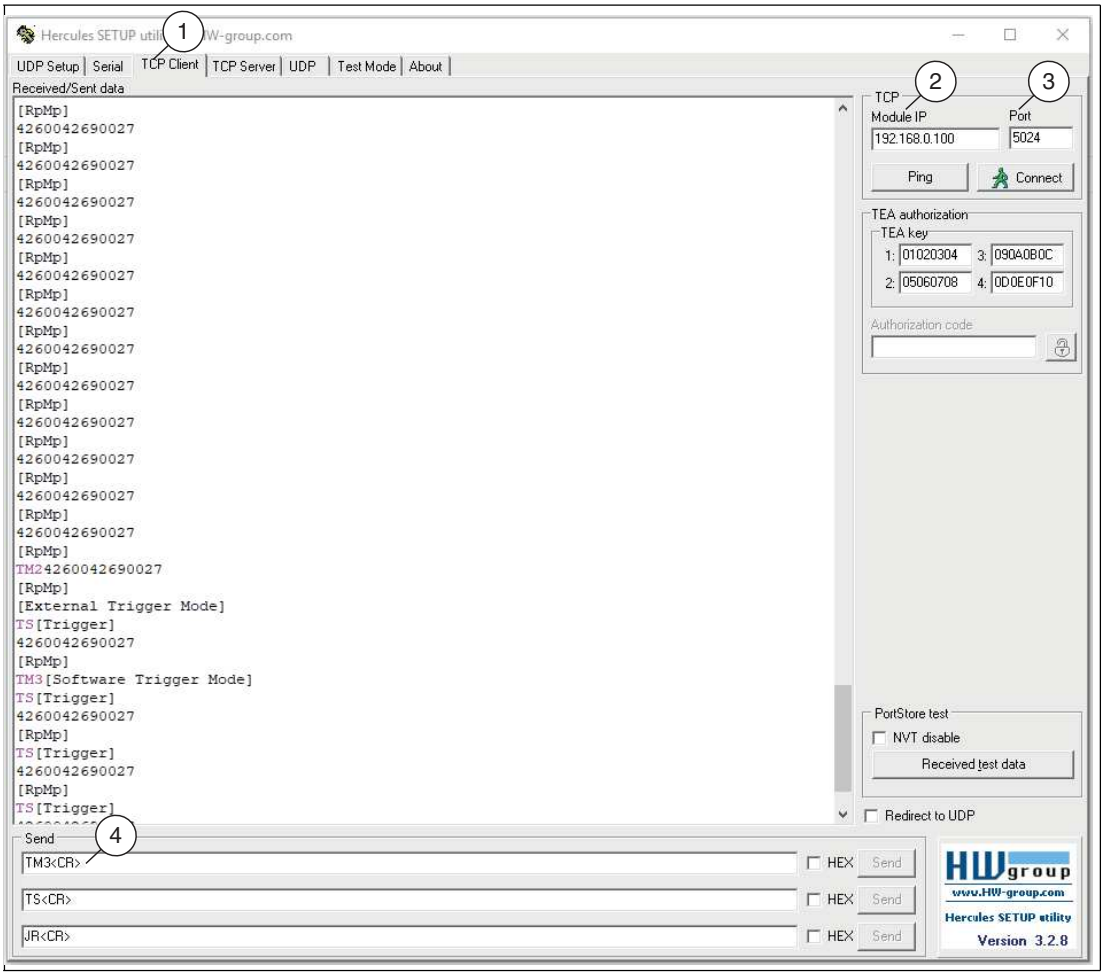


Figure 7.41 Port monitor

1. Open the "Hercules SETUP utility" program and select the "TCP Client" tab (1).
2. Under "TCP", enter the IP address (2) of the sensor and enter the port number 5024 (3).
3. Type "help" (4) followed by <CR> and Enter to display a list of available commands.

TCP/IP Protocol Via Pre-Installed Jobs

The prerequisite for the TCP/IP protocol is having jobs 00 – 05 pre-installed on the sensor. Before using the script, you must select one of the pre-installed jobs 00 – 05 (see chapter 7.1).

The sensor's script periodically monitors the specified TCP/IP port for input commands. These commands are decoded and executed by the sensor. Each command is composed in sequence from the following fields:

- 2 character command identifier
- 1 or 2 character index field
- Variable length character string

There are no spaces between the fields. A command character string must be completed with a CR.

The commands can be entered in the Monitor program, see "Displaying Interface Commands" on page 98.

The following commands are available:

Command	Index	String	Description
TM	1 = internal 2 = external 3 = software	None	Trigger mode
TS	None	None	Software Trigger
JL	00-31	None	Load job
JR	None	None	Job read ID
VS	0 = stop 1 = start	None	Set VOS offline/online
VR	None	None	Read VOS operating state
SR	None	None	Read status changeover
MS	Store string at array index	Match string	Save match string
MI	0-9 (array index of string)	None	Match string index
ME	0 = disable 1 = enable	None	Enable match string
MT	0-9 (array index)	None	Trains and saves the next code (after trigger) at the selected array index
HS	0 = disable header 1 = enable header	None	Enable/disable Send Header
NS	0 = disable 1 = enable	None	Enable Unreadable String
SS	0 = disable 1 = enable	None	Enable/disable Send String
RS	0 = disable 1 = enable	None	Enable/disable Send Result
FE	0 = disable 1 = enable	None	Enable Command Feedback
CR	None	None	Read count
CS	None	None	Reset counters
LE	0 = disable 1 = enable	None	Log enabling

Command	Index	String	Description
LI	None	Image name	Log image name
LU	None	usr:pwd@host	Log user name (FTP)

Example

"TM3" = changes the Trigger mode to Software. Returns [Software Trigger mode].

"TS" = Software Trigger. Returns [trigger generated].

"JL03" = Load job #3 (valid IDs are 0-31). Returns [Switching solution to 3].

"JR" = Job read ID. Returns [Job ID = #].

"VS1" = VOS Online set. Returns [VOS Started].

"VR" = VOS operating state. Returns [VOS Online] or [VOS Offline].

"SR" = VOS switch state. Returns [VOS Switching Enabled] or [VOS Switching Disabled].

"MS5abcdef" = match memory string "abcdef" at index 5. Returns [match string abcdef stored at index #].

"MI3" = "Match Index set to 3". Returns [Match Index = 3].

"ME1" = Match Enable." Returns [Match Mode Enabled].

"MT1" = Match Train enabled. Returns [Match Train Array 1].

"HS1" = Header Send enable. Returns [String Header Enabled].

"NS1" = Noread String enabled. Returns [Noread String Enabled] and "NoRead" if no code was found.

"SS1" = Send String enabled. Returns [Send String Enabled].

"RS1" = Send Status Enabled. Returns [Send Status Enabled].

"CR" = Counts Read. Returns [Cycle Counts = part count, read count, noread count, match count, mismatch count] (match and mismatch count only apply when match string is enabled).

"CS" = Set (reset) counter. Returns [Reset Cycle Counters].

"LE1" = FTP Log Enable. Returns [FTP Logging Enabled].

"LIMyimage" = Log Image Name. Returns [Log Image Name = Myimage].

"LUvos:vosftp@GV400" = Log FTP user name. Returns [Log Username = vos:vosftp@GV400].

Note

There is a certain syntax check for commands as it applies for a valid index. For example, if a "JL34" command is received, it is reported as "invalid." Similarly, a "LE2" command is invalid. The syntax check does not look beyond the valid index, so a command of "LE15" would be considered valid because the command only looks at the first number after the "LE" command. If a command is considered invalid, the sensor returns an [Invalid Command] response.

Note

Job change

If you are using the sensor's graphical user interface for executing tasks, note that some interactions with the graphical user interface disable the job change. This is to prevent jobs from being switched in the middle of the process while the sensor is executing user commands. To enable the job change again, you must save or load the job before accessing the monitor window. When loading the job from the monitor window, the switch is not enabled again.

Note

Switching Disabled

Switching Disabled refers to switching from an external source, script, or network command. You can always switch via the user interface.

Result output

The sensor sends the results or status back to the host every time an image is captured. These outputs include the following:

- 1-D or 2-D decoded data strings (enabled by default, disabled using the "SS0" command)
- 1-D or 2-D code symbology header (enabled using the "HS1" command)
- Two-line result status code (S1-S5) (enabled using the "RS1" command)

The result output strings can be disabled using the "SS0" command.

Note

Disabling result strings does not disable command strings. Use the "FE0" command to disable responses to command sequences. The sensor sends the string outputs based on the commands as shown above.

String Output

VOS scripting provides a high degree of variability in string formatting. For simplicity, the default script defines the following:

- Strings can be sent with or without headers. A header is simply a data string that defines the type of code read (i.e., "Code128"). An example of a string sent with a header is "Code128,12345678."
- Headers and strings are separated by a comma ",".
- The sensor decodes the first 1-D and 2-D codes it finds in the image. If one of the two codes is found, the sensor returns the resulting string. If both codes are found, the sensor returns the 1-D data string followed by the 2-D data string as follows: "Code128,12345678,ECC200,http://www.www.pepperl-fuchs.com"
- Based on the test result, the sensor outputs a second string that indicates the result status when enabled by the "RS1" command. The definition of the status results is as follows:
  - "[Rp]" = Good reading
  - "[Rf]" = Bad or no read
  - "[RpMp]" = Good read and match
  - "[RpMf]" = Good read no match
  - "[Rf]" = Bad read no match
- The result status string can be disabled using the "RS0" command

Match String

The sensor supports the comparison of a decoded character string with a predefined or configured match string. The "good match" status is set if the match is valid.

The match function uses a custom match string or a configured match string. The following describes the match function with custom character strings:

- The user sends a match string to the sensor using the "MS" command. The data string is saved in an array to the index specified by the "MS" command ("MS6ABCD" saves the ABCD data string to index # 6).
- Match string must contain a header and separator (",") if send header is enabled.
- Match works on images with one code (1-D code or 2-D code) or two codes (one 1-D code and one 2-D code).
- If the image contains both 1-D and 2-D codes, the match string must contain multiple codes along with separators and headers if they are enabled. Note that the sensor generates a concatenated result string that begins with 1-D codes, followed by 2-D codes. Each code is separated by a comma.
- The sensor can store up to ten data strings. Each data string is stored in an array that is referenced from 0 – 9 by an index value. Example: The "MS012345678" command saves the "12345678" string in the array at position 0.
- The index value remains until it is changed. Either by a new "MS" command or the "MI" command. Example: "MI7" sets the match index to entry #7.
- The same index is used to reference a data string when executing the match function.
- The match function is enabled using the "ME" command. Example: "ME1" sets the function to "enable."
- When the match function is enabled: At the next trigger, the sensor compares the strings from codes found in the image with the string referenced by the match index value.
- If the data strings match precisely, the sensor returns the "good match" status (if the "send status" function is enabled).

Example: Sending match string to sensor

This example describes how to save a match string on the sensor at array index #3. Set the Software Trigger mode and enable calibration and result status. Triggering capture:

1. Set the Trigger mode to Software Trigger using the "TM3" command.
2. Save the ABCDEF match string to array index #3 using the "MS3ABCDEF" command.
3. Enable matching using the "ME1" command.
4. Enable the result status using the "RS1" command.
5. Trigger a recording using the "TS" command.

↳ If the sensor detects a code with a matching character string, the outputs are as follows:
[Trigger generated]
ABCDEF
[RpMp](good read, good match)

The matching feature is described below based on configured match strings:

- When a "Train Match String" command is received, the sensor performs the following:
  - Saves the next result string in the match array to the index specified in the command
  - Enables the match function.
- The "Train Match String" command is only valid for the next trigger cycle. To configure a new string, a new command must be sent with the corresponding array index.
- Up to ten configured hits can be saved in the array. Example: Configuring and saving to index 0 ("MT0" followed by trigger "TS"). Configure and save to index 9 ("MT9" followed by trigger "TS").
- The match function is disabled while the "Train Match String" function is enabled.
- The match index is used to save and reference data strings in the array.
- The match string automatically saves the symbology header and separator when the header is enabled.
- If there are multiple codes in the image, the composite string is saved in the array at the defined index.
- The match function can be disabled and enabled again using the "ME" command. Example: "ME0" sets the function to "disabled." The configured "Match String" function automatically switches the match function back on and sends the "ME1" command.
- When the match function is enabled: At the next trigger, the sensor compares the data strings from codes found in the image with the data string referenced by the match index value. This value is set using the "MT" command. It can be changed at any time using the "MI" command.
- If the data strings match precisely, the sensor returns the "good match" status (if the "send status" function is enabled).

Example: Saving a configured string

This example describes how to save a configured character string on the sensor at array index #7. Set the Software Trigger mode and enable calibration and result status. Triggering capture:

1. Set the Trigger mode to Software Trigger using the "TM3" command.
2. Use the "MT7" command to set "Match Train" to save the next result in array index 7.
3. Trigger the pull cycle using the "TS" command. The configured code (i.e., 12345678) is saved in the array at index 7.
4. Enable the result status using the "RS1" command.
5. Trigger a recording using the "TS" command.

   ↳ If the sensor detects a code with a matching character string, the outputs are as follows:
   [Trigger generated]
   12345678
   [RpMp](good read, good match)

7.4.6 EtherNet/IP

Setting up the Connection

Menu	Description
Register Type	Select a register type here from the drop-down list for the variable that is attached to the PLC. For example, "int - signed 16."
VariableName	A default variable name will appear (e.g., "EIPint") once the register type has been selected. You can use the suggested name or change it to a more meaningful name.
Reg Base Addr	This field is used to assign a basic index to the register and the variable. The default base address "0" is usually the appropriate option. The attached variable is used as an array. Therefore, any base index specified here will be added to your array index to form the total offset. If you specify a "Reg Base Addr" of "100" for an attached "EIPint" variable, a reference to "EIPint[80]" would reference a position with an offset of 180. (Reg Base Addr 100 + Array Offset 80).

EtherNet/IP Example

This example describes how to add scripts to communicate with the PLC.

1. Click "Edit scripts" in the navigation bar.
2. Click "Post Image Process" in the settings window (left).
3. Click "Edit" at the bottom, below the image area.
4. Create equations or instructions for the control panel. Add your variables. Your variable name appears in the list of variables.

Example

Function: **Post Image Process** //send results to the SPS input register.

EIPdint[0] = IntenAvg

EIPdint[1] = L

EIPdint[2] = Result.0

// Check is complete

inspBusy = 0

5. Click "Check Syntax" to check for errors.
6. Click "Save" to save and to close the "Free Edit" window.

Note

You usually want the sensor to switch jobs and trigger a signal from the PLC:

7. Click "Periodic: 200 ms" in the "Edit scripts" settings field.
8. Click "Edit" at the bottom, below the image area.
9. Add instructions that indicate to the control panel what job is being executed. Read a job change request from the output registers of the control panel. (See example below).
10. Add instructions that trigger the sensor, delete, and then reactivate the trigger.

Example

Send results to the PLC input register:

Function: **Periodic: 200 ms**

```
//Send the current jobs to the PLC
EIPdint[3] = GetSolutionID()
EIPdint[4] = Global.FrameCount
//Query any job change request from the PLC output register
solReq = EIPdint[1]
if(solReq > 0) //this test instruction is different in each job
    ChangeSolution(solReq)
endif
//
//Query any trigger requests from PLC output registers
trigReq = EIPdint[0]
//Trigger only on leading edge of the register transition
if((trigReq = 1) AND (trigArmed = 1))
    trigArmed = 0 //prevents multiple triggering.
    inspBusy = 1 //the check is performed
    trigger()
endif
//Restart trigger if PLC register is 0
if(trigReq = 0) trigArmed = 1
//Record the status of the PLC connection
plcStat = IsConnected(EIPdint)
//
//Sends feedback number
hb = hb + 1
if(hb > 999) hb = 1
EIPdint[5] = hb
```

11. Click "Check Syntax" to check for errors.
12. Click "Save" to save and to close the "Free Edit" window.
13. Save the job, reload the job, and then execute the job.

Configuring the Control Panel

This example shows the RSLogix5000 control panel.

1. Open the PLC programming environment.

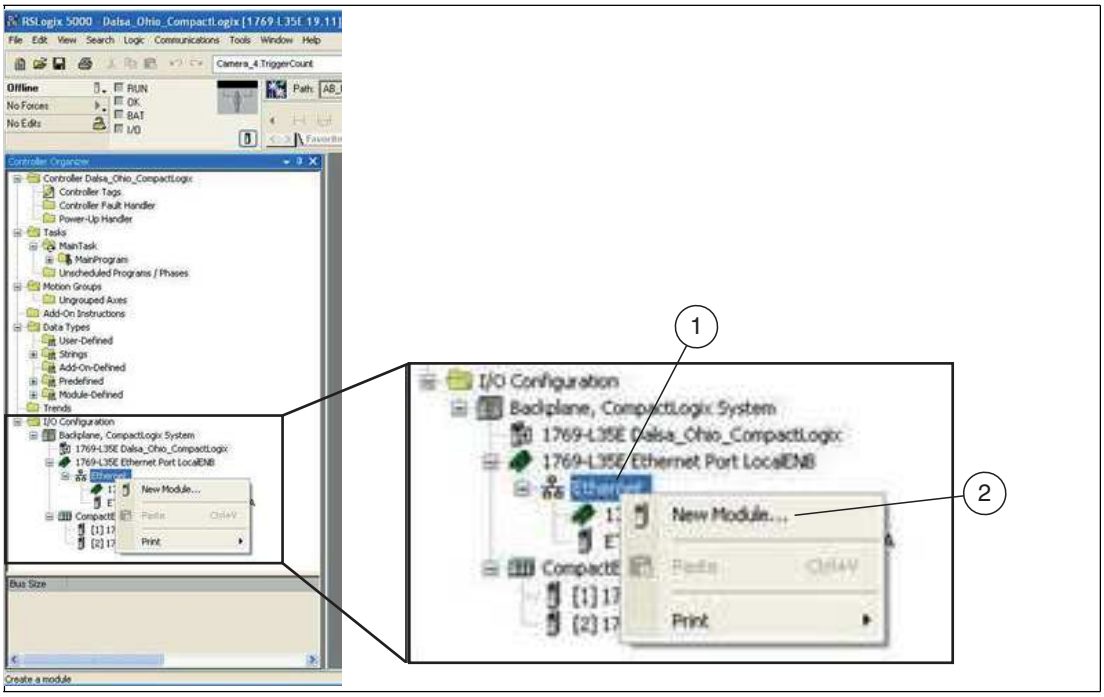


Figure 7.42 Ethernet—selecting a new module

2. Right-click on the Ethernet connection "Ethernet" (1) and select "New Module..." (2).

↳ This opens the "Select Module" selection window.

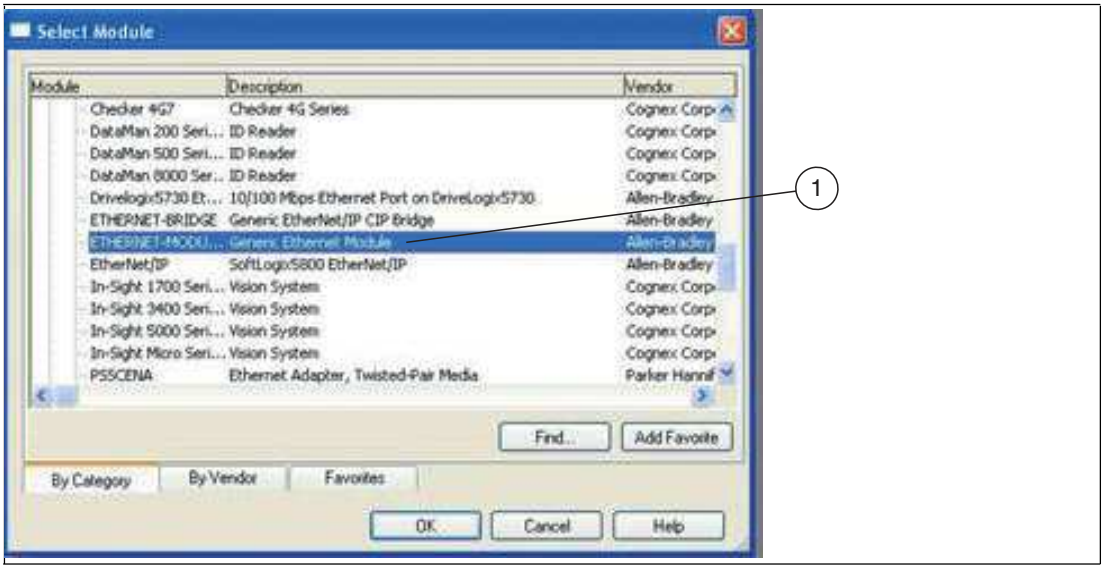


Figure 7.43 "Select Module"

3. In the "Select Module" selection window, select the "Generic Ethernet Module" (1) entry.

2021-07

PEPPERL+FUCHS

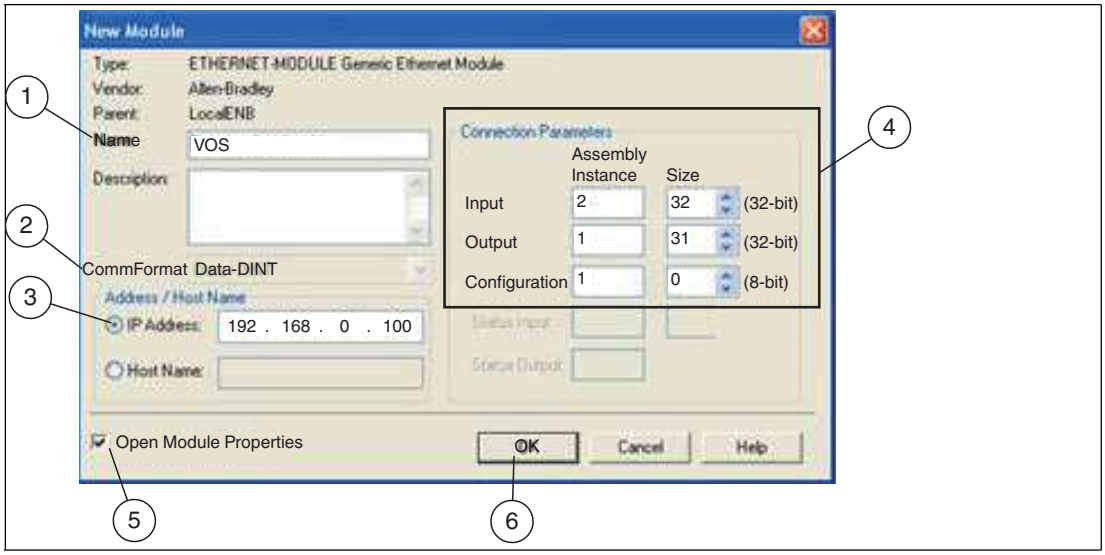


Figure 7.44 Module connection properties

4. In the "New Module" menu, give the sensor a name (1) such as VOS. Keep the CommFormat as Data-DINT (2). Enter the IP address (3) of your sensor.
5. Enter the input, output, and configuration data for the "Assembly Instance" and "Size" (4) as shown:
   - Input: 2, 32
   - Output: 1, 31
   - Configuration: 1, 0
6. Make sure the "Open Module Properties" (5) check box is selected and click "OK" (6).

Note

If more than 30 register values need to be read and written, you could use the following settings:

- Input: 2, 125
- Output: 2, 124,
- Configuration 1, 0

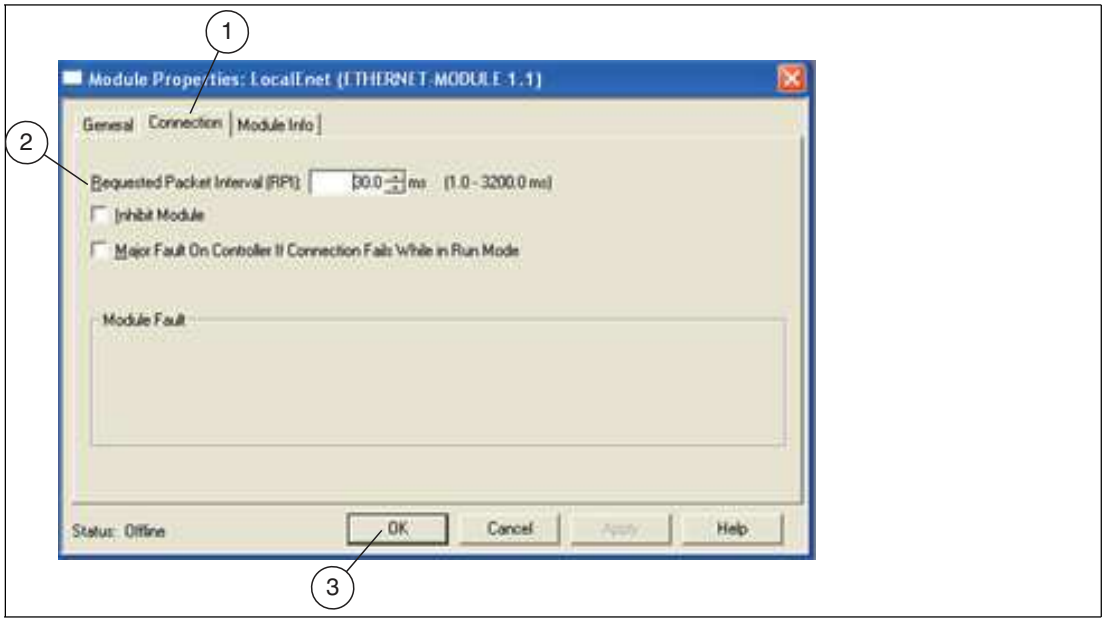


Figure 7.45 Module properties

7. In the "Module Properties" menu, click on the "Connection" tab (1). Enter a "Requested Packet Interval (RPI)" (2) that is not less than 30 ms. This is the time interval during which the Logix processing unit requests new data from the sensor. Depending on network traffic, this number must be significantly higher (150 ms).
8. Click "OK" (3) to confirm your entries.
9. Save your settings and load them to the PLC.

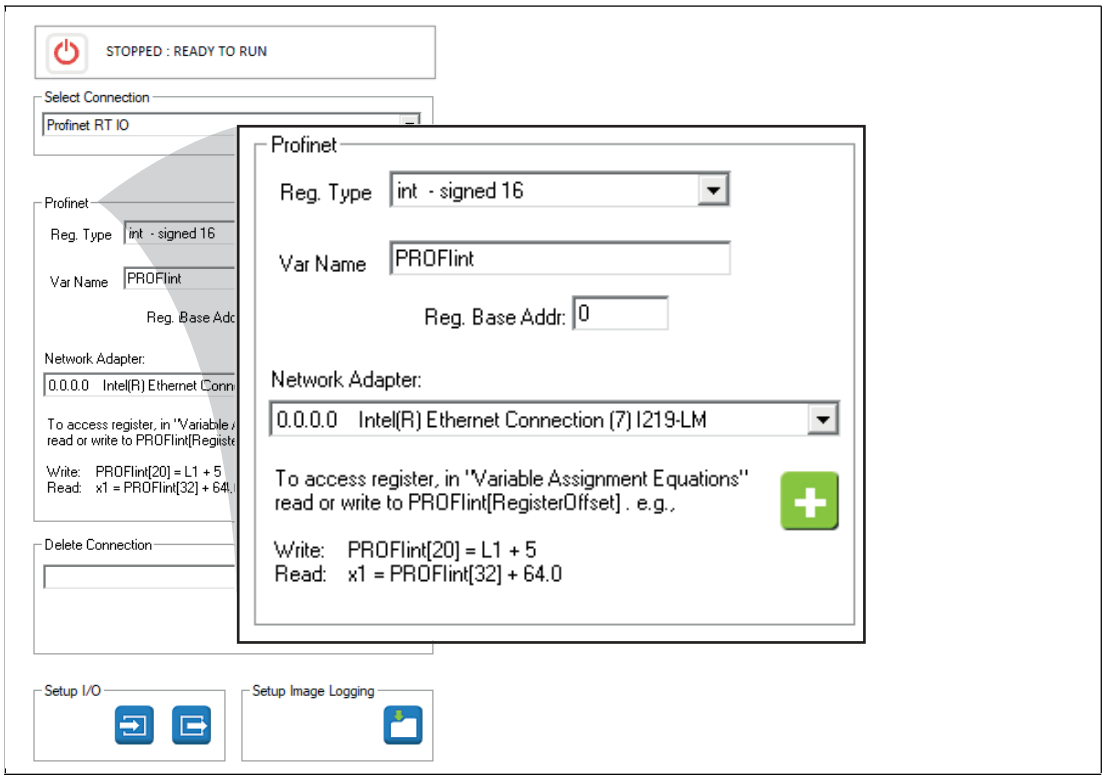
   ↳ The program is compiled before downloading to the PLC. You can also compile the program manually before downloading it.

Note

The sensor writes to EIP[0] and maps to the PLC position I:DATA[1] (I for input). This does not happen during a read operation. EIP[0] is mapped to the PLC position O:DATA[0].

7.4.7 PROFINET

The sensor is compatible with PLCs that use the PROFINET UDP or PROFINET RT protocol.



Setting up the Connection

Menu	Description
Reg. Type	Select a register type from the list. This defines the data type or data size.
VariableName	A default variable name is displayed in this field. The default name is based on the data type (e.g., "PROFIint"). You can use the suggested name or change it to a more meaningful name.
Reg. Base Addr.	This field is used to enter a register base address if your system requires it.

PROFINET Example

This example describes how to add scripts to communicate with the PLC.

1. Click "Edit scripts" in the navigation bar.
2. Click "Post Image Process" in the settings window (left).
3. Click "Edit" at the bottom, below the image area.
4. Create equations or instructions for the control panel. Add your variables. Your variable name appears in the list of variables.

Example

Function: **Post Image Process**

//Send results to the PLC input register.

PROFIdint[0] = IntenAvg

PROFIdint[1] = L

PROFIdint[2] = Result.0

5. Click "Check Syntax" to check for errors.

Note

When the sensor writes to the attached PROFIdint variable, it updates the input module on the PROFINET control panel (I address range 256 ... 509). PROFIdint is a 32 bit signed value. The sensor page references therefore index an array of the dint size.

PROFIdint[0] = x, writes the value that appears at position %ID256 on the control panel,

PROFIdint[1] = x, writes the value that appears at position %ID260 on the control panel, which is the next double-size index.

6. Click "Save" to save and to close the "Free Edit" window.

Note

You usually want the sensor to switch jobs and trigger a signal from the PLC:

7. Click "Periodic: 200 ms" in the "Edit scripts" settings field.
8. Click "Edit" at the bottom, below the image area.
9. Add instructions that indicate to the control panel what job is being executed. Read a job change request from the output registers of the control panel. (See example below).
10. Add instructions that trigger the sensor, delete, and then reactivate the trigger.

Example

Send results to the PLC input register:

Function: **Periodic: 200 ms**

```
//Send the current jobs to the PLC
PROFIdint[3] = GetSolutionID()
PROFIdint[4] = Global.FrameCount
//Query any job change request from the PLC output register
solReq = PROFIdint[1]
if(solReq > 0) //this test instruction is different in each job
    ChangeSolution(solReq)
endif
//Query any trigger requests from PLC output registers
trigReq = PROFIdint[0]
//Trigger only on leading edge of the register transition
if((trigReq = 1) AND (trigArmed = 1))
    trigArmed = 0 //prevents multiple triggering.
    trigger()
endif
//Restart trigger if PLC register is 0
if(trigReq = 0) trigArmed = 1
```

11. Click "Check Syntax" to check for errors.
12. Click "Save" to save and to close the "Free Edit" window.
13. Save the job, reload the job, and then execute the job.

Note

When the sensor writes to the attached PROFIdint variable, it reads the output module on the PROFINET control panel (Q address range 256 ... 509).

profiCmd= PROFIdint[0], reads from memory %QD256 to the control panel,

profiCmd= PROFIdint[1], reads from memory %QD260 to the control panel,

.....

profiCmd= PROFIdint[62], reads from memory %QD504 to the control panel,

Commissioning with Siemens TIA Portal

Creating a Project

To create a project, proceed as follows:

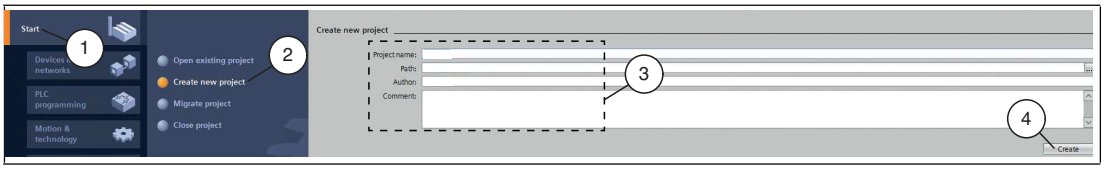


Figure 7.46 Creating a project

1. Launch the TIA Portal.
2. In the portal view, select **Start** (1) > **Create new project** (2).
3. Define fields for the project (3), e.g., by entering a name for the project in the **Project name** field.
4. Use the **Create** button (4) to confirm your entry.

Integrating the Control Panel

To integrate the control panel, proceed as follows:

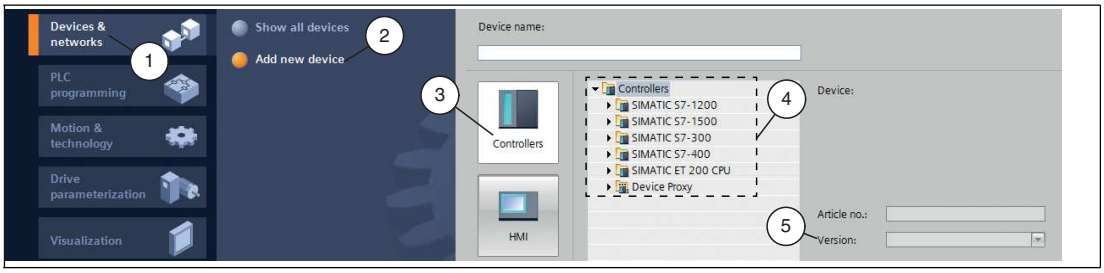


Figure 7.47 Integrating the Control Panel

1. In the portal view, select **Device & networks** (1) > **Add new device** (2).
2. Click the **Controllers** button (3).
3. Select your control panel from the hardware catalog (4). Make sure that you select the right firmware version of the control panel in the **Version** field (5).
4. Double-click on **Add** to add the control panel to the project.

↳ The project view opens.

Note

Now configure the control panel as required. For example, define settings for the PROFINET interface or startup/cycle behavior. Additional information can be found in the Siemens AG user documentation.

Installing the GSDML File

You require a GSDML file to operate the sensor. The GSDML file can be downloaded from our website: www.pepperl-fuchs.com. Enter the product name or item number in the Product/Keyword search box and click on Search. Select your product from the list of search results and click on the Software tab in the product information list. A list of all available downloads is displayed. To install the GSDML file, proceed as follows:

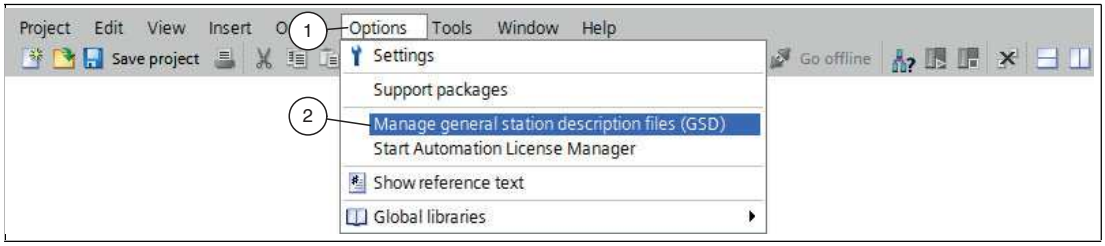


Figure 7.48 GSDML file

1. Select **Options** (1) > **Manage general station description files (GSD)** (2).

   ↳ The **Manage general station description files** window opens.

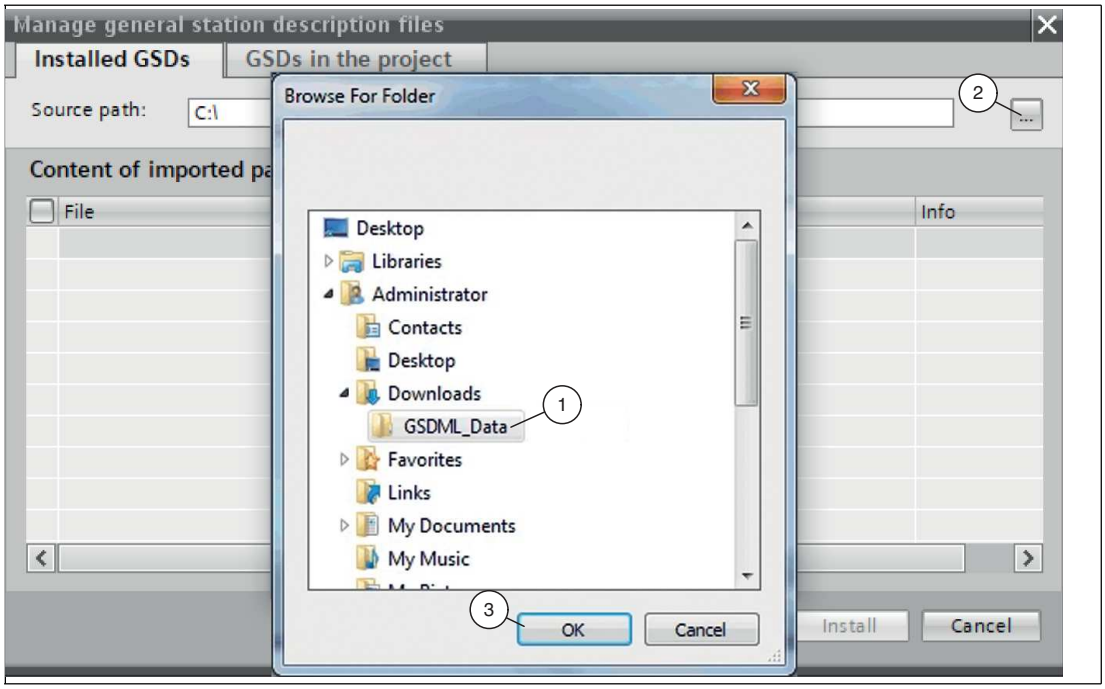


Figure 7.49 Searching for a GSDML file

2. Click the **Browse** button (2) to search for the GSDML file on your computer.
3. Select the folder containing the GSDML file (1).
4. Click **OK** (3) to confirm your selection.

   ↳ All the GSDML files in the selected folder are displayed in the list.
5. Select the relevant GSDML file by checking the box to the left of the file name.
6. Click **Install**.

   ↳ The installation process starts automatically.

   ↳ Once the file is installed successfully, the system issues a notification that installation was successful. Close this window. The device data is added to the hardware catalog.

Note

Ensure that the project is in offline mode. Otherwise, it will not be possible to replicate the hardware components in the configuration software.

Integrating the Sensor

To integrate the sensor and connect it to the control panel, proceed as follows:

1. In the project tree, right-click on the control panel and select **Go to network view**.

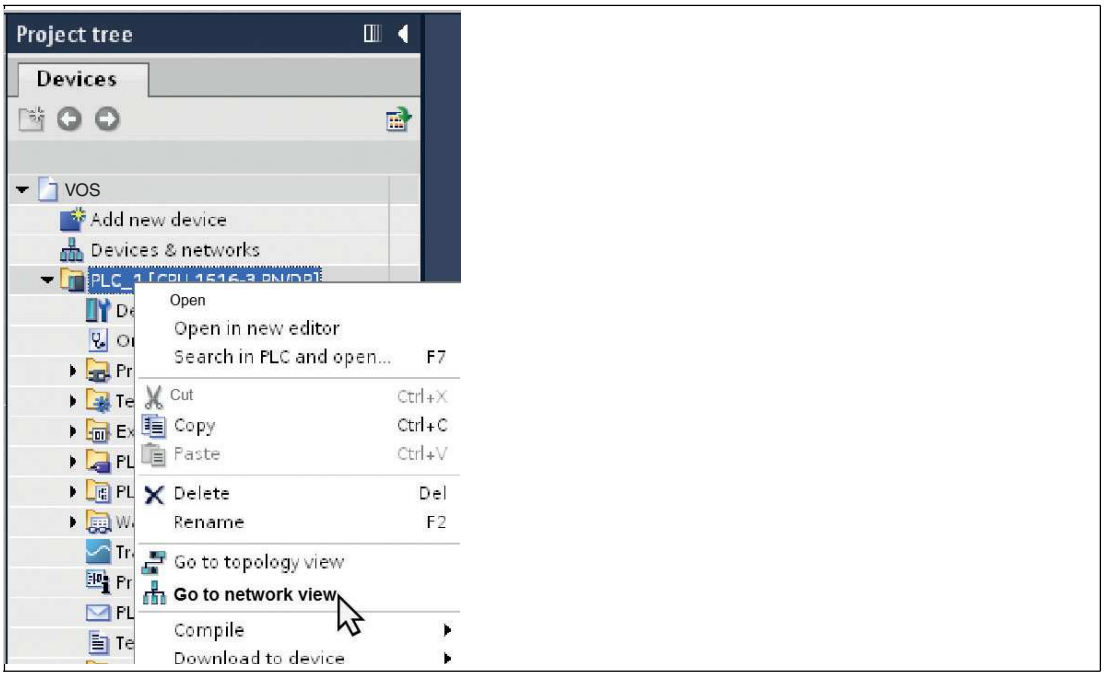


Figure 7.50 Selecting the network view

↳ The **Network view** opens in the work area.

2. Open the hardware catalog and navigate through the tree structure (1) to your sensor.
3. Select your sensor from the hardware catalog (1) and drag and drop it into the network view (4).

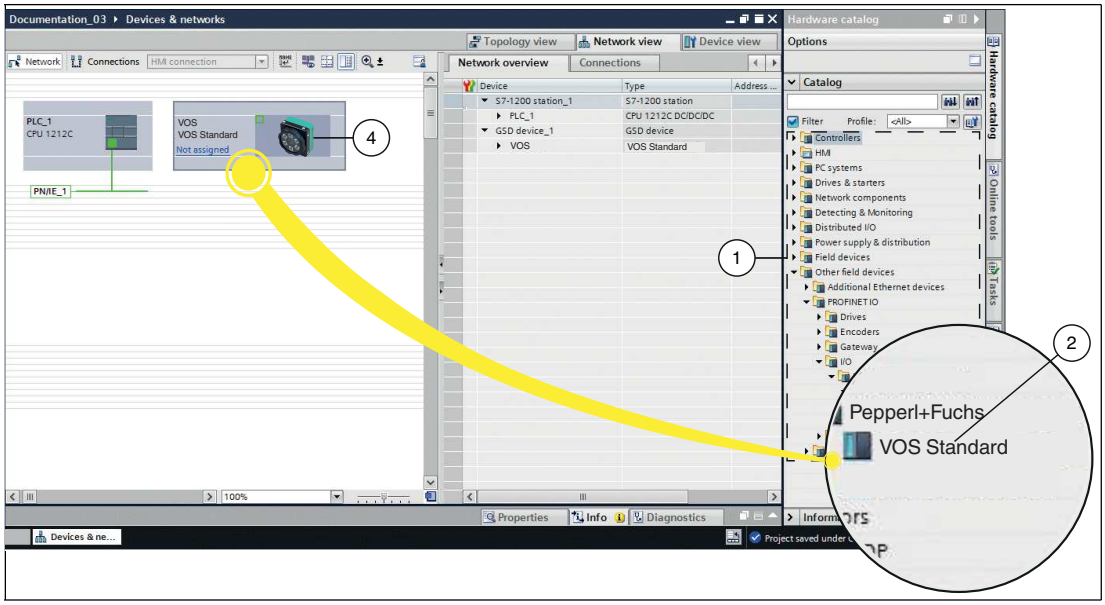


Figure 7.51 Integrating the sensor

↳ The sensor is displayed in the network view window (4).

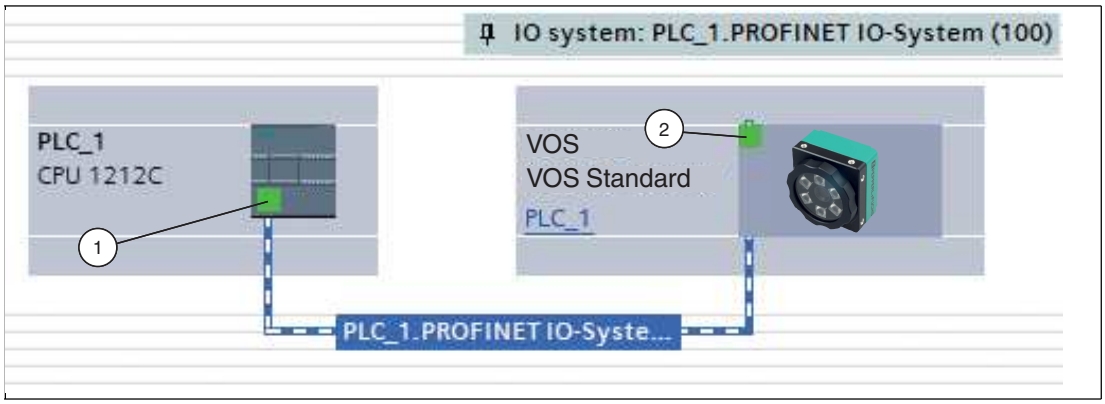


Figure 7.52 Connecting the sensor to the control panel

4. To connect the sensor to the control panel, proceed as follows:

1. Click the green PROFINET interface (1) for the control panel and do not let go of the mouse button.
2. Drag the line to the PROFINET interface (2) on the gateway.
3. Release the mouse button.

↳ The sensor is now connected to the control panel.

Setting the Sensor Name and IP Address

For the sensor to be addressed as a participant in the PROFINET network, the following parameters are required:

1. In the **Devices and networks** window, switch to the **Device view** tab.
2. Select your sensor from the drop-down list. In this example, the sensor is called "VOS."

   ↳ The sensor properties appear in the inspection window.
3. Open the **Properties** (1) > **General** (2) tabs.
4. Click the **Ethernet addresses** subtree node (3).

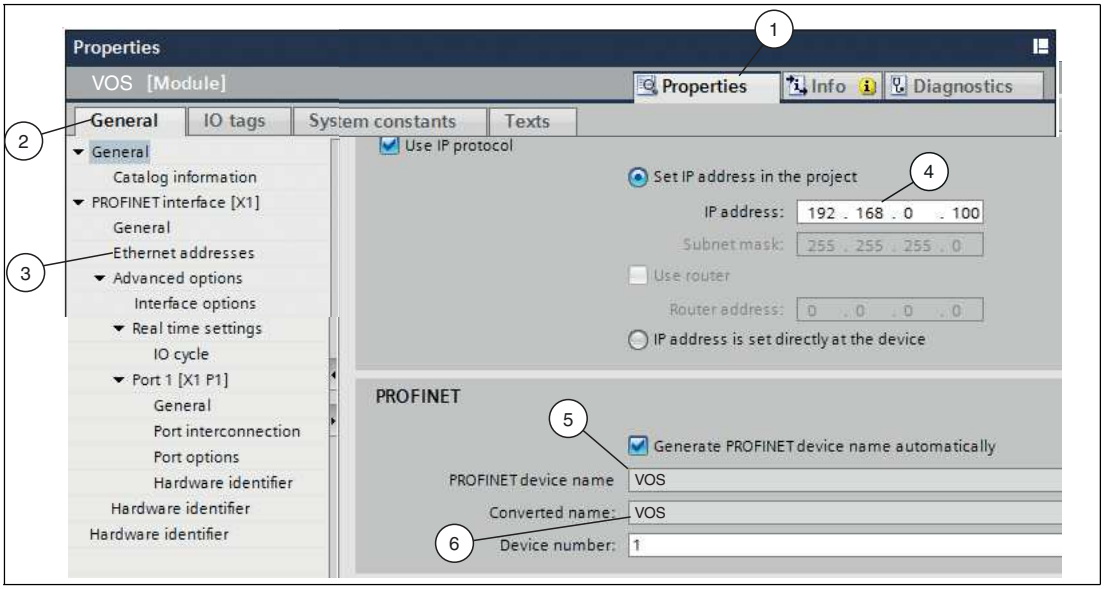


Figure 7.53 Sensor properties

5. If you have selected **Set IP address in the project**, define the IP address in the project. To do this, enter the address of your sensor in the **IP address** (4) field.

Note

Sensor name

In the PROFINET system, each device has a unique device name (symbolic name). In the configuration program, the device name is automatically assigned using the device name from the GSDML file and a sequential number. If using a sensor, the "PROFINET device name" (5) and "Converted name" (6) fields are identical. You cannot change the device name in the view shown above. The "PROFINET device name" (5) sensor must match the name in the "Converted name" field (6), so that PROFINET can connect correctly. If necessary, adapt the device name, see "Changing the IP address" on page 35.

Setting the Watchdog Time

1. Check the "Watchdog time" (3). Go to the "General" (1) tab and select "Advanced options > Real time settings" (2).
2. If necessary, reset the "Update time" (5) and the "Watchdog time" (3). You may need to experiment with the values to determine which watchdog and cycle time works best for your program and your environment. Larger values may be required for a more complex program, a slow trigger rate, or a slow product line.

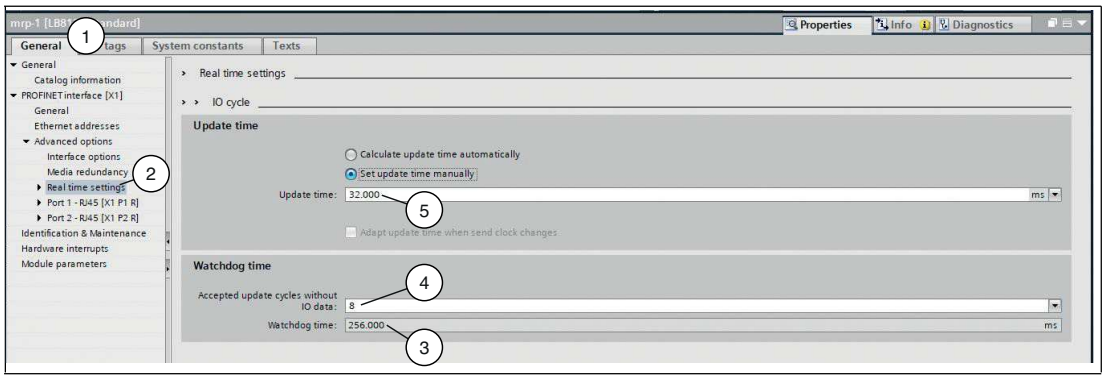


Figure 7.54 Watchdog time

Note

The **"Watchdog time"** (3) is calculated from the **"Update time"** (5) multiplied by **"Accepted update cycles without IO data"** (4).

If, as in this example, eight cycles are accepted without IO data, the resulting watchdog time is 256 ms.

7.5 Script Settings

Creating scripts allows you to define or modify system control and integration functions. The Script Tool differs from the rest of the user interface in that it supports the use of functions and more traditional programming constructs. How the script works with other sensor processes is outlined in section 7.4.5 (see chapter 7.4.5).

Script Editor

In the Script Editor, you can preprocess the variables or data outputs of the vision tools and send them to a communication interface or activate/control the switching outputs according to a script logic. You can define commands to control the camera via an external interface. This allows you to perform tasks such as executing a job change or triggering an image capture.

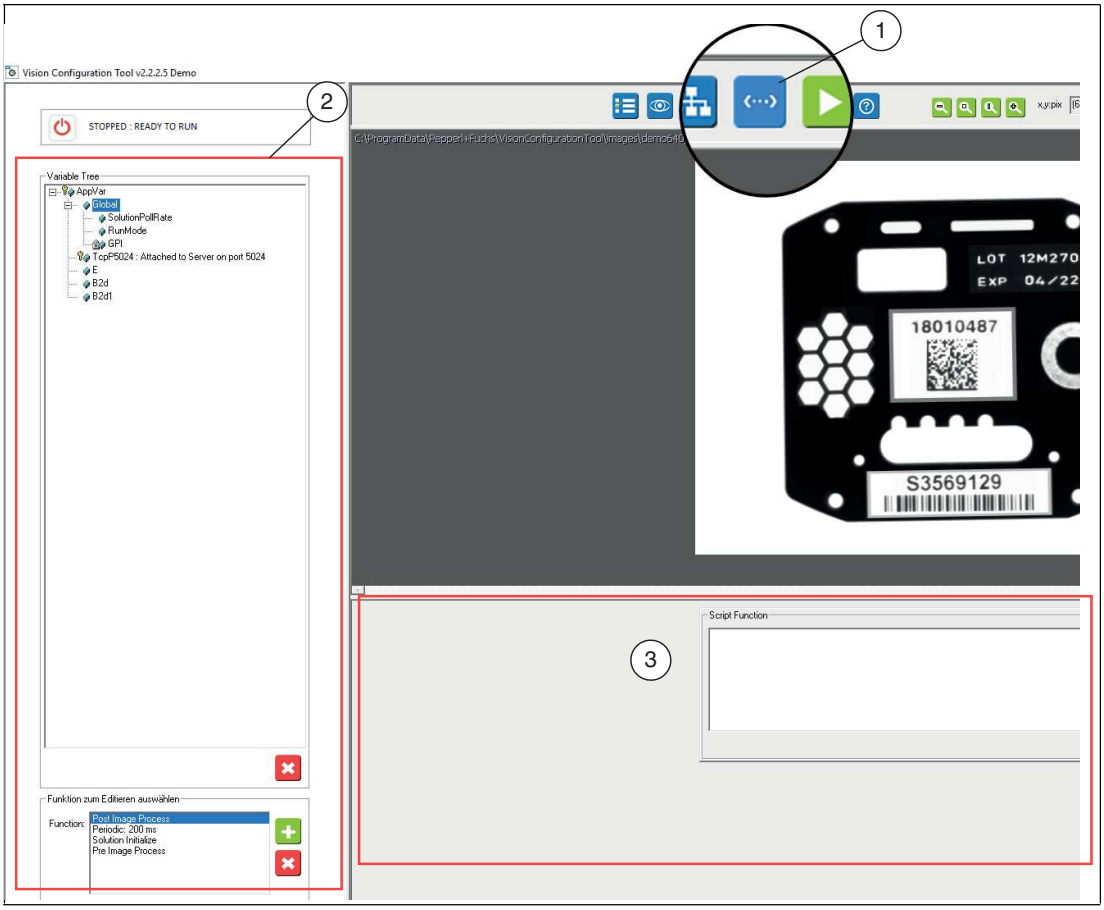


Figure 7.55 Script Editor

Opening the Script Editor

Click the "Script" button (1) in the navigation bar.

↳ This opens the Scripts menu.

Note

There are three sections within the script window. The operating panel (2) on the left contains a tree list of variables and a list of functions. The Script Function manager (3) is located on the right-hand side, below the image window.

7.5.1 Formatting Data Strings

You can use the Data String Editor to create an output string for the COM interface or TCP/IP interface from the existing variables. The Script Editor automatically generates a corresponding function script.

You have the option to write your own data strings. Use "Free Edit" (1) > "Format String" (2) to open the "String Editor" window (3).

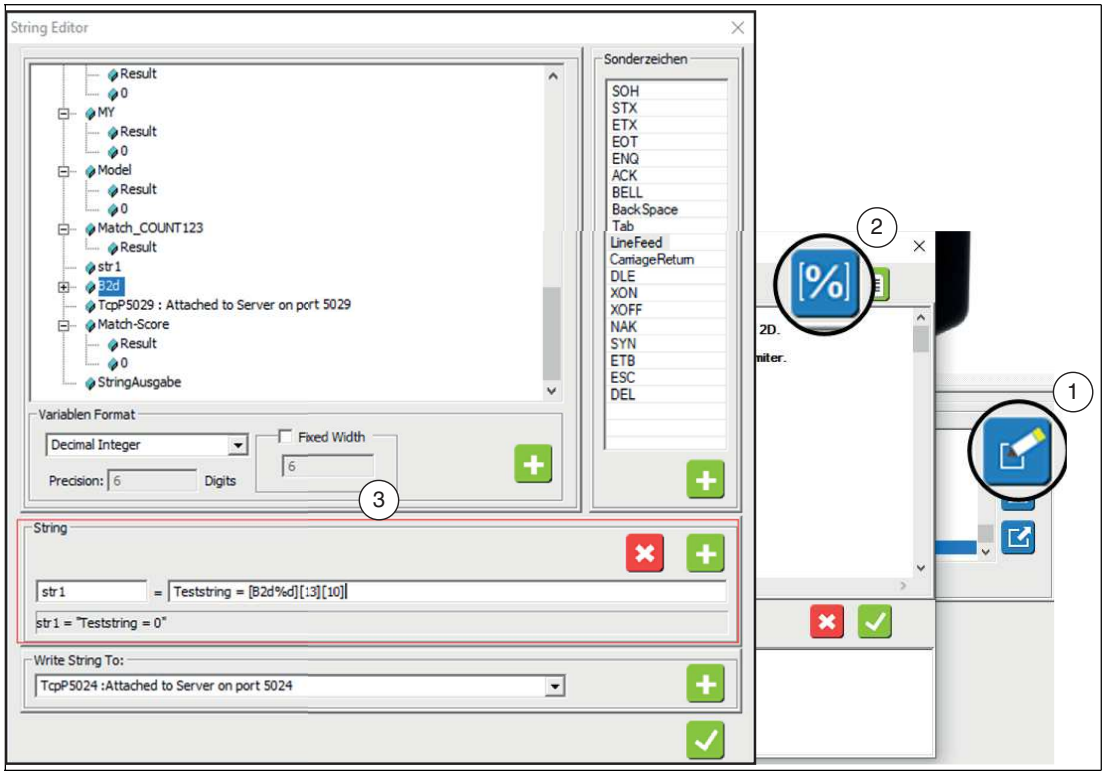


Figure 7.56 String Editors

A variable name for the output string "str1" is defined in the String field.

Fixed ASCII characters are entered in the output string and certain result variables of the camera in square brackets [B2d%d], each with a defined data format (e.g., variable name%d = decimal integer), and control characters such as Carriage Return [13] and Line Feed [10].

Use the plus symbol to assign the string to a previously defined output port, e.g., TCP5024. The corresponding string is created directly in the Script Editor as code and can be used.

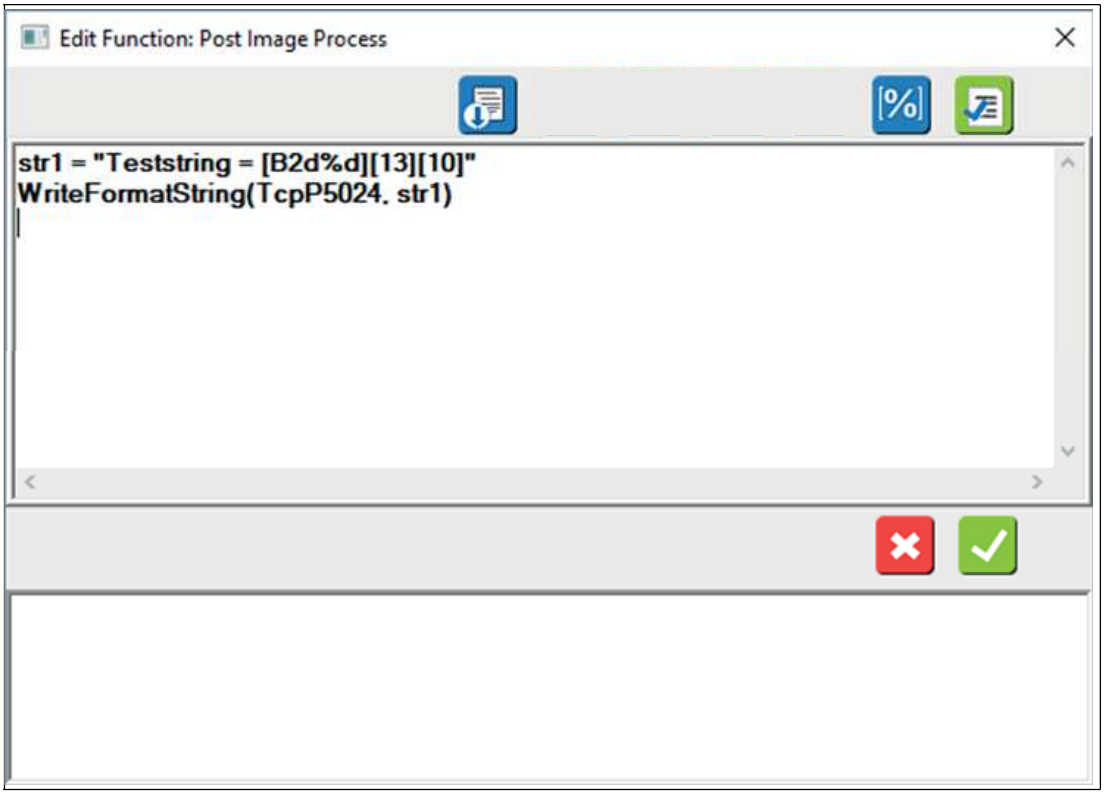


Figure 7.57 Edit Function

7.5.2 Variables

The Variable Tree contains all available variables of the device. This includes the available global result variables (e.g., PassCount), variables of the previously defined vision tools (Boolean, string, integer, etc.) and the previously set interfaces.

There are essentially three types of variables that can be used in a script:

1. **Global device variables:** variables connected to external hardware, such as I/O.
2. **Global control variables:** variables that enable system-wide status or control.
3. **Local user variables:** variables associated with user-defined tools or functions.

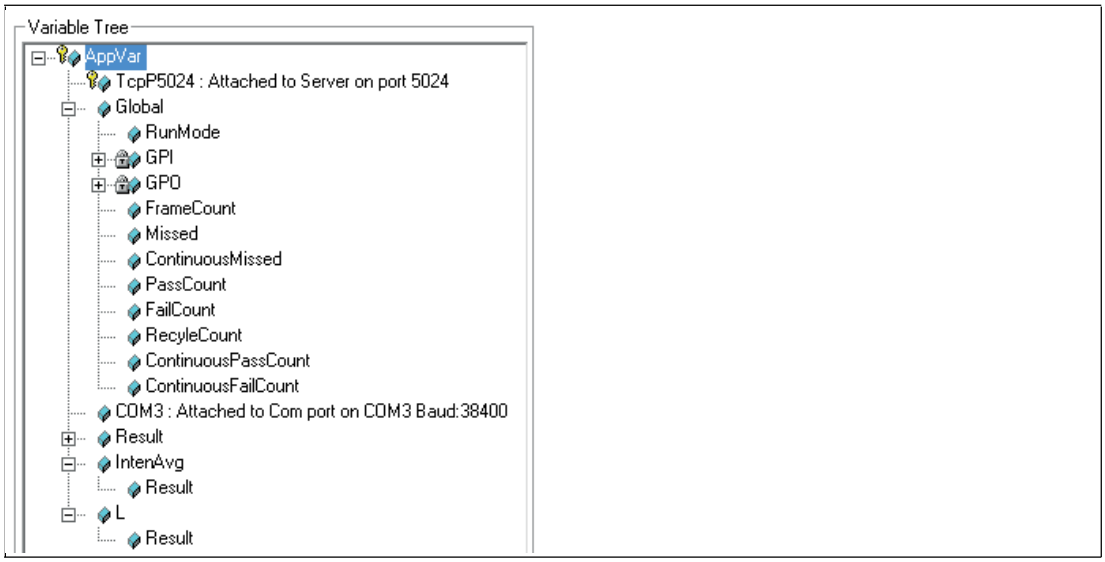


Figure 7.58 Variable Tree

Variable Basics and Nomenclature

Use square brackets for variable names, especially names with spaces. Variable names inserted or dragged into a field are enclosed in parentheses. All of your equations and variables are saved in the job file and are retained when the job file is reloaded. Variables added by the user belong to the current job. When you load another job, your user variable set is replaced by the set belonging to this new job. There are many predefined variables with special functions for use in scripts. You can also create your own variables. When you reference a variable, this variable is automatically created or instantiated. A separate creation or declaration step is not required. The following table lists predefined variables.

Variable Definitions

Variable	Description
Result.0	This variable is read-only. This is the value of "Result" before it is outputted. This allows results to be evaluated by equations before the output is sent to the monitoring unit, the decision I/O, and other communication systems. Result.0 returns three values: 1 = "Pass", 2 = "Recycle", 3 = "Fail."
Result	This variable is read-only. The variable returns the result of all measurements (and all devices, if more than one is connected). This result is sent to the monitoring unit, the decision I/O, and other communication systems (e.g., PLC, Ethernet, serial interface, etc.) The result returns three values: 1 = "Pass", 2 = "Recycle", 3 = "Fail." Each measurement has one result (L1.result).
Global.GPI[#]	This is an input for general applications. The device treats and evaluates all inputs like a stationary logic input.
Global.GPO[#]	This is a universal output. By default, the outputs are kept at "high" or "low" until the next result is available. (The pass/recycle/fail decision outputs are pulsed.) You can use the pulse function or the "delayed event" function to create a pulse output.
Global.RunMode	This is the current operating state or operating mode. 0 = running, 1 = stopped.
Global.FrameCount	This is the number of images that have been captured since job files or programs named "Solutions" were loaded or since the statistics were reset ("Reset Statistics" button on the user interface).
Global.Missed	Number of failed captures. These are also called skipped captures.
Global.ContinuousMissed	Number of successively failed captures.
Global.PassCount	Number of successfully captured objects.
Global.FailCount	Number of failed objects.
Global.RecycleCount	This is the value of the "Recycle" counter or the number of recycled objects.
Global.Continuous-PassCount	The number of objects that were captured in a series or in succession.
Global.Continuous-FailCount	The number of captures that failed in a series or in succession.
Global.SolutionPoll-Rate	If this variable is set to a non-zero value, the Job Change and Job Select inputs are queried at a rate equal to the value of Global.SolutionPollRate in Hertz. For example, if Global.SolutionPollRate = 5, the inputs are queried at 5 z or 200 ms.

Special Global Variables

Some special variables are not visible in the variable tree or in the free editor. These variables are as follows:

Variable	Description
ShowPrepro-cessed	Enables (=1) or disables (=0) the display of preprocessing in the search field at runtime.
RelearnIndex	Uses one of the general input values to trigger the customization or relearning of certain tools (Match, Locator). This variable applies only to VOS devices!
RelearnOnZero	RelearnOnZero =1 forces relearn if the defined relearn input (RelearnIn-dex) is 0 and not 1. This instruction must be added in the "Solution Initial-ize" function. This variable applies only to VOS devices!
Prog	Prog (= permanent variables) is the bracket for variables that are to be retained through job changes. Use "Prog" to create variables that retain their current value and are not deleted when a job file is loaded. Permanent variables are deleted when the device is switched off. Permanent variables are defined with a prog prefix and can be stored in a job file (i.e., Prog.myvariable).

Note

Deleting variables

Select a variable in the Variable Tree and click "Delete" to remove it. You cannot restore a variable once deleted. This affects the output signals, the decision table, and the operating panel. Deleting a measurement tool variable (e.g., L1) does not delete the tool, but the measurement is no longer available for scripting.

7.5.3 Function Blocks

Function blocks consist of functions, equations, or instructions that affect an output or result. Most function blocks can be shared or retrieved by other function blocks (such as subprograms). Some function blocks are executed in sequence (higher-level function blocks), while others are based on a user-defined event, such as a time interval or a transition to a global input. Higher-level function blocks are a special class of function blocks that are executed in a predefined sequence. They can retrieve other function blocks, but cannot be retrieved by other function blocks.

Scripting instructions must be created for communication with a PLC.

Scripting instructions allow you to define the result output and the data format. These can be calculations based on measurement results, displayed messages for the user, or variables that are read from a PLC or written to a PLC.

The function menu lists the four most commonly used functions to retrieve scripts:

Predefined Function Blocks

Function	Description
Post Image Process	This function block is executed after the tools are applied to the image and before the results are sent.
Periodic 200 ms	This function block runs or is repeated every 200 milliseconds. This is useful for checking inputs, status, or PLC communication.
Solution Initialize	This function block runs once when the job file is loaded (from the memory) but before it is launched.
Pre Image Process	This function block runs after an image has been captured and before tools are applied.

Creating or Editing Scripting Instructions

1. Select one of the four function blocks (1) from the list.

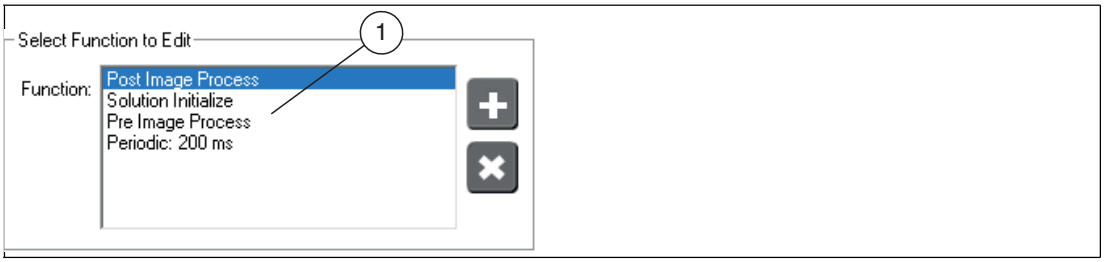


Figure 7.59 Selecting function blocks

↳ If the function block already contains instructions, they appear in the "Script Function" field.

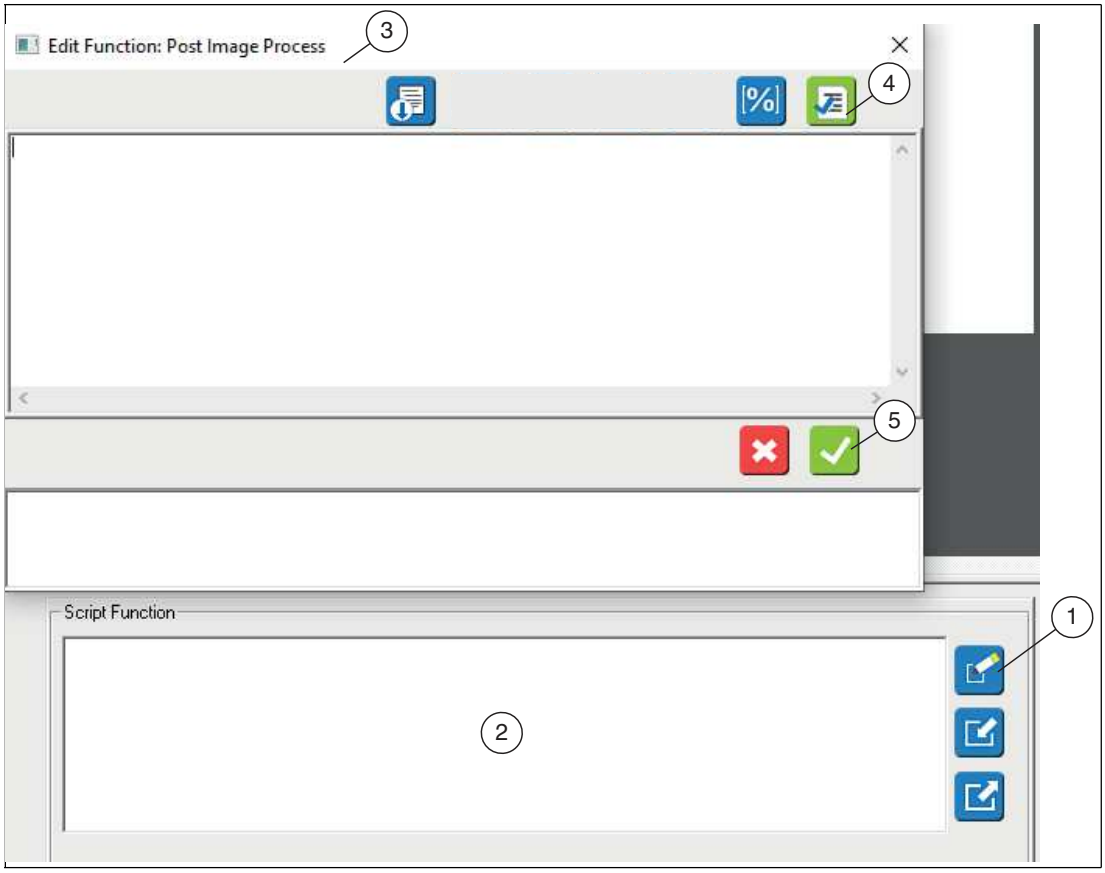


Figure 7.60 Script function

2. Click "Free Edit" (1) or in the text box (2).

   ↳ This opens the "Edit Function:" input window (3).

3. Enter your instruction or function in the input window.

Note

- Instructions are made in algebraic form, for example: `a = b+c`
- Function blocks are called up in the form:
  `z = aFunctionName(param1, param2)`

4. Click "Check Syntax" (4) to check your inputs. Runtime errors associated with the application of the job are not checked.

5. Click "OK" (5) to confirm your entry.

Function Elements in the Script Editor

Function	Description
	Insert a template script. The selection options are trigger or switching solutions from a PLC variable. The variables that you have created are listed. You can create new variables by selecting a PLC type.
	Opens the data string editor window. See chapter 7.5.1.
	Syntax check. Messages appear at the bottom of the window range. You can double-click a message to highlight the code at the top of the window range.

7.5.4 Script Functions

Camera Functions

Function	Description
AutoSaveEnable(enable)	Switches automatic job saving on or off. When autosave is switched on, the job is automatically saved when a user exits the top-level (main level) settings window. The job changeover is automatically enabled. Enable = 0 — turn off autosave. 1 — enable autosave.
ChangeSolution(requested-SolutionID)	Loads a job file. If the "requestedSolutionID" does not exist, the current job file continues to run without fault detection.
ChangeStartupSolu-tion(startupSolutionID)	Changes the job number that is automatically loaded at startup or restart. This change is retained during a system restart until it is changed again. If there is no "startupSolutionID", the system starts with job 00 if this exists.
Copy(source, dest, numEle-ments)	Copies numElements from source (an element field) to dest (an element field). The copy function can be used to update multiple PLC registers in a single transaction.
Delay(milliseconds)	Generates a delay or wait time in a script. This is more efficient than a loop over a timestamp. Adding a delay increases the execution time. Do not add a delay that is greater than your average capture time or the time between triggers. Use Delay() only in a periodic function; it is ignored in all other function groups.
FormatTime(timeVal)	Converts a time value in milliseconds since January 1, 1601 to a character string representing the current date and time. Example: time1 = GetTime()
GetColor(colorID, pixelValue)	Returns the specified primary color value. colorID — indicates the desired primary color: 0 = blue, 1 = green, 2 = red. pixelValue — value for a color pixel, e.g., a value returned by GetPixel. Example: centerPix= GetPixel(0, 320, 240) centerRed= GetColor(2, centerPix)
GetInspectTime()	Returns the inspection time of the previous execution. When it is retrieved in the "Post Image Process" function, this is the cur-rent execution.
GetMaxInspectTime()	Returns the maximum inspection time since the last reset.
GetMinInspectTime()	Returns the minimum inspection time since the last reset.
GetPixel(camID, x, y)	Returns the value for the pixel specified by camID, x, and y. x: the x coordinate. 0 = the left-most column. y: the y coordinate. 0 = top line. Example: centerPix= GetPixel(0, 320, 240)
GetSolutionID()	Returns the current job ID number.
GetTime()	Returns a value representing the current date and current time. The value returned is the number of milliseconds since January 1, 1601 (in the local time zone). See "FormatTime" function. Note: "GetTime" is not accurate enough to accurately calculate your cycle time to milliseconds. Use the "TimeMillisec" function for these calculations.
GetTimeString()	Returns a data string value representing the current date and current time (local time zone). Example: "now = GetTimeString()" sets "now" to a string value of "9/7/2009 16:25:28:429." dateString= FormatTime(time1) sets "dateString" to a string value of "9/7/2009 15:25:28:429."

2021-07

Function	Description
GetUserName()	Returns the user name for the currently logged in user.
GetVersion()	Outputs the firmware version of the sensor (the Vision Configuration Tool version).
ImageProcess(DllFileName, functionName, numVars, VarArray)	Retrieve a "functionName" function exported from the "DLL DllFileName." numVars — the number of elements in the "VarArray" field. VarArray — parameter field with a size of up to 20 elements. Parameters can be used as In and/or Out.
Print(string, endOfFile)	Sends a character string of data to the default printer. Does not evaluate embedded variables. endOfFile - 0 = data string is stored temporarily until a subsequent print retrieval is made with "endOfFile" = 1. 1 = All saved text from previous retrievals to print and the transferred character string are sent immediately to the printer. Note: Use this function carefully because every retrieval with "endOfFile=1" causes at least one page to be printed.
SetDisplayStatus(statusMsg, color)	Specifies the message that appears in the "Inspection Status" field (in the configuration and status window) linked to the monitor window. This overrides the display of "Pass" or "Fail." statusMsg — the character string that is displayed. For multiple lines, add the \n character to display a new line. The body of the message is automatically large enough to be as large as possible and still appear in the "Inspection Status" field. String formatting information for variables of the [Var%FormatData] form is supported. msg1="[L1%0.2f]" means that the value of L1 is displayed with two digits to the right of the decimal point. Color — the string name of the color in which the message text is displayed. Possible values are: "black", "red", "green", "yellow", "blue", "magenta", "cyan", "white", "dark red", "dark green", "dark yellow", "dark blue", "dark magenta", "dark cyan", "light gray 1", "yellow green", "sky blue", "cream", "light gray 2", "medium gray."
StartInspect( )	Starts the image inspection.
SetAppButton(buttonNumber, buttonName)	Creates a user-defined button in the Run or Monitor operating panel. Up to four buttons can be added. A "User" function group appears with an "X button clicked" for each defined button. encodeMethod • 0 = without compression. • 1 = JPEG compression (default for color). • 2 = proprietary high-speed compression (default for mono).
SetImageEncode(encodeMethod)	Changes the image encoding used to compress images sent to a connected end device.
StartInspect( )	Starts image inspection.
StopInspect( )	Stops image inspection. Note: Use this function in a periodic function. The function must not be used in image post-processing.
SwitchingIsEnabled( )	Returns 1 if job switching is enabled. Returns 0 if job switching is not enabled.
TimeMillisec( )	Returns the current time in milliseconds.
return(FunctionReturnValue)	Returns the specified value "FunctionReturnValue" from a user-defined function. Example: Return((p1 + p2) /2).

Statistical and Vision-Tools-Related Functions

Function	Description
GetMean(measurementVar)	Returns the arithmetic mean for the specified measurement. Example: L1Mean = GetMean(L1)
GetStdDev(measurementVar)	Returns the standard deviation for the specified measurement. Example: L1StdDev = GetStdDev(L1)
GetMin(measurementVar)	Returns the minimum value that occurred for the specified measurement. Example: L1Min = GetMin(L1)
GetMax(measurementVar)	Returns the maximum value that occurred for the specified measurement. Example: L1Max = GetMax(L1)
ResetVarStats(measurementVar)	Resets the measurement statistics (min, max, mean, std dev) for the specified measurement. All earlier data samples for the measurement will be deleted for statistical calculations. Example: ResetVarStats(L1)
GetToolType(measurementVar)	Returns a number indicating the type of measurement tool for the specified variable. Example: L1type = GetToolType(L1) variable L1type returns the value 6.
GetNthToolType(varIndex, CamID)	Returns a number indicating the type of measurement tool for the specified variable index and camera ID. Returns zero if no tool exists for "varIndex" and "camID." varIndex — 0 to (number of tools minus 1) that exist for the specified camera.
GetToolName(varIndex, camID)	Returns the simple name (no camera prefix) of the measurement tool, for the specified variable index and camera ID. varIndex — 0 to (number of tools minus 1) that exist for the specified camera.
RequestRelearn(measurementVar)	Causes the measurementVar to be re-taught in on the next image. Example: RequestRelearn(Bar)
SetTolerances(measurementVar, toleranceArrayIn)	Specifies the five tolerance "pivot points" for the specified measurement variable. measurementVar — variable name. toleranceArrayIn — five-element field of tolerance values. (as shown below, under GetTolerances).
GetTolerances(measurementVar, toleranceArrayOut)	Obtains the five tolerance "pivot points" for the specified measurement variable. measurementVar — variable name. toleranceArrayIn — five-element field of tolerance values. (as shown below, under GetTolerances). • Index 0: minimum recycling value or minimum value for "Pass" • Index 1: minimum value for "Pass" or 0 • Index 2: perfect value • Index 3: maximum value for "Pass" or 0 • Index 4: maximum recycling value or maximum value for "Pass"
SetMatchString(measurementVar, perfectMatch)	Sets a "measurementVar" data string a new "Perfect-Match" value. Used only for measurements that return a data string (barcode, 2-D code, OCR) to determine a new "perfect" value.

2021-07

Function	Description
SetNthTolerances(varIndex, camID, toleranceArrayIn)	Sets the five tolerance "pivot points" for the specified variable index and the camera ID. varIndex — 0 to (number of tools minus 1) that exist for the specified camera. toleranceArrayIn — five-element field of tolerance values (as shown above in GetTolerances).
GetNthTolerances(varIndex, camID, toleranceArrayIn)	Sets the five tolerance "pivot points" for the specified variable index and the camera ID. varIndex — 0 to (number of tools minus 1) that exist for the specified camera. toleranceArrayIn — five-element field of tolerance values (as shown above in GetTolerances).
GetToolValue(toolName)	Outputs the measured value for the tool. The tool name ("L1") is transferred. Example: value = GetToolValue("L1")
GetToolResult(toolName)	Returns the result value for the tool. The tool name ("L1") is transferred. Example: value = GetToolResult("L1")
GetVarDimension(varName)	Returns the number of child variables of the "varName" variable.
SetParam(numArgs, argList)	Changes the position of an existing region (ROI) using a list of data arguments. • numArgs — the number of data arguments in "argList" • argList — a list of values for setting the ROI parameters.
SetToolText(toolName, toolText)	Sets or changes the "toolText" that appears in the "toolName" Graphics Tool (Express only).
SetToolPenColor(toolName, red, green, blue)	Sets or changes the outline of the "toolName" Graphics Tool to red, green, or blue. Express only.
SetToolFill(toolName, red, green, blue)	Set or change the fill color of the "toolName" Graphics Tool to red, green, or blue. Express only.
WriteVar(narName, value)	Writes a value to a script variable. • varName — the name of the variable to be written to. • Value — the value to be written to the variable. Example: WriteVar("InputThreshold", 4.5)
Sort(keyVarName, ascend, followVarNames)	Sorts the field specified by "keyVarNam" in ascending order. • ascending - 0 = sort in descending order. 1 = sort in ascending order. • followVarNames — an optional list of the names of field variables that are sorted in the same reassignment as "keyVarName." Example: If the index value 0 is moved from keyVarName to index 5, the same reassignment occurs for all "followVarNames" fields.
ReadVar(varName)	Reads the value of a script variable. varName — the name of a script variable.
DeleteVar(varName)	Delete the "varName" variable. This is useful for removing a variable that is left over from a previous script or function.
GetNumElements(varName)	Returns the actual dimension or number of elements for a "varName" measured variable that generates a field of values.
EnableFormat(varName, enable)	Enables or disables the output of the "varName" measured value in the output log or script output. enable - 0 = does not output "varName." 1 = allows output of "varName."

Mathematical Functions

Function	Description
sin(radians)	The result is the sine of the angular dimension. The argument must be specified in the angular dimension.
cos(radians)	The result is the cosine of the angular dimension. The argument must be specified in the angular dimension.
tan(radians)	The result is the tangent of the angular dimension. The argument must be specified in the angular dimension.
asin(x)	The result is the arc sine of x in the range -p/2 to p/2 angular dimension, where -1 <= x <= 1.
acos(x)	The result is the arc cosine of x in the range -p/2 to p/2 angular dimension, where -1 <= x <= 1.
atan(x)	The result is the arc tangent of x in the range -p/2 to p/2 angular dimension.
atan2(y, x)	The result is the arc tangent of y/x in the range -p to p angular dimension.
exp(x)	The result is the exponential value of x.
log(x)	The result is the natural logarithm of x.
sqrt(x)	The result is the square root of x.
pow(x, y)	The result is set x to the power of y.

Data String Functions and Character Functions

Function	Description
Find(substring, inString)	Finds the first occurrence of a substring in the inString input and returns the zero-based index digit of the first matching character. Returns -1 if no match was found. Spaces are counted. Example: idx = find("00", "SM WRA 0057 4321") returns 7, or sets idx= 7.
Substring(string, startIndex, length)	Forms a substring from the input string, starting with the start index (zero-based) of length characters. If length = 0, all characters up to the end of the data string are included in the substring. Example: s2 = substring("SM WRA 0057 4321", 9, 0) returns string "57 4321" in s2.
StrLen(string)	Outputs the number of characters in a character string.
GetChar(string, index)	Outputs the character that is in the data string at index (zero-based).
SetChar(string, index, char)	Sets the character in string, which is at the index (zero-based), to "char."
int(string)	Converts the entered character string (of numbers) to an integer value. Example: x = int("33") sets x = 33
float(string)	Converts the entered character string (of numbers) to a floating-point value. Example: x = float("57,499") sets x = 57,499
char(int)	Converts the entered integer to a character.
string(int)	Converts the input number to a character string, e.g., an integer int(base 10), or a floating-point number. A floating-point number uses the same formatting as "WriteFormatString." Example: LengthStr = string("[L%0.3f]")
FormatString(stringForm)	Outputs a formatted character string by using the format in the "stringForm" argument. "stringForm" can be created using the String editor. Example: The statement str1 = FormatString("[IntenAvg%.4f]") results in a string "55,7015" (returned in str1).

Note

For additional functions, refer to the Vision Configuration Tool online help section.

7.5.5 Script Examples

The following examples show how to use the Script Tool. Each example includes a brief description of the application and associated code excerpts from the corresponding functions. The examples cover basic script concepts that apply to standard applications.

Using Script Command to Switch Switching Outputs

pulse(activeVal, offsetMillisec, durationMillisec)—generates a pulse output.

activeVal—1 = active high pulse, 0 = active low pulse.

offsetMillisec—Offset or delay in milliseconds from the time this specification is executed.

durationMillisec—Pulse duration in milliseconds.

This function is executed in the **Post Image Processing** function.

GPIO Settings

Purpose/Task	Script	Description

After global result variable (result), output a pulse signal either via GPO[0] or GPO[1].	if (Result=1)
	Global.GPO[0] = pulse(1,0,20)
	else
	Global.GPO[1] = pulse(1,0,20)
	endif()
		The equation outputs an active high pulse of 20 ms to GPO [0] (with no delay) if the job result is equal to 1 (Result=1).
		The equation outputs an active high pulse of 20 ms to GPO [1] (with no delay) if the job result fails.

FTP File Transfer

There are several ways to transfer images. This example shows how to store images on an FTP server.

WriteImageFile(fileName, camID): Retrieved only from the "Post Image Process" function, transfers the current image from the sensor specified by **camID** to the specified **fileName** (0 for the sensor).

fileName: Full path of the file to be saved.

In this example, no user name or password is required. The name of the image file is "img#.bmp."

Function: **Post Image Process**:

x = x+1

fn = "ftp://ftp.xyz.com/images/img" + x + ".bmp"

WriteImageFile(fn, 0)

Text Customization on the User Interface

You have the option of using scripting to display results or messages in the "Display Status" window on the user interface.

If(Result= 1) // if the global result variable is PASS=1

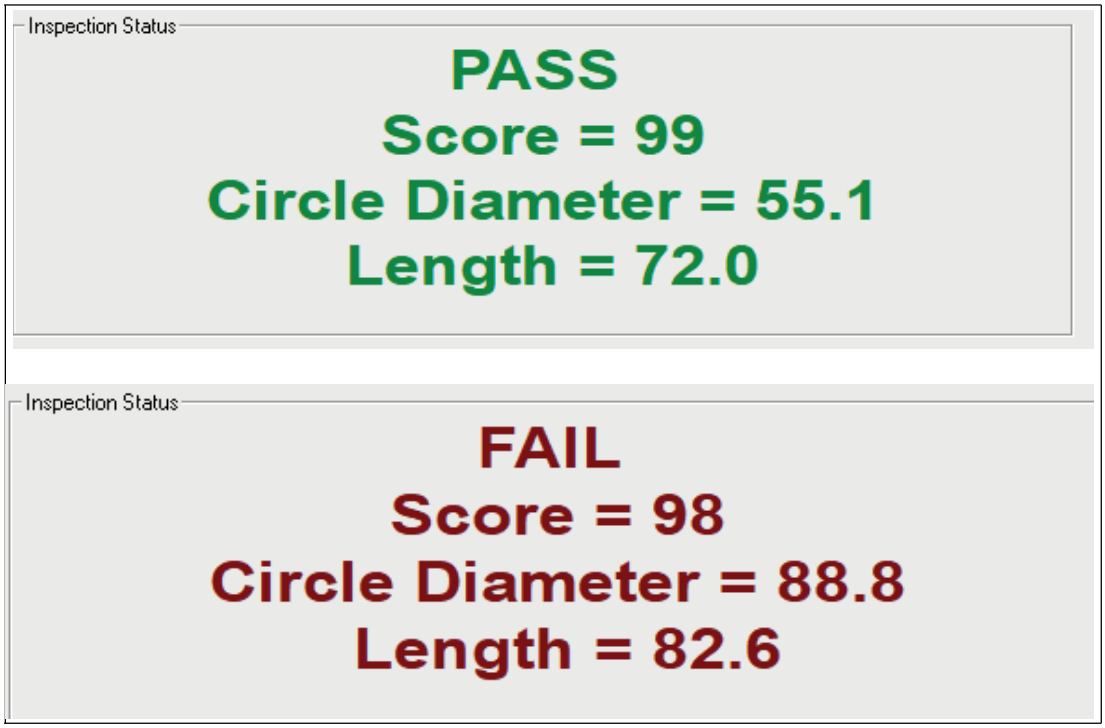
SetDisplayStatus("PASS \n Score = [MS%d] \n Circle Diameter = [CDiam%.1f] \n Length = [L%.1f]", "darkgreen") // Set the user interface to display the tool results in green

else

str1 = "FAIL, Score = [MS%d]. Circle Diameter = [CDiam%.1f], Length = [L%.1f] [13][10]" // Set the user interface to display the tool results in red

SetDisplayStatus("FAIL \n Score = [MS%d] \n Circle Diameter = [CDiam%.1f] \n Length = [L%.1f]", "darkred")

endif



The string name of the color in which the status indicator appears. Possible values are: "black", "red", "green", "yellow", "blue", "magenta", "cyan", "white", "dark red", "dark green", "dark yellow", "dark blue", "dark magenta", "dark cyan", "light gray 1", "yellow green", "sky blue", "cream", "light gray 2", "medium gray."

Enabling and Disabling Vision Tools

You can disable the "Factory Settings" in the tool table (). The tool is hidden in the image and the results are disabled. All results are displayed with = 0. The tool table disables the tool for all image captures. You can also disable a tool in Scripts. The tool is enabled or disabled on the basis of a variable or input. Use the tool name with the "Disable" switch. Disable=1 hides the tool and disables its results. Disable=0 displays the tool and enables its results. The tool remains disabled if you do not enable it again. If you are disabling a tool on the basis of a variable, we recommend the "Pre Image Process" function. If you want to disable a tool on the basis of an input, you can periodically use the function used to monitor the input.

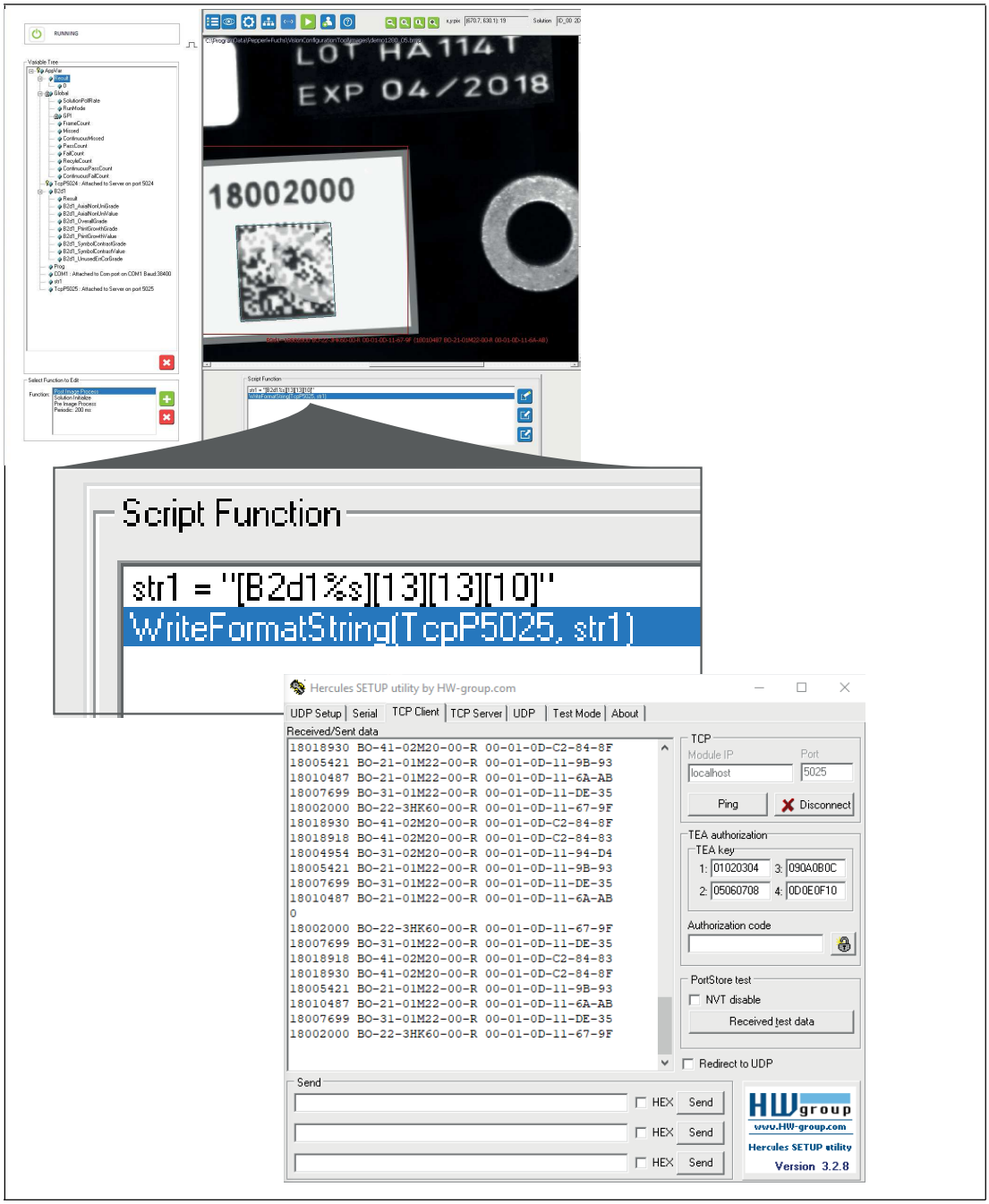
```
If(GPI[3]=1)
   MS.Disable=1 // Disable and hide Match Tool if input = 1.
else
   MS.Disable=0 // Enable & display Match Tool if input = 0.
endif
```

TCP/IP

Outputting of a Complete String via TCP/IP

Note

In the following examples, the script and output are generated via the emulator. Therefore, "localhost" is entered in "Hercules SETUP utility" under "Module IP." If you display the TCP/IP output from the sensor, the correct IP address must be entered instead of "localhost." The default IP address of the sensor is 192.168.0.100.

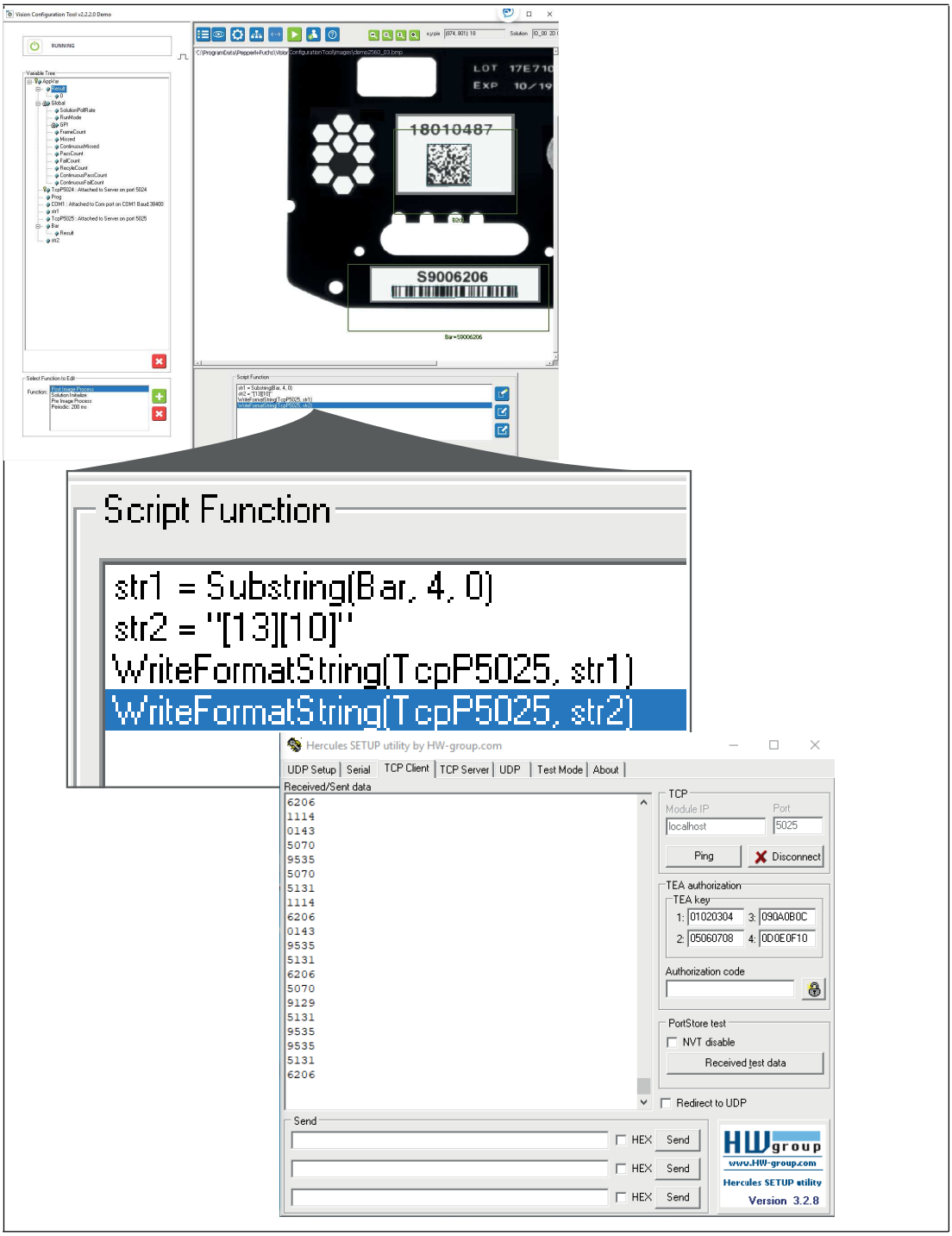


str1 = "[B2d1%s][13][13][10]"

WriteFormatString(TcpP5025,str1)

Substring via TCP/IP (Example 1)

Outputs from the fifth digit to the end —> 6206
+CarriageReturn +LineFeed

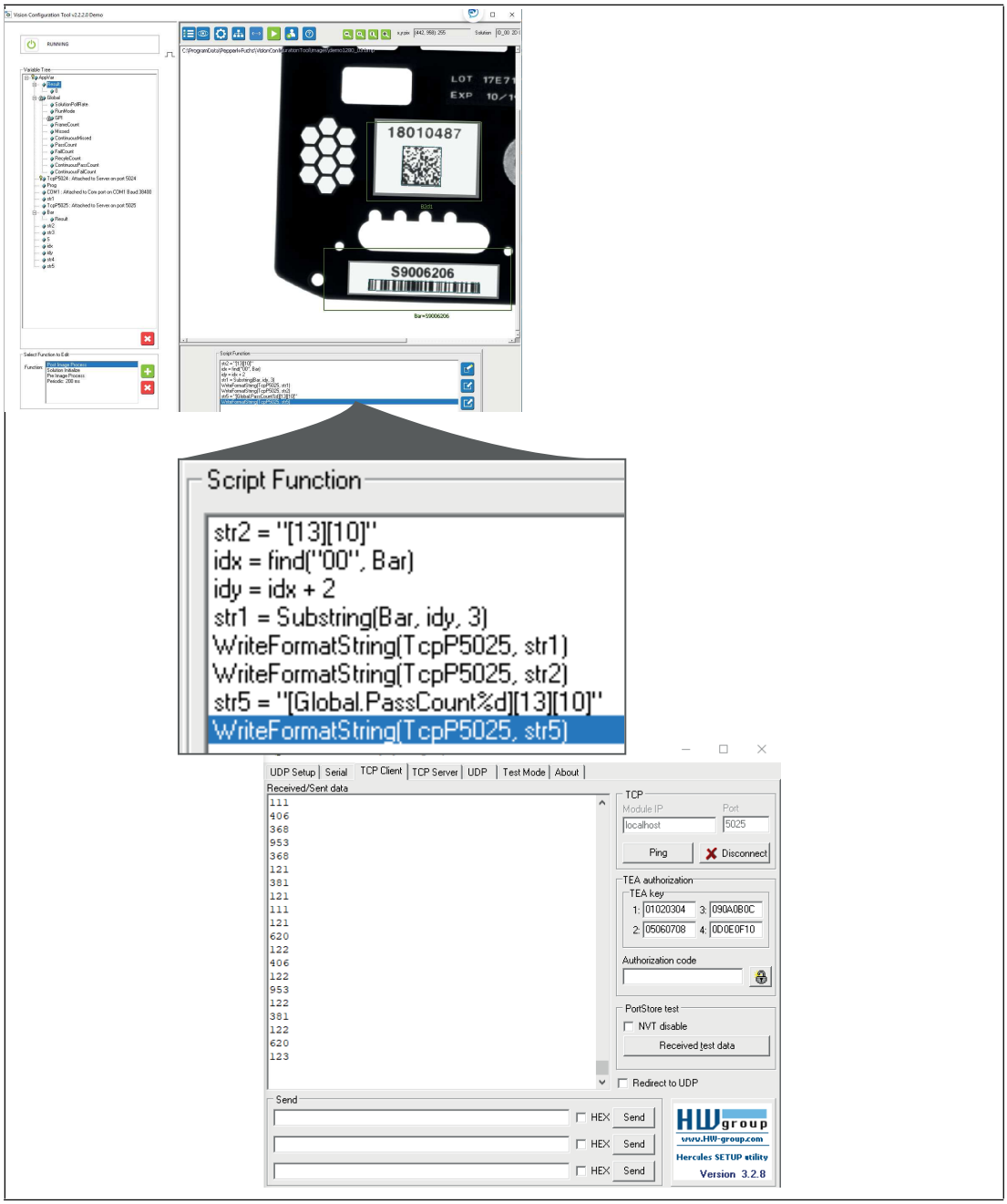


str1 = Substring(Bar,4,0)
str2 = "[13][10]"
WriteFormatString(TcpP5025,str1)
WriteFormatString(TcpP5025,str2)

Substring via TCP/IP (Example 2)

Searches for "00", outputs three digits —> 620

Outputs the good counter —> 123



str2 = "[13][10]"

idx = find("00",Bar)

idy = idx + 2

str1 = Substring(Bar,idy,3)

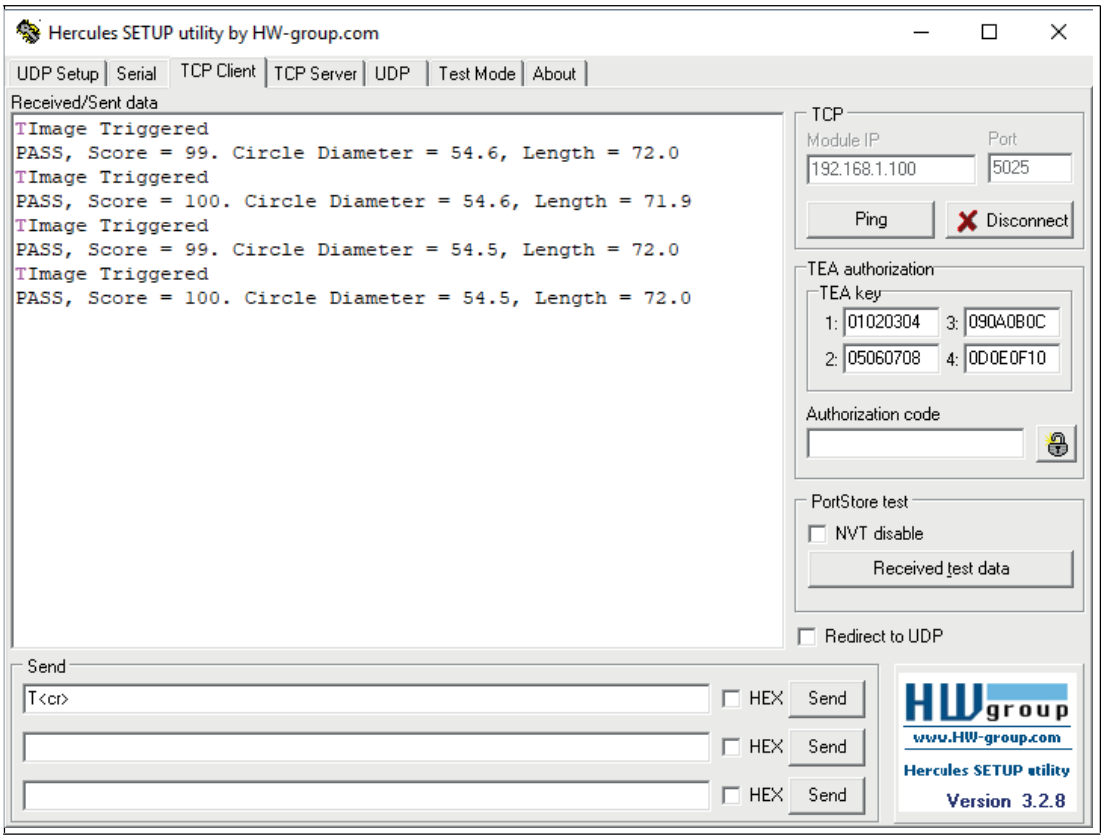
WriteFormatString(TcpP5025,str1)

WriteFormatString(TcpP5025,str2)

str5 = "[Global.PassCount%d][13][10]"

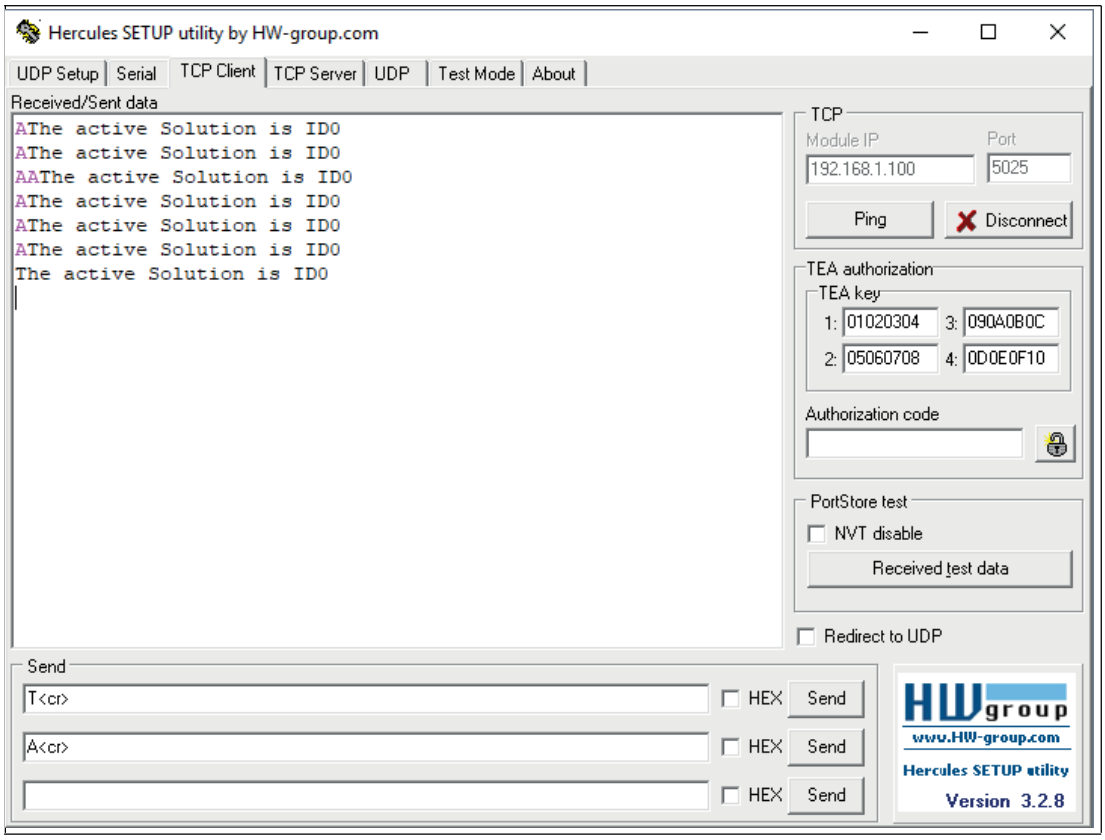
WriteFormatString(TcpP5025,str5)

TCP/IP: Triggering Image Capture—"T" Command




```
ReadBuffer = ReadString( TcpP5025 , 13 ) // 13 (<cr>) is the end character
if(ReadBuffer!= "") // if NOT an empty data string
  CommandString= ReadBuffer
  Counter = Counter + 1
  CommandCharacter =Substring(CommandString, 0, 1) // Trigger command
   if(CommandCharacter= "T") // "T" trigger command
    trigger()
    WriteFormatString (TcpP5025, "Image triggered[10][13]") // Optional feedback for triggers
   endif
endif
```

TCP/IP: Retrieving Job ID—"A" Command

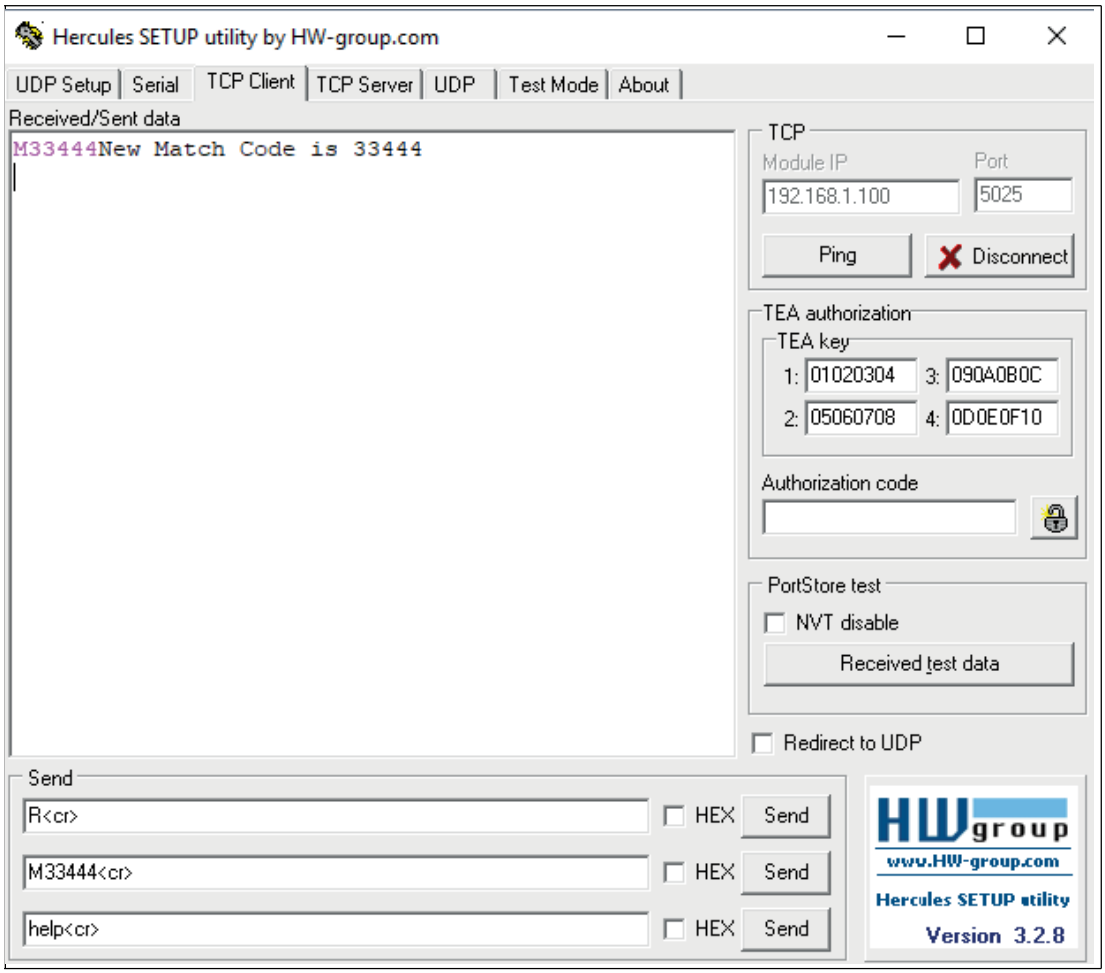



```
ReadBuffer = ReadString( TcpP5025 , 13 ) // 13 (<cr>) is the end character
if(ReadBuffer!= "") // if NOT an empty data string
  CommandString= ReadBuffer
  Counter = Counter + 1
  CommandCharacter =Substring(CommandString, 0, 1) // Trigger command
   if(CommandCharacter= "A") // Letter "A" outputs the active job
     ActiveSolution = GetSolutionID( )
     WriteFormatString(TcpP5025, "The active Solution is ID[ActiveSolution%s][10][13]" )
   endif
endif
```

TCP/IP: Changing Job ID—"S#" Command

```
ReadBuffer = ReadString( TcpP5025 , 13 ) // 13 (<cr>) is the end character
if(ReadBuffer!= "") // if NOT an empty data string
  CommandString= ReadBuffer
  Counter = Counter + 1
  CommandCharacter =Substring(CommandString, 0, 1) // Trigger command
  JobNumber = Substring(CommandString, 1, 1)
   if(CommandCharacter= "S" AND INT(JobNumber) >0 AND INT(JobNumber)<9)
     JobNumber = INT(JobNumber)
     // Conversion of string to INT
     ChangeSolution(JobNumber)
     // Switch to the specified job
     WriteFormatString( TcpP5025, "Changed to Solution ID[JobNumber%s][10][13]" )
   endif
endif
```

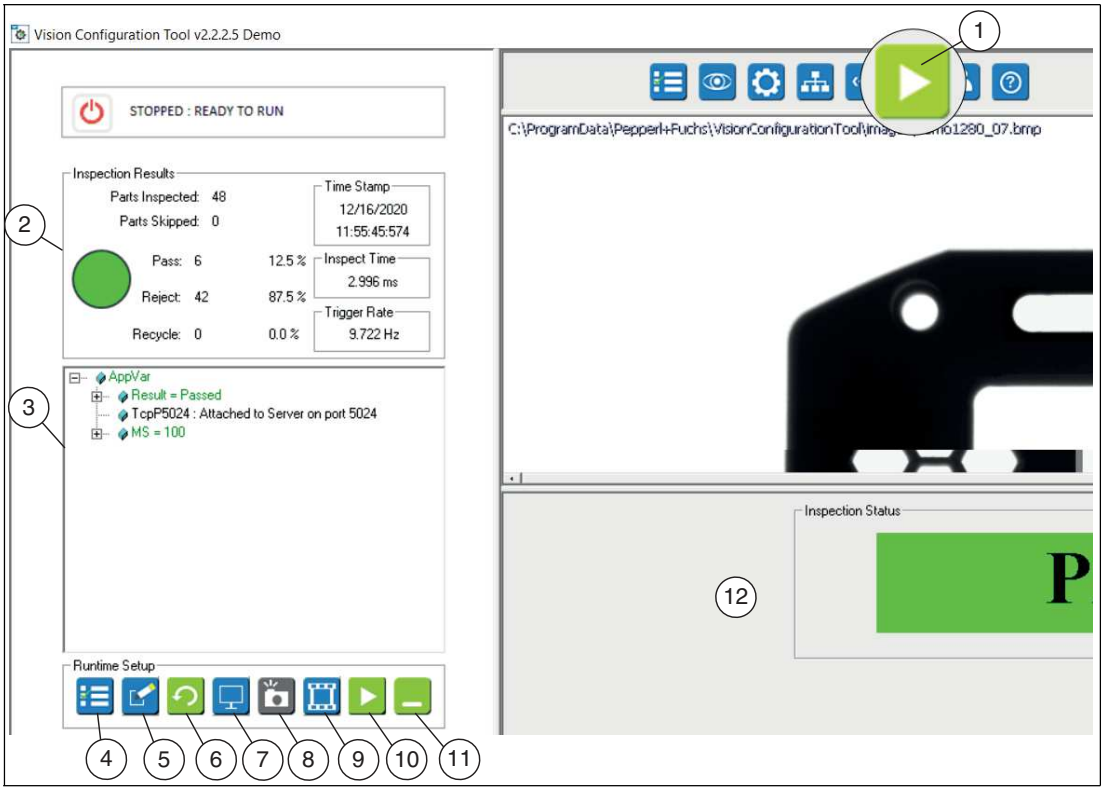
TCP/IP: Match Code Sent from the PLC M##### <cr>




```
ReadBuffer = ReadString( TcpP5025 , 13 ) // 13 (<cr>) is the end character
if(ReadBuffer!= "") // if NOT an empty data string
  CommandString= ReadBuffer
  Counter = Counter + 1
  CommandCharacter =Substring(CommandString, 0, 1) // Trigger command
  MatchCode = Substring(CommandString, 1, 0)
if(CommandCharacter="M" AND MatchCode!= 0 )
  SetMatchString (bar, MatchCode) // SetMatchString(measurementVar, perfectMatch) sets a
measurementVar string to a new perfectMatch value. For measurements that return a string
(barcode, 2-D barcode, OCR), a new "perfect" value is set.
 WriteFormatString( TcpP5025, "New Match Code is [Matchcode%s][10][13]")
 endif
endif
```

7.6 Application Test

The "Run solution" button (1) in the navigation bar takes you to the main menu of the "Application Test." Here you can check whether your tools are working as expected.



Item	Menu	Description
1	Run solution	"Run solution" main menu
2	Inspection Results	The inspection results are displayed in this field. • **Parts Inspected**: Total number of parts or images processed. This should be equal to the sum of "Pass, Reject, and Recycle." • **Parts Skipped**: Number of unprocessed parts or images if the trigger or internal timer is too fast to process all images. VOS is internally buffered, so you can take a new image while processing the previous image. Skipped parts are classified as failed. • **Pass, Reject, Recycle**: The (total) counter values for all camera images or for all parts (or since clicking "Reset Statistics") and the counter values for the current or selected camera. In some cases, the counter values of the camera images may be greater than the counter values for all cameras. • **Inspect Time**: The total time spent locating (finding) and measuring (testing or checking) the part.
3	AppVar	Application variables, measurements taken and logged, and other program variables.
4	Select Solution	Use this button to open a saved job. This button changes to "Return to Monitor" when you select a job.
5	Edit Tolerances	This displays a table containing all variables and measurements in the status area. You can change the measurement tolerances or adjust the size and position of the tool search area. This button changes to "Return to Monitor" when you edit the tolerances.

2021-07

Item	Menu	Description
6	Reset Statistics	Resets the "Pass", "Reject", and "Recycled" counters on this display to 0. If your test runs faster than one part per second, you may not see the values go to 0. But you will see that they are changing to smaller numbers.
7	Setup Display	Select what appears in the image area and status window. This button changes to "Return to Monitor" when you configure the display.
8	Manual Trigger	Triggers the camera shot when "Inspection Trigger" is selected in the "Sensor Settings" menu. This button is disabled if "Inspection Trigger" is not enabled.
9	History Recall	History logging for tested parts. This button is not active if you have disabled "History Recall" in the "History Recall Setup" section. This button changes to "Return to Monitor" when you display history logs.
10	Start	Restarts the inspection if "Stop" was clicked.
11	Minimize Maximize	Hides the status field so that only the "Image area" and "Settings" fields are displayed. The settings field appears at the bottom of the screen. A "Maximize" button is added to the setup field. Opens the status field again and returns the setup field to its normal position on the left-hand side.
12	Configuration and status window	Further properties of the selected menus are displayed in the configuration and status window.

Edit Tolerances

You can use the "Edit Tolerances" function in the configuration and status field to adjust the tolerance values (temporarily or permanently) in the table. In addition, you can move or resize the tool's search field in the image area when "Edit Tolerances" is open. Changes affect the values in the Setup Tools windows.

The new value takes effect when you press Enter, or click anywhere in the display outside the edited cell.

There are no "Cancel" or "Undo" buttons. Close or reload the project—without saving—to undo any changes.

The new settings are used while the project is running. You can restore your previous settings by reloading the previously saved project.

The new settings will be lost if you reload the project, load another project, or exit the project without saving.

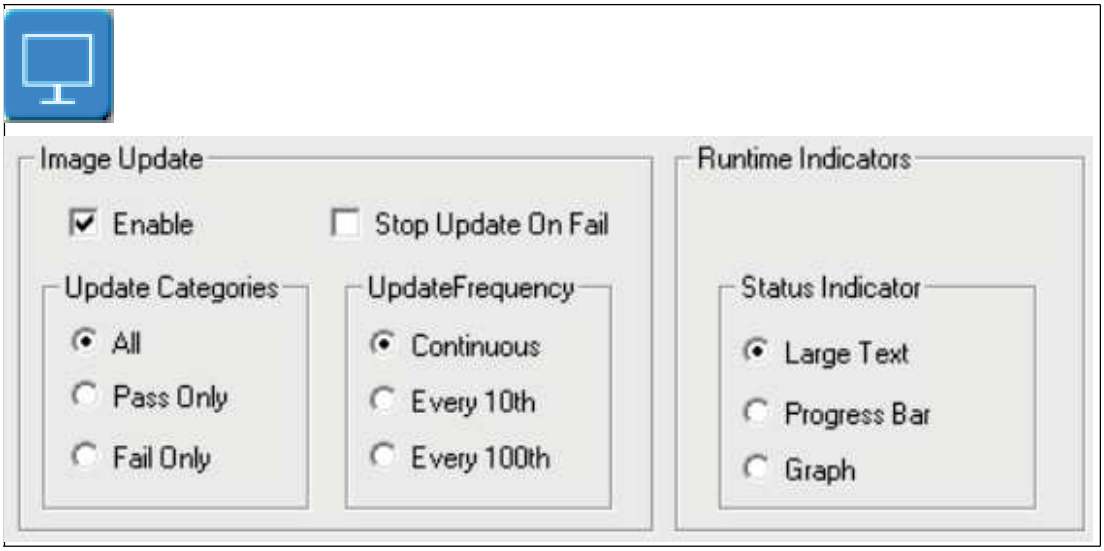
To retain your new settings, save the job files (under a new job number or overwrite the same job number) to keep these settings for reloading later.

Setup Display

You can use the "Setup Display" function to set how often the sensor updates the display on the PC during the application test.

Note

The display options are grayed out when image logging is enabled.



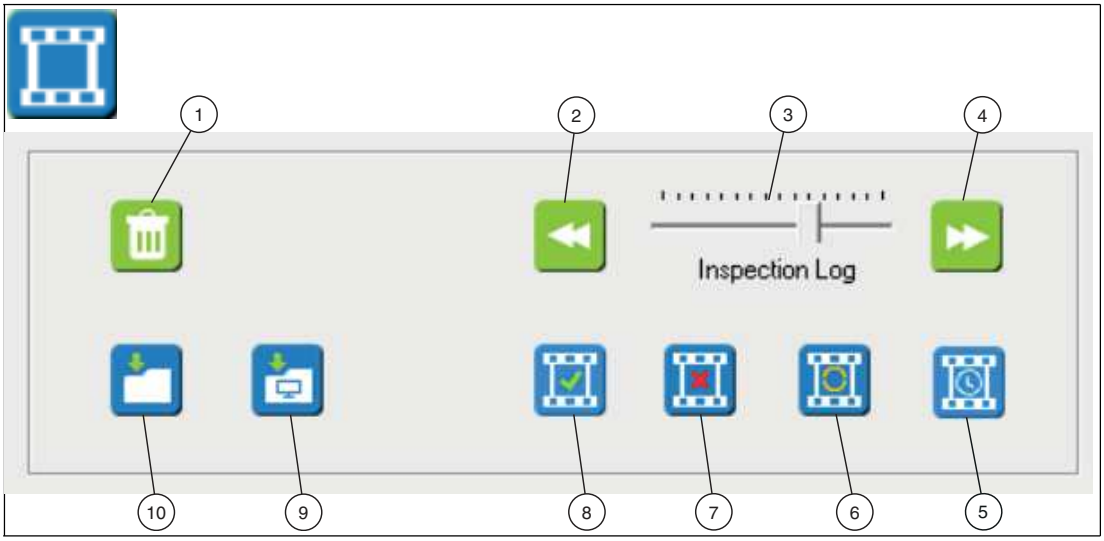
Function	Option	Description
Image Update		Controls the display of the image area during the inspection runtime:
	Enable	Enables update of the display. Uncheck the "Enable" check box to disable all updates or disable the display in the image area.
	Stop Update on Failure	Select the "Stop Update on Failure" check box to freeze the display in the event of an error. The sensor continues to run and tests parts. Only the display is affected. The "Reset Statistics" button changes to "Reset Display." Click "Reset Display" to resume a live display.
Update Categories		Select which images or results cause the display to update:
	All	All tested parts: Pass, Recycle, Fail
	Pass Only	The display is updated for "Pass" parts.
	Fail Only	The display is updated for "Fail" parts.
Update Frequency		Selects how often the display is updated.
	Continuous	Displays all parts in the selected category.
	Every 10th	Display every 10th part in the selected category.
	Every 100th	Display every 100th part in the selected category.

2021-07

Function	Option	Description
Runtime Indicators		Controls the "Monitor" display in the configuration and status window during the inspection runtime.
	Large Text	Displays "Pass, "Recycle", or "Fail" in large font with a colored background (green, yellow, red).
	Progress Bar	Displays a horizontal bar with red, yellow, and green colored areas corresponding to the percentages of "Failed", "Recycled", or "Passed" parts.
	Graph	Displays a bar chart with three bars in red, yellow, and green, corresponding to the number (count) of "Failed", "Recycled", or "Passed" parts.

History Recall

You can use the "History Recall" function to display and save the recorded history of the measurement results.



Item	Function	Description
1	Clear History	Clears (deletes) all images and data from the history log.
2, 3, and 4	Inspection Log	Use the double-arrow keys to scroll through the records. The sensor saves the last 20 records for the resolution of 640 x 480 pixels and two recordings for the resolution of 1280 x 960 pixels.
5	Time Log	Displays all parts: passed, recycled, and failed.
6	Recycle Log	Displays the history of the recycled parts.
7	Fail Log	Displays the history of the "failed" parts.
8	Pass Log	Displays the history of the "passed" parts.
9	Save Screen	Saves the entire screen of the PC display, all three window areas.
10	Save Image	Saves the image of the object without graphics or measurement tools.

7.7 User Administration

To prevent others from changing your jobs, you can set a password for your sensor. Without entering a password, other users can only access the "Monitor" screen to display the "Run" screen.

Creating User Accounts

1. Enter the administrator password.
2. Click "Log In."
3. Check "Enable."
4. Click "Add" to add a new account.
5. Enter your user name and the login name. Assign a password.

Note

The user name is limited to a maximum of 31 characters.

The user's full name is limited to a maximum of 63 characters.

Passwords are limited to a minimum of 6 characters and a maximum of 15 characters.

6. Set the options for the user account.
7. When you are finished, click "Log out."

Note

If you leave "Enable" enabled and log out, all users will need to log back in to use the sensor menus. If you clear the "Enable" check box, a password is not required and access is not restricted.

Enabling or Disabling Access Permission

1. Click the "User administration" button in the navigation bar.
2. Log in as an administrator to make changes. Enter the administrator password and click "Log In."
3. You can check "Auto LogOff after" and set the time taken until it logs off automatically.
4. Click the "Enable" check box if you want to enable user accounts and require a password each time you attempt to access the sensor. Clear the check box if you do not want to require logins. Anyone can see the "Monitor" window without logging in.
5. Click "Log Out" when you are finished.

Note

If you leave "Enable" enabled and log out, all users will need to log back in to use the sensor menus. If you clear the "Enable" check box, a password is not required and access is not restricted.

8 Appendix

8.1 System Settings in Windows®

The application interface is accessed via the Ethernet connection to a PC running Microsoft® Internet Explorer 6 or higher.

Configuring the PC for Sensor Access

The paths to some tools depend on your Windows® configuration and display settings. Some elements may appear in the left or right window range of the menus.

Setting the System Performance in Windows® 10

Use your PC's performance options to customize the visual effects of Windows® 10 or Windows® 7. This setting increases the overall performance of your PC as well as the performance of the sensor applications. If you do not reduce or disable the performance options on your PC, it may cause malfunctions of the "VOS Configuration Tool" software.

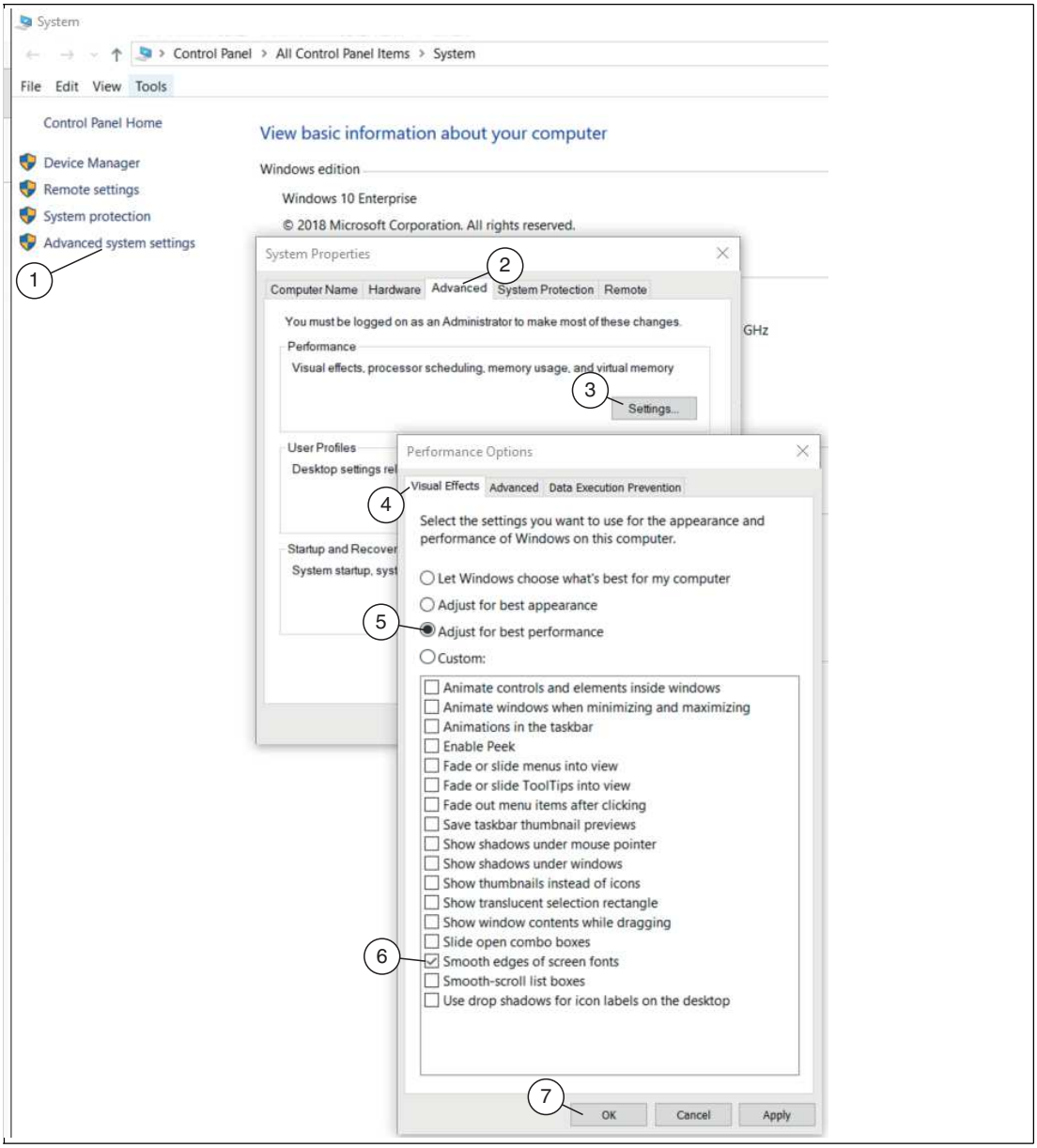


Figure 8.1 Performance options

1. Open the system control panel. Click on "System".

   ↳ The start page of the system control panel opens.

2. From the start page of the system control panel, select the "Advanced System Settings" menu (1).

   ↳ The "System Properties" window opens.

3. In the "System Properties" window, select the "Advanced" tab (2).

4. On the "Advanced" tab, select the "Settings" function (3).

   ↳ The "Performance Options" window opens.

5. In the "Performance Options" window, select the "Visual Effects" tab (4).

6. On the "Visual Effects" tab, select the "Adjust for best performance" setting (5).

7. Enable the following settings: "Refine edges of screen font" (6) and "Use visual styles for windows and buttons."

2021-07

↳ The performance setting is changed to "Custom".

Note

Some systems may not have the "Use visual styles for windows and buttons" setting.

8. Click "OK" (7) to confirm your selection.

↳ The design is changed to resemble Windows® Classic.

Disabling or Customizing User Account Control in Windows® 10

User Account Control (UAC) is a security tool in Windows® 7 and Windows® 10. UAC controls the assignment of rights to individual programs. If a program wants to make changes to your system, UAC responds and you must actively agree to the change. This also prevents the system from loading and running the **"Vision Configuration Tool"** software. As soon as you start the **"Vision Configuration Tool"**, a warning message is issued by the system. Depending on the UAC level, you may need to enter an administrator password.
If your PC is in a closed and secure factory environment that is not connected to the Internet and an administrator account is also used, it is usually safe to switch off UAC.
If your PC is connected to an office network and the Internet, you should use the default setting or a higher security setting. The software should be manually approved when the prompts appear.
The procedure for customizing the UAC settings or disabling UAC completely is described below.

Figure 8.2 Settings

1. Press the [Windows®] + [X] key combination and then click on the "Settings" menu item (1).

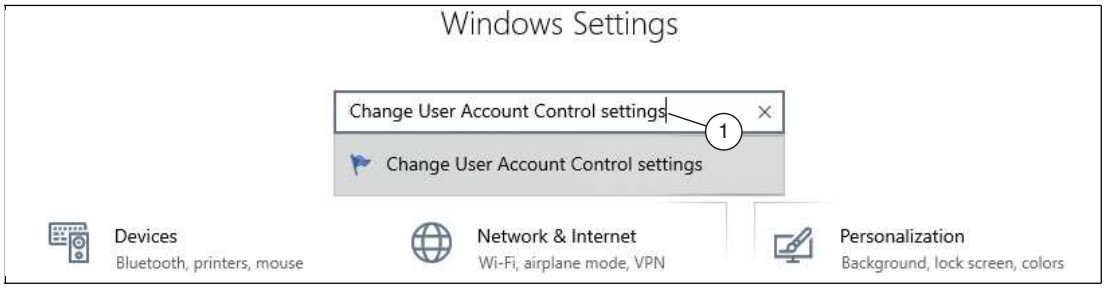


Figure 8.3 Search

2. Type "Change user account control settings" (1) in the upper search bar and confirm with [Enter].

↳ The "User account control settings" window opens.

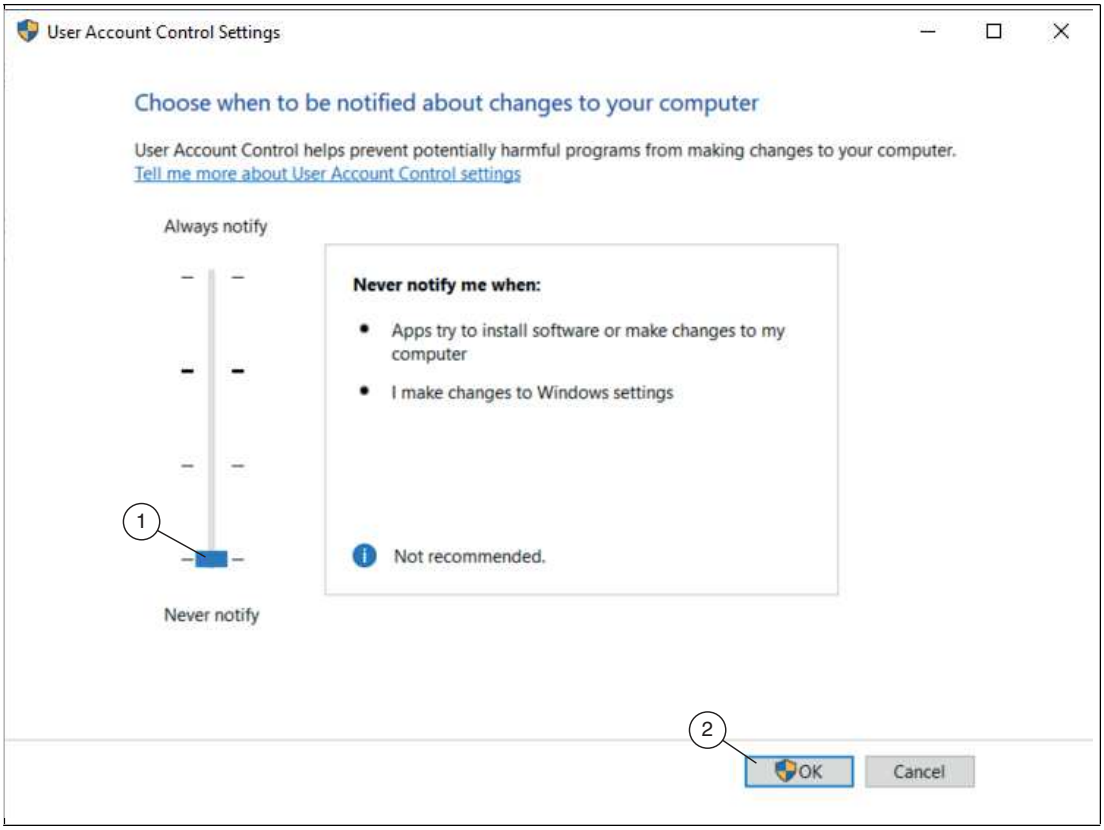


Figure 8.4 Disabling UAC

3. Use the **slider** (1) on the left to customize notifications from user account control. If the **slider is at the bottom**, **UAC is disabled** and you are no longer notified when a program changes settings on your system. If the slider is at the top, you will always be notified. Save your settings with "OK" (2).

Switching Windows® Firewall On or Off

Windows® Firewall protects your system from unauthorized access from the Internet. For your own security, the firewall should always be enabled. However, it is sometimes necessary to disable it. The sensor communicates with the PC via the network connection. Windows® Firewall blocks this communication. The easiest way to use the sensor software is to turn off the firewall. If you need to connect to an office network or the Internet, you can turn on the firewall again.

1. Go to the Windows® 10 Start menu at the bottom left of the screen. In the search, type "Windows® Defender Firewall" and select the appropriate entry.

   ↳ The Windows® Firewall settings will open.

   Note

   If you see a message that the service is not running, click on "No" and do not start the firewall. Continue downloading the **Nexus** software.

2. In the firewall options, select "Turn Windows Defender Firewall on or off".

   ↳ The "Customize Settings" window opens with the default settings for your Windows® Firewall.

3. In the settings, select the two "Disable Windows Defender Firewall" options and confirm your entry with "OK".

2021-07

Note

To turn the firewall back on, select "Enable Windows Defender Firewall" for both options.

↳ Windows® Firewall is disabled, and network security for Windows® 10 is slightly eased.

Note

You also have the option to leave the firewall enabled. In this case, when you run the application, you must approve each security-critical step because the firewall reports it as "potentially unsafe".

Windows® Firewall—Adding an Exception

1. Go to the Windows® 10 Start menu at the bottom left of the screen. In the search, type "Windows Defender Firewall" and select the appropriate entry.

   ↳ The Windows® Firewall settings will open.
2. In the new window, select the option "Allow a program or feature through Windows Firewall".
3. Click on "Change settings", then click on "Allow other app..."
4. If the **"VOS Configuration Tool"** software is already in the list, simply tick the box for the rule.

   Note

   If this is not the case, or the enabling still does not work, you must use the "Browse..." option to select the program.
5. Confirm with "OK". All open windows can be closed.

Changing the IP Address in Windows® 10

The following section describes how to customize the network connection settings of your Windows® PC for connection to your sensor.

1. Open the system control panel.
2. Click on "View network status and tasks".
3. Select "Change adapter settings" from the list on the left.
4. Right-click on the appropriate network and click on "Settings".
5. Then double-click on the entry "Internet protocol, version 4 (TC/IPv4)".

   ↳ The settings window for the TCP/IP protocol opens.
6. Select "Use the following IP address:" and enter the IP address of the device, but only the first three segments of the IP address. The last segment must be different from the IP address of the sensor. The new IP address must not already be in use elsewhere in the network.

   Note

   Contact your system administrator for further instructions. The subnet mask should be set to 255.255.255.0.

Your automation, our passion.

Explosion Protection

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

Industrial Sensors

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

Pepperl+Fuchs Quality
Download our latest policy here:

www.pepperl-fuchs.com/quality

PEPPERL+FUCHS

www.pepperl-fuchs.com

Printed in Germany / DOCT-7204