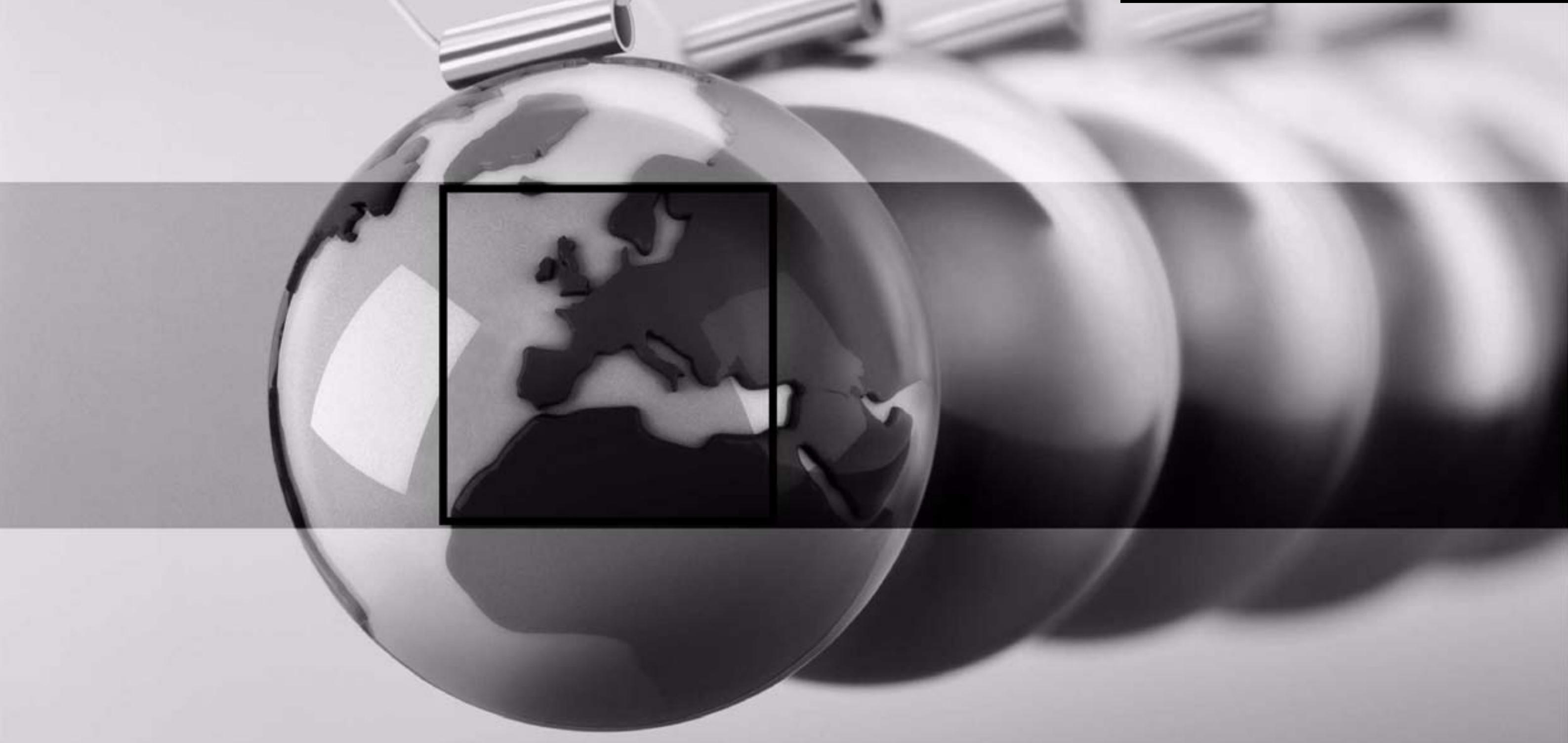
FACTORY AUTOMATION



MANUAL

OIT500-F113-B12-CB3

High-Temperature Identification System

CE

PEPPERL+FUCHS

SENSING YOUR NEEDS

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:

STOP

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

ℹ

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.

1.4 Registered Trademarks

Microsoft®, Windows®, Windows 7® are registered trademarks of Microsoft Corporation. All featured trademarks and company names are subject to the copyright of the respective companies.

2 Product Description

2.1 Functional Description

OIT500-F113-B12-CB3 (referred to as the OIT system from now on) uses infrared lighting and evaluates special code sheets that feature code patterns. For this purpose, the OIT system is fitted with a normal lens and an internal LED board for illuminating the code sheets. The infrared lighting guarantees optimal contrasts during a read operation, so that even soiled code sheets can be reliably identified. If the read result deteriorates during operation, the diagnostics will automatically provide an analysis. You can use CB1 code sheets and CB3 code sheets for the OIT system. Further technical details can be found in the datasheet for the OIT system.

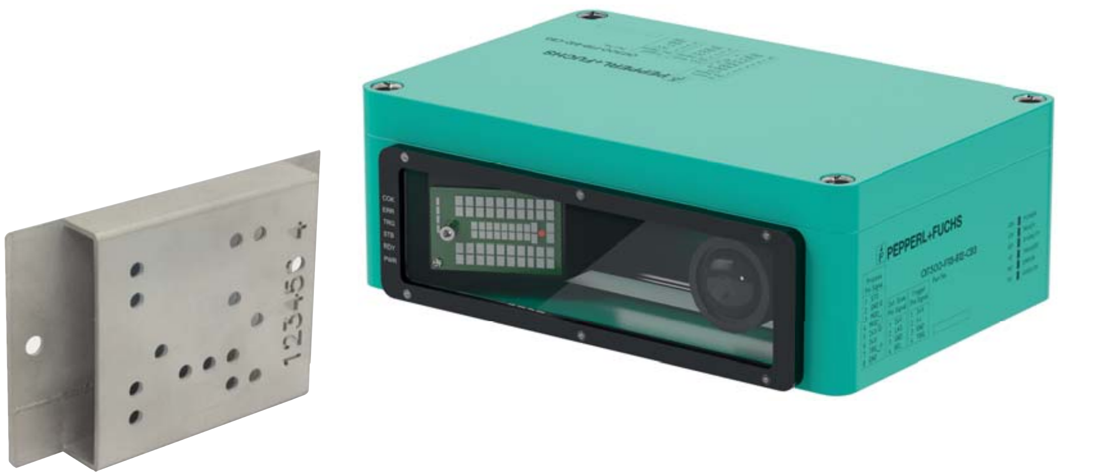


Figure 2.1 Code sheet and identification system

2.2 Application and Areas of Use

The OIT system is used for automated manufacturing processes in harsh ambient conditions where the use of read-only tags with electronic components is difficult or even impossible. Code sheets with a perforated matrix are used as read-only tags. These code sheets are designed for use at temperatures of up to 500 °C and can withstand high mechanical stress. The device is easy to mount and is then ready for use immediately with no teach-in required. Plug-in connections enable quick replacement of devices. The system is controlled via the Ethernet interface using simple sets of commands, making it easy to operate. The scratch-resistant quartz glass panel can be replaced if necessary.

The code pattern punched into the code sheets features an ID that is detected and processed by the OIT system. The read result is forwarded to a computer via the network interface. The ID can contain information such as a type designation, paint color, or similar information for the object on which the code sheet is mounted. To detect the code sheets, the OIT system is mounted directly in the plant.

The device is only approved for appropriate and intended use. Ignoring these instructions will void any warranty and absolve the manufacturer from any liability.

Use the device only within the specified ambient and operating conditions.

Protection of the personnel and the plant is not ensured if the device is not used according to its intended use.

2.3 Displays and Operating Elements

The seven LEDs mounted on the lighting unit can be used to read different information.

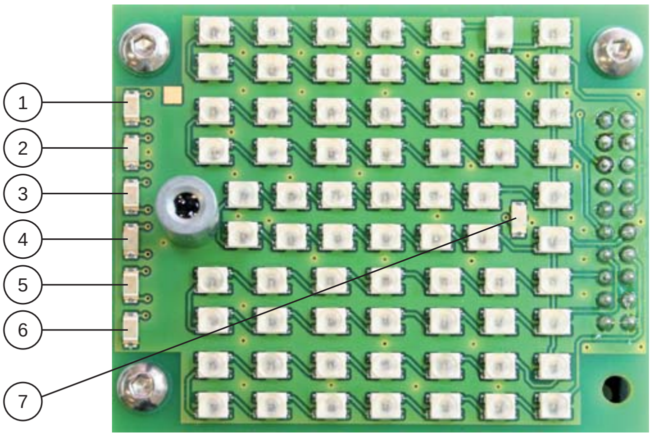


Figure 2.2 Lighting unit of the OIT system

Item	Designation	Function
1	**Code OK**	Reading successful. Flashes for approx. 5 seconds when the parameters have been applied successfully.
2	**Error**	Lights up red when a read error occurs.
3	**Trigger**	Lights up yellow when a connected trigger sensor is activated.
4	**Stability of image information**	Lights up red if the captured image could be read but increasing deterioration of the ambient conditions may result in a reading error. This can occur, for example, in the following situations: if the camera image is too light/dark, if the contrast is too low, if there are too many structures in the image, if the code sheet is just within the read distance, or if the code sheet is only just within the image area.
5	**Ready for operation**	The LED flashes for approx. 30 seconds while it is starting up. During this time, no network connection can be established with the OIT. The LED lights up as soon as the OIT is ready for operation.
6	**Power**	Lights up green when the OIT system is supplied with power.
7	**Lighting control**	Lights up red when infrared illumination is active.

2.4 Interfaces and Connections

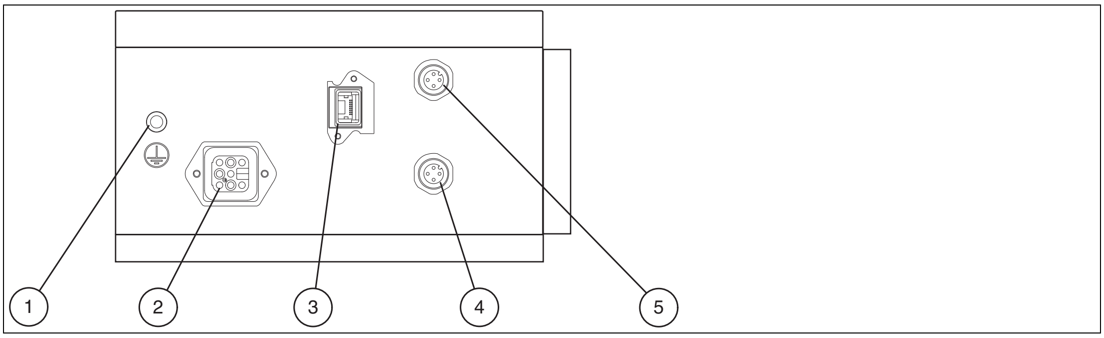


Figure 2.3 Connections

1. Ground connection
2. Power supply (Harting Han® 8D-F plug)
3. Network connection (RJ45 network socket, IP65)
4. Trigger connection (4-pin M12 socket)
5. Do not use connection (reserved)

Pinout for Power Supply

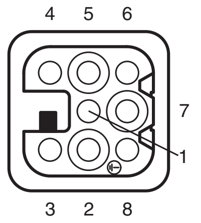


Figure 2.4 Pinout for power supply

1. Do not use connection (reserved)
2. Do not use connection (reserved)
3. Do not use connection (reserved)
4. Do not use connection (reserved)
5. Do not use connection (reserved)
6. 24 V power supply for device (24 V)
7. Do not use connection (reserved)
8. Device ground (GND)

Pinout for Network Connection

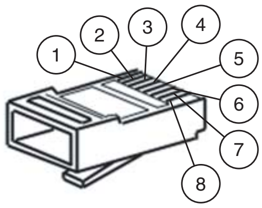


Figure 2.5 Pinout for Network Connection

1. TD+
2. TD-
3. RD+
4. Not used
5. Not used
6. RD-
7. Not used
8. Not used

Pinout for Trigger Connection

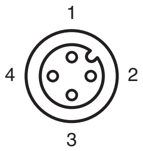


Figure 2.6 Pinout for trigger connection

1. 24 V power supply for trigger sensor
2. Not used
3. Ground (GND)
4. Trigger (TRIG)

2.5 Scope of Delivery

Check the packaging and contents for damage.

Check if you have received every item and if the items received are the ones you ordered.

Included in delivery:

- OIT500-F113-B12-CB3
- Mounting base (pre-assembled on the housing)

Always store and transport the device in the original packaging.

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

2.6 Accessories

Various accessories are available.

2.6.1 Power Supply

Designation	Description
V8HAN-G	Field-attachable female connector
V8HAN-G-10M-PUR-ABG	Single-ended female cordset, shielded PUR cable, 10 m

2.6.2 Connection Cable for Trigger Sensors and External Lighting

Use the following connection cable to connect a trigger sensor or external lighting.

M12 cordset

	Material	Length	M12 plug, straight 	M12 plug, angled
4-pin M12 socket, straight 	PUR	2 m	V1-G-2M-PUR-V1-G	V1-G-2M-PUR-V1-W
		5 m	V1-G-5M-PUR-V1-G	V1-G-5M-PUR-V1-W
		10 m	V1-G-10M-PUR-V1-G	V1-G-10M-PUR-V1-W
4-pin M12 socket, straight 	PVC	2 m	V1-G-2M-PVC-V1-G	V1-G-2M-PVC-V1-W
		5 m	V1-G-5M-PVC-V1-G	V1-G-5M-PVC-V1-W
		10 m	V1-G-10M-PVC-V1-G	V1-G-10M-PVC-V1-W
4-pin M12 socket, angled 	PUR	2 m	V1-W-2M-PUR-V1-G	On request
		5 m	V1-W-5M-PUR-V1-G	On request
		10 m	V1-W-10M-PUR-V1-G	On request

Field-attachable M12 connectors

Order designation	Description	mm^2	Cable dia.
V1-G	4-pin M12 socket, straight	max. 0.75	4 ... 6 mm
V1-G PG9	4-pin M12 socket, straight	max. 0.75	6 ... 8 mm
V1-W	4-pin M12 socket, angled	max. 0.75	4 ... 6 mm
V1-WV2A	4-pin M12 stainless steel socket, angled	max. 0.75	4 ... 6 mm
V1S-G	4-pin M12 plug, straight	max. 0.75	4 ... 6 mm
V1S-W	4-pin M12 plug, angled	max. 0.75	4 ... 6 mm
V1-G-Q2	4-pin M12 socket, straight	0.34 ... 0.75	7 ... 7.5 mm
V1S-G-Q2	4-pin M12 plug, straight	0.34 ... 0.75	7 ... 7.5 mm
V1-G-Q3	4-pin M12 socket, straight	0.14 ... 0.34	4 ... 6 mm
V1S-G-Q3	4-pin M12 plug, straight	0.14 ... 0.34	4 ... 6 mm

Other lengths on request.

2.6.3 Network Cable

Note!

Network Connection with Degree of Protection IP65

The network connection on the OIT has degree of protection IP65. To ensure the IP65 degree of protection, only use a network connector with degree of protection IP65. Do not use a standard RJ45 network connector.

The OIT system is connected to the network via an RJ45 network connector.

Designation	Description
V45-G	Field-attachable RJ45 network connector
V45-GP	Field-attachable push-pull RJ45 network connector (IP65)
V45-GP-10M-PUR-ABG-V45-G	10 m cordset, push-pull RJ45 network connector (IP65)/RJ45 network connector

2.6.4 Code Sheets

The code sheet serves as a read/write tag with a punched hole pattern for use at high temperatures. The robust code sheet is suitable for use in environments up to 500 °C and remains legible even in the case of heavy contamination. You can use CB1 code sheets and CB3 code sheets for the OIT system.

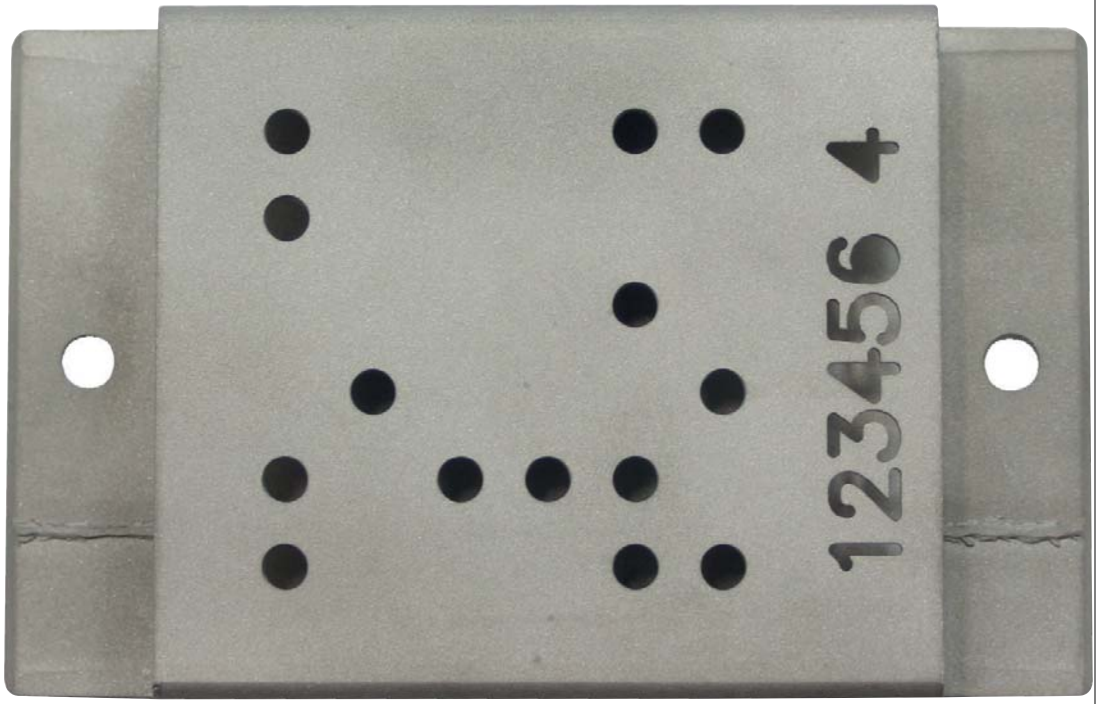


Figure 2.7 CB1 code sheet

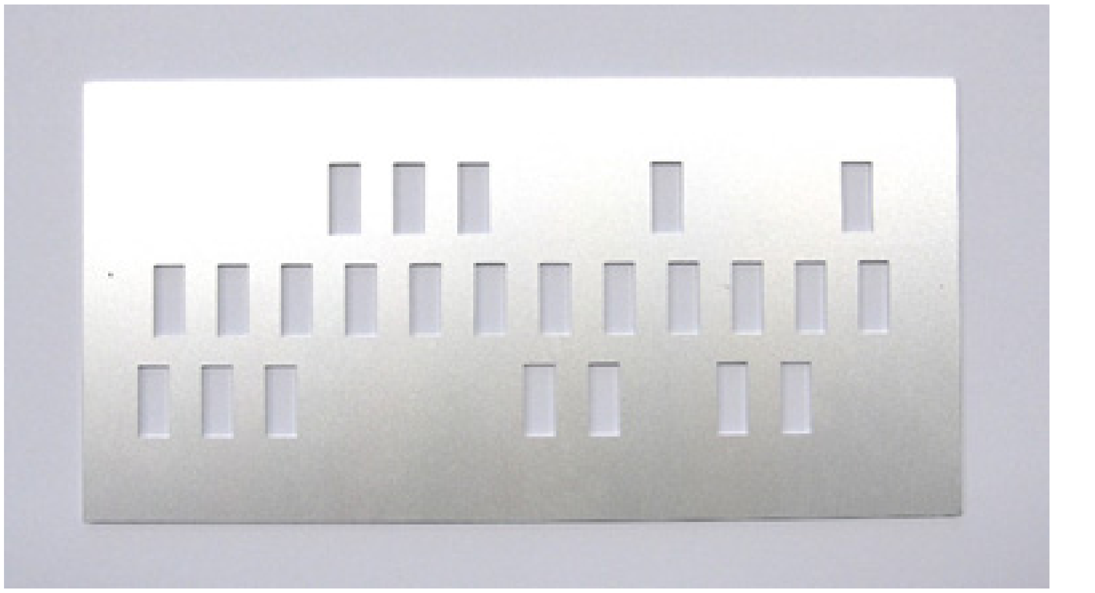


Figure 2.8 CB3 code sheet

Note!

Ordering Code Sheets

The CB1 code sheet can be ordered from Pepperl+Fuchs.

The CB3 code sheet is still on the market, but cannot be ordered from Pepperl+Fuchs.

Note!

Contamination of the Code Sheet

Protect the code sheet from excess dirt. The code sheet can be cleaned using aggressive or abrasive cleaning agents.

Ensure that no other markings are made on the code sheet, as this can affect the reading.

3 Mounting and Installation

3.1 OIT System Overview

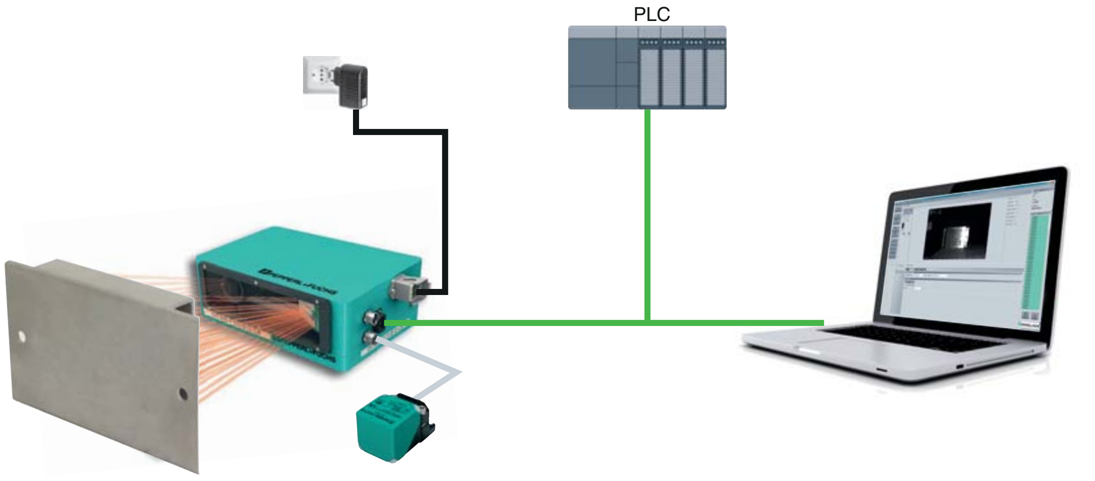
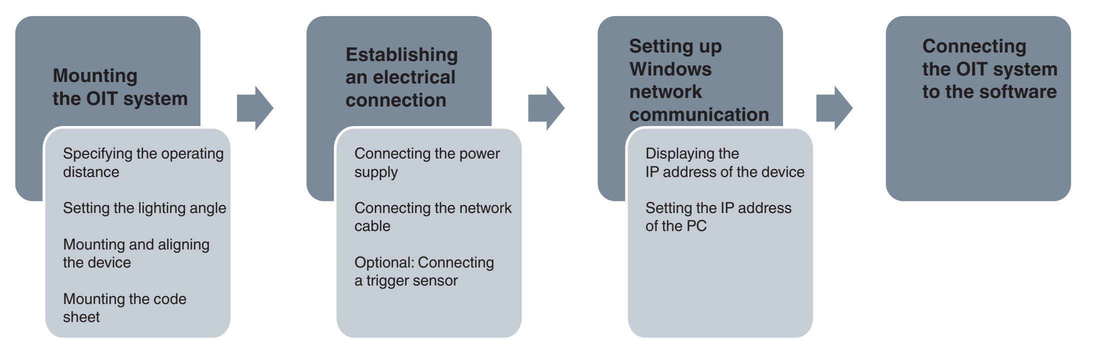


Figure 3.1 OIT system overview

The installation of the OIT system involves just a few steps. In addition to the Ethernet TCP/IP interface, the device has a trigger input for an optional trigger sensor (NBB20-L2-A2-V1 inductive sensor shown for illustrative purposes).

The initial commissioning sequence is described in the following diagram. Each box indicates one of the following subsections:



Mounting the OIT system

- Specifying the operating distance
- Setting the lighting angle
- Mounting and aligning the device
- Mounting the code sheet

Establishing an electrical connection

- Connecting the power supply
- Connecting the network cable
- Optional: Connecting a trigger sensor

Setting up Windows network communication

- Displaying the IP address of the device
- Setting the IP address of the PC

Connecting the OIT system to the software

Figure 3.2 Schedule for initial commissioning

Note!

Before carrying out the initial commissioning of an OIT system, you must install the latest version of the **Vision Configurator** software on your PC/laptop. An up-to-date description of the Vision Configurator software can be found on our website at http://www.pepperl-fuchs.com.

3.2 Mounting the OIT System

Warning!

Danger to life due to defective mounting

Errors during mounting can cause life-threatening injuries and significant property damage.

- Ensure the installation is performed only by sufficiently trained and qualified personnel. Trained and qualified personnel have relevant experience in this area. They know and understand the rules and standards for the components and systems.
- Prior to mounting, ensure that the plant is de-energized.
- The device is relatively heavy. Handle the device carefully.

Note!

Avoid bright areas in the background when mounting the code plate. The code plate must be brighter than the background. If this is not the case, reading may be impossible.

3.2.1 Operating Distance

The operating distance is the distance from the glass pane of the OIT system to the code sheet. The OIT system is set to a reading distance of **380 mm** by default.

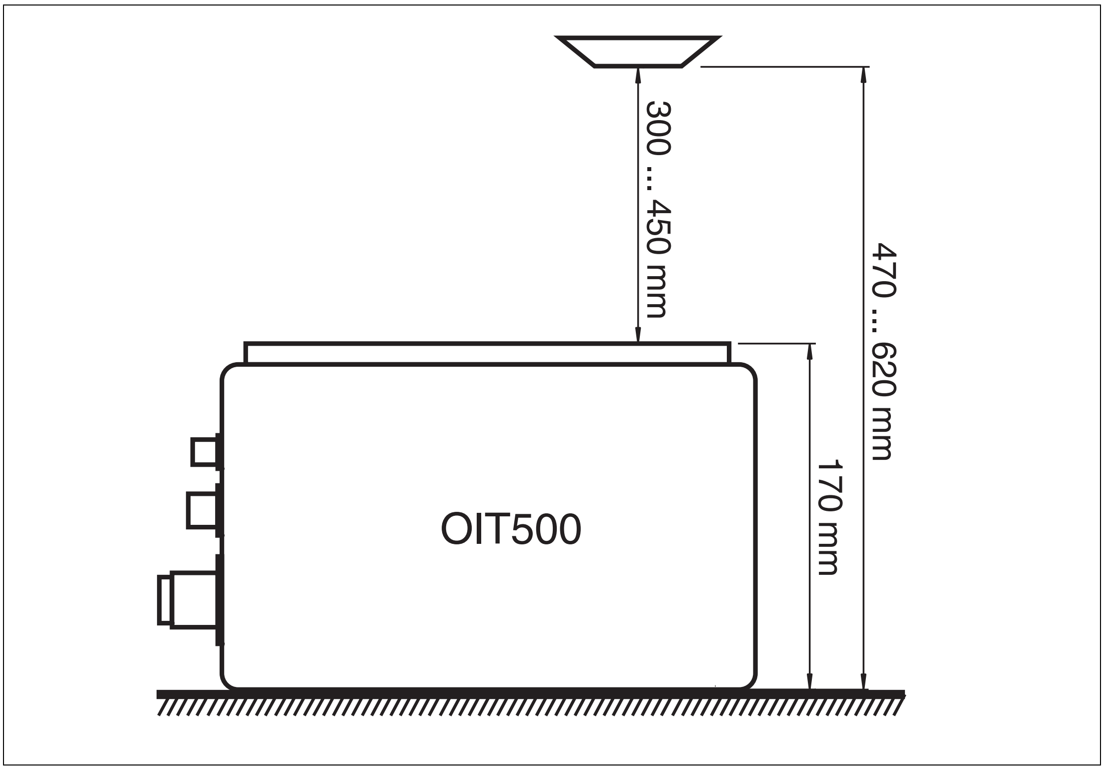


Figure 3.3 Operating distance (top view)

Permissible Code Sheet Displacement for CB1 Code Sheets

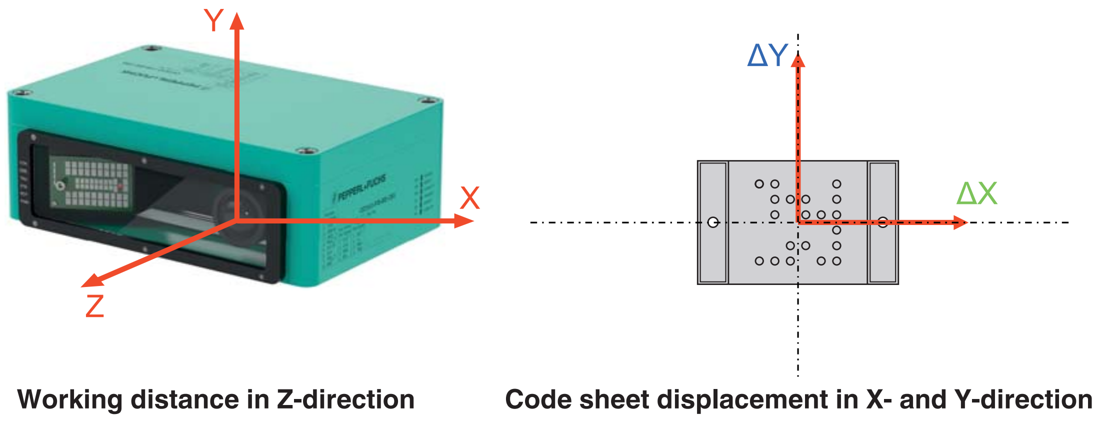

Y
X
Z
ΔY
ΔX
Working distance in Z-direction
Code sheet displacement in X- and Y-direction

Figure 3.4 X/Y direction for the displacement of **OIC-xxxx-CB1** Code sheet.

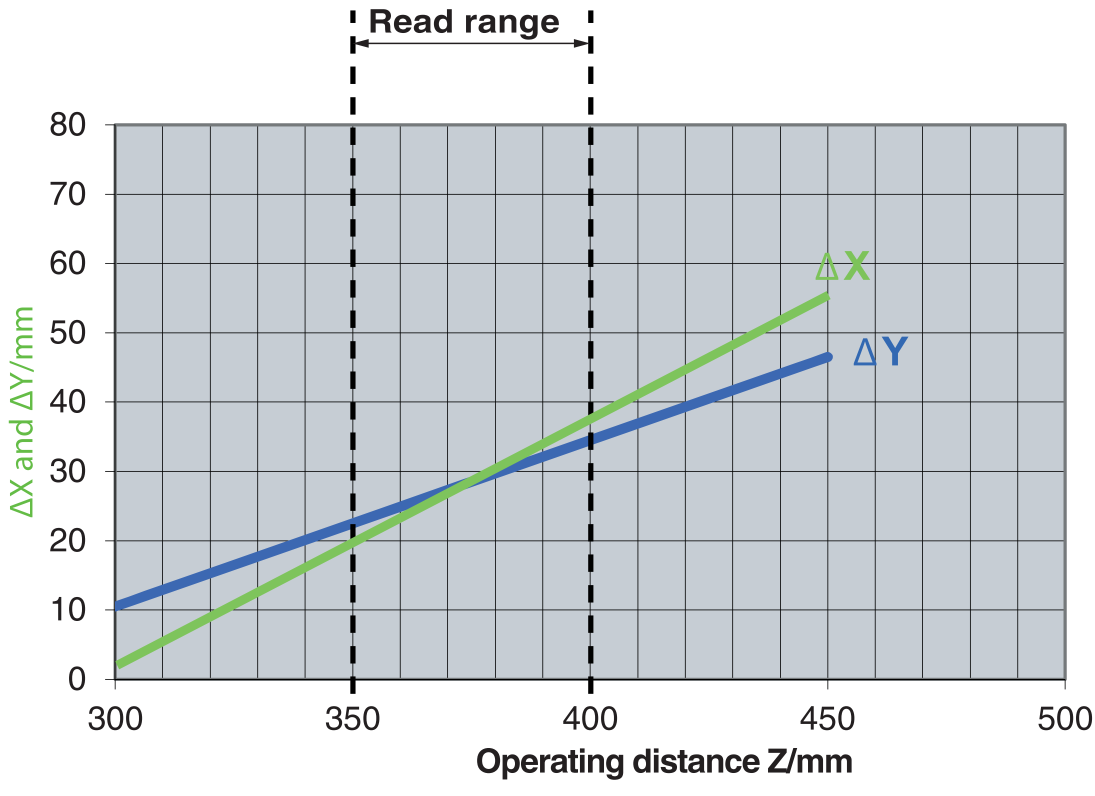


Figure 3.5 Permissible displacement of read-only tag ± ΔX and ± ΔY depending on reading distance Z

A reading window in the X and Y direction determines the permissible displacement of read-only tags. The graphic shows that as the operating distance increases, the possible code sheet displacement increases. The graphic is applicable provided that the code sheet is parallel with the OIT system and the center of the code sheet is shown on the optical axis of the lens.

Permissible Reading Window Displacement for CB1 Code Sheets

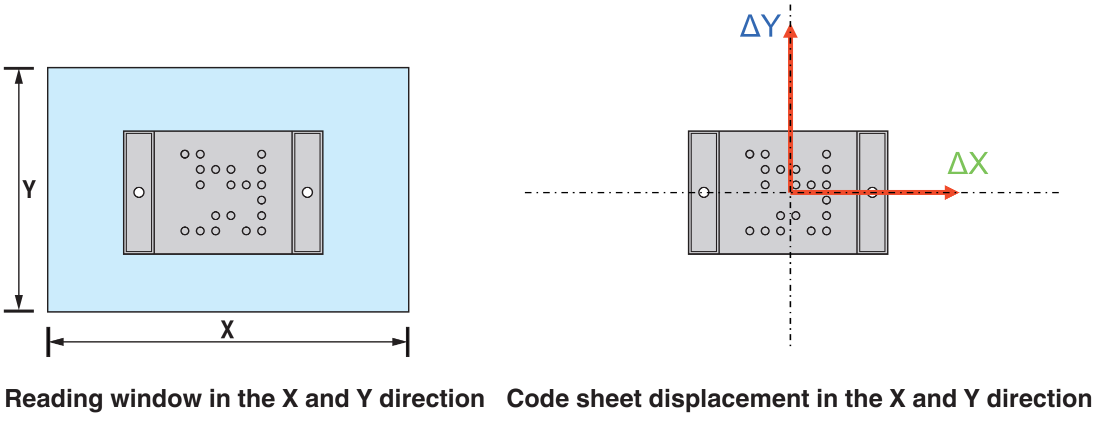
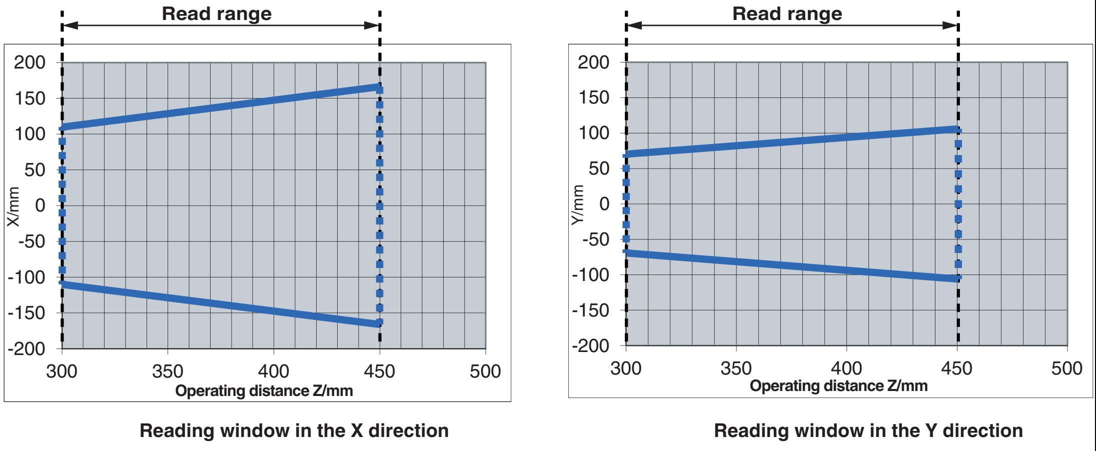

Y
X
ΔY
ΔX
Reading window in the X and Y direction
Code sheet displacement in the X and Y direction

Figure 3.6 X/Y direction for the displacement of **OIC-xxxx-CB1** code sheets

Reading Windows for OIC-xxxx-CB1 Code Sheets



Permissible Code Sheet Displacement for CB3 Code Sheets

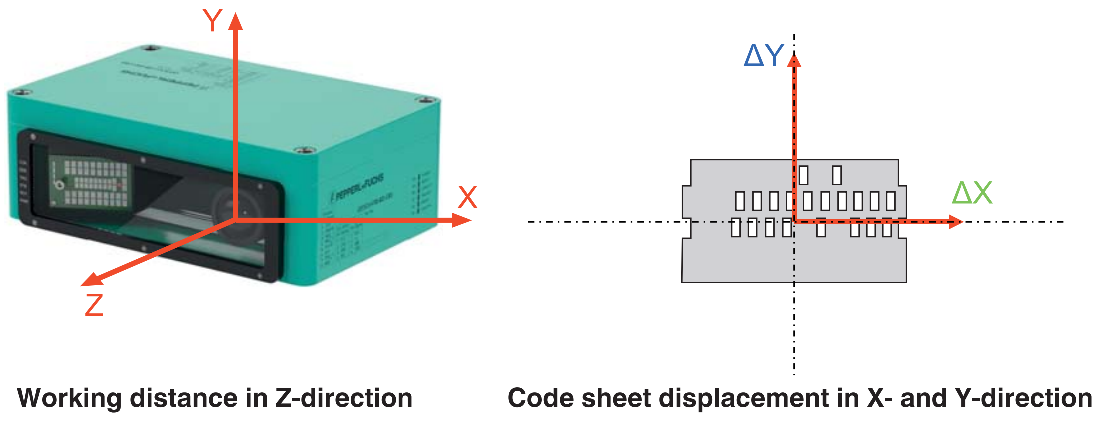

Y
X
Z
ΔY
ΔX
Working distance in Z-direction
Code sheet displacement in X- and Y-direction

Figure 3.7 X/Y direction for the displacement of **OIC-xxxx-CB3** Code sheet.

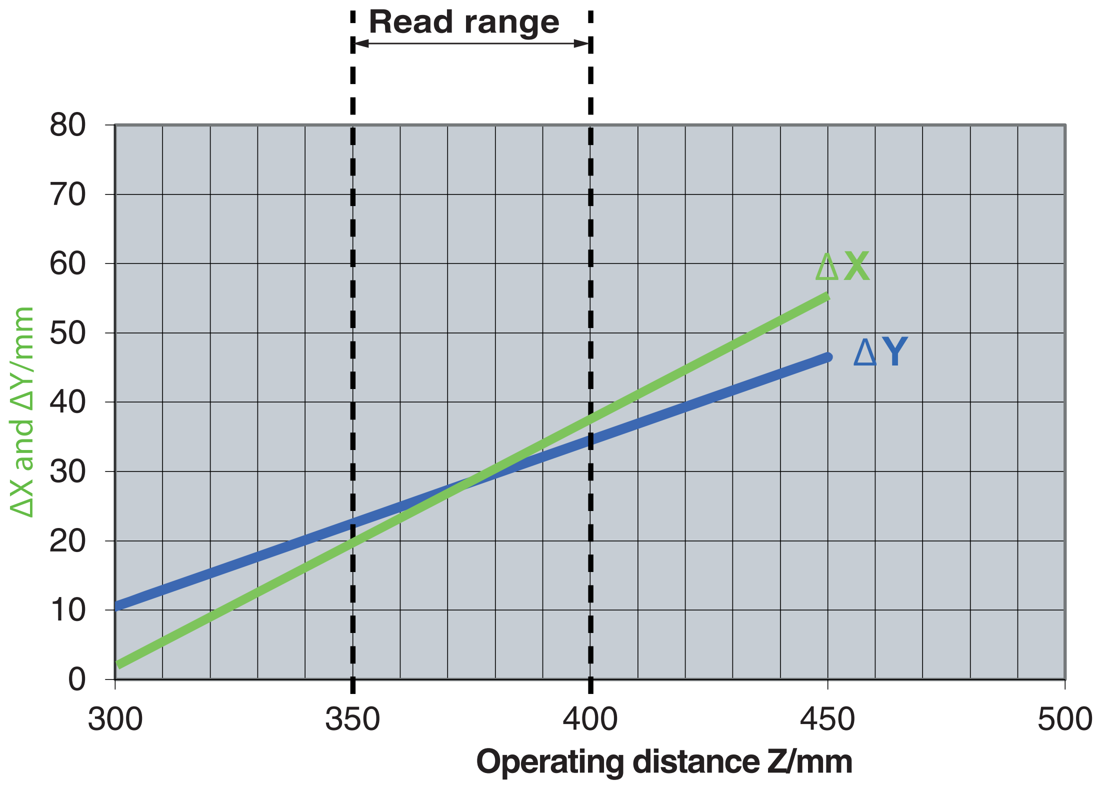


Figure 3.8 Permissible displacement of read-only tag ± ΔX and ± ΔY depending on reading distance Z

A reading window in the X and Y direction determines the permissible displacement of read-only tags. The graphic shows that as the operating distance increases, the possible code sheet displacement increases. The graphic is applicable provided that the code sheet is parallel with the OIT system and the center of the code sheet is shown on the optical axis of the lens.

Permissible Reading Window Displacement for CB3 Code Sheets

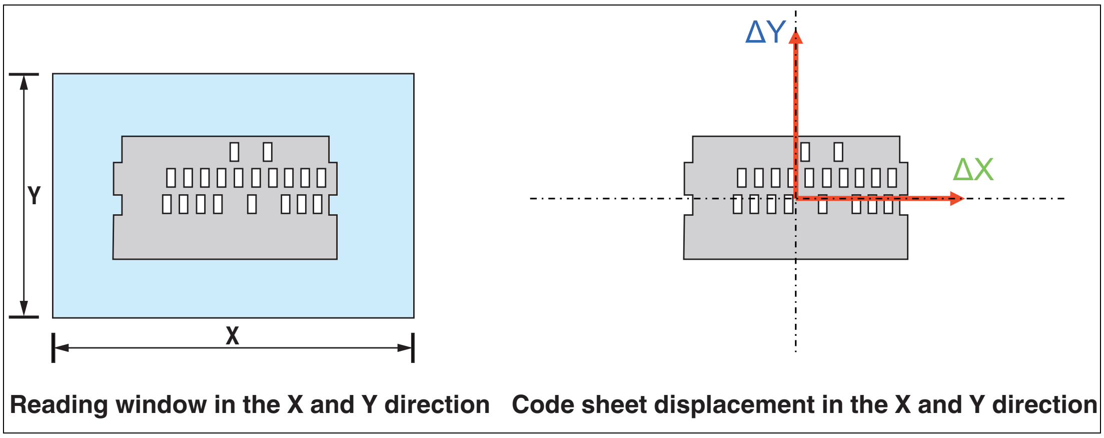
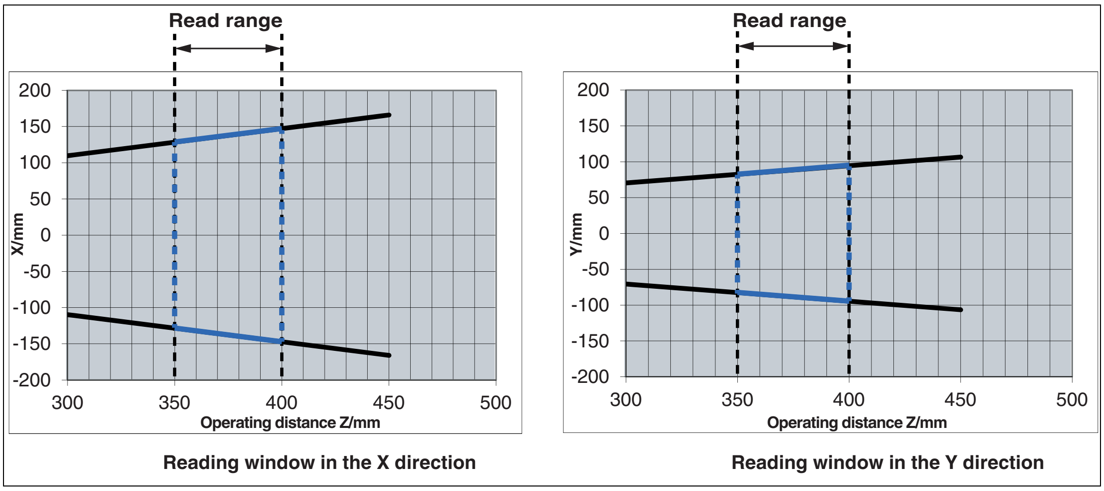

ΔY
ΔX
Y
X
Reading window in the X and Y direction
Code sheet displacement in the X and Y direction

Figure 3.9 X/Y direction for the displacement of **OIC-xxxx-CB3** code sheets

Reading Window for OIC-xxxx-CB3 Code Sheets



3.2.2 Setting the Lighting Angle

For optimal detection of the perforated matrix, code sheets must be illuminated at a certain angle to achieve maximum contrast between the holes and the metal plate. The angle must be set manually depending on whether you use CB1 or CB3 code sheets. The lighting unit is mounted on a rail in the OIT housing. The lighting unit can be moved to the right position using a scale.

The lighting unit is preset for the CB1 code sheet as the factory default.

Caution!

Property Damage Caused by Electrostatic Discharge

If the housing is opened, the electronics may be damaged by electrostatic discharges.

- Avoid electrostatic charges.
- Please note the general requirements for the protection of electronic components against electrostatic phenomena in accordance with IEC 61340-5-1.

Setting the Lighting Angle to CB3 Code Sheets

1. Disconnect the device from the power supply.
2. Unscrew the enclosure cover.

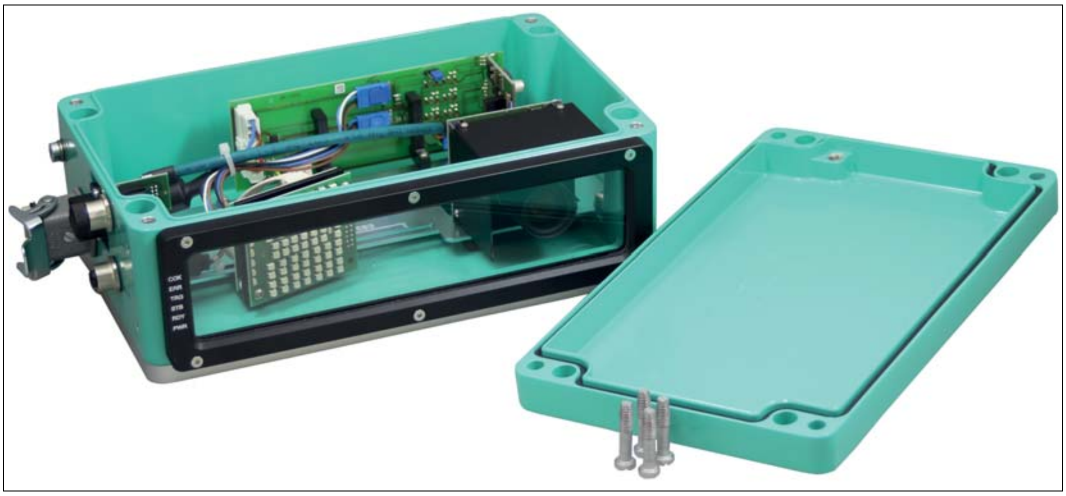


Figure 3.10 Removing the enclosure cover

3. Use a 2.5 mm hexagon socket wrench to loosen the two hexagon socket cap head screws in the lighting unit. The screws do not need to be completely removed.

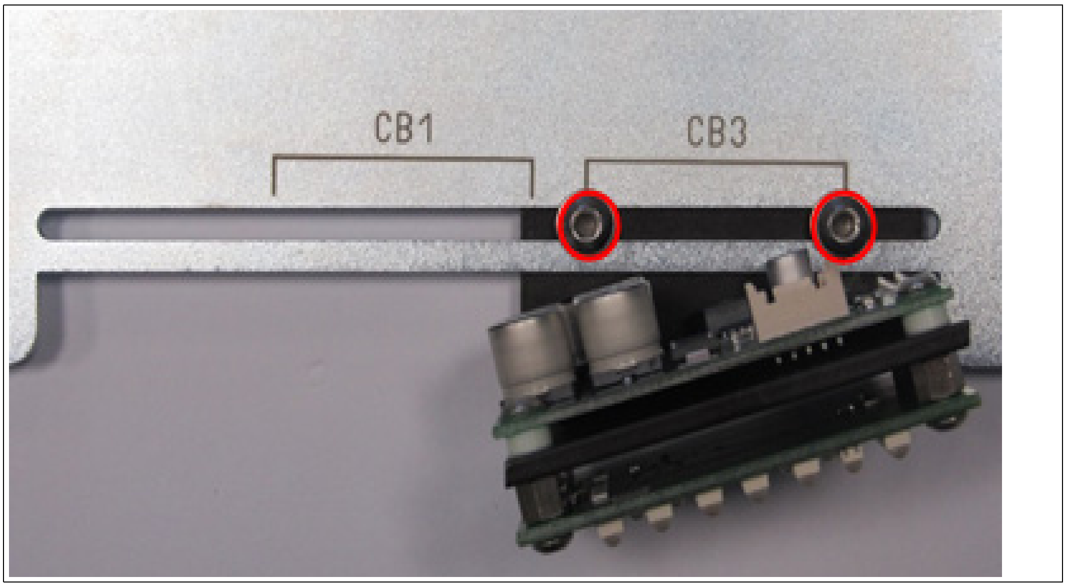


Figure 3.11 Positioning on the rail (CB3)

4. Move the lighting unit along the rail to the **CB3** position.
5. Tighten the two hexagon socket cap head screws to a torque of 1.5 Nm.
6. Screw the enclosure cover back down.

Setting the Lighting Angle to CB1 Code Sheets

The procedure for setting the lighting unit for the CB1 code sheet is almost identical and differs only in the positioning on the rail (CB1).

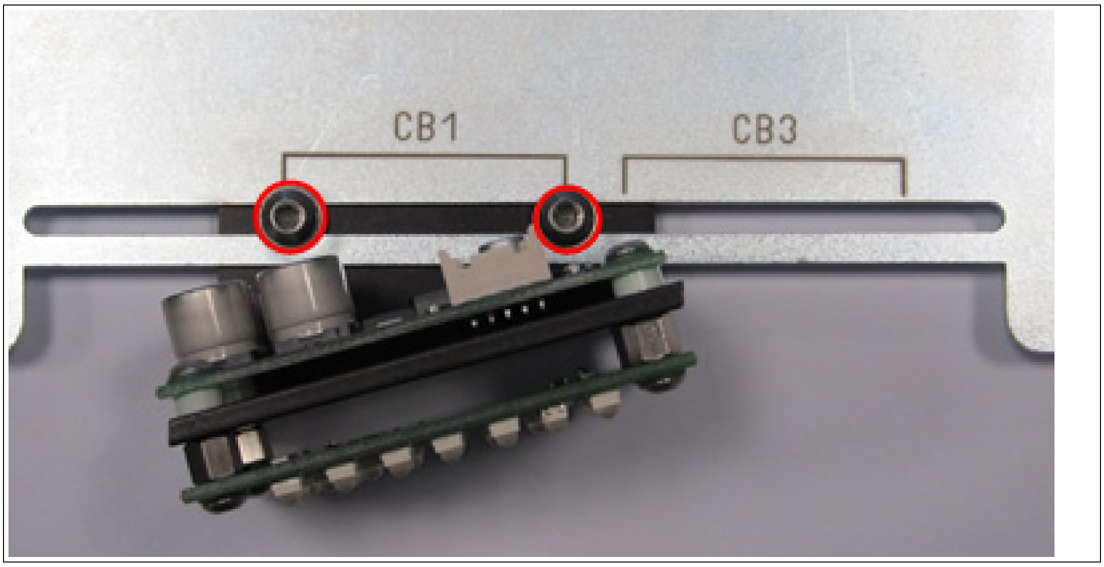


Figure 3.12 Positioning on the rail (CB1)

3.2.3 Mounting the Device

The device has a preassembled mounting base with four symmetrically positioned M6 threads on the base of the housing for easier installation.

The illustration below shows all the relevant housing dimensions in mm:

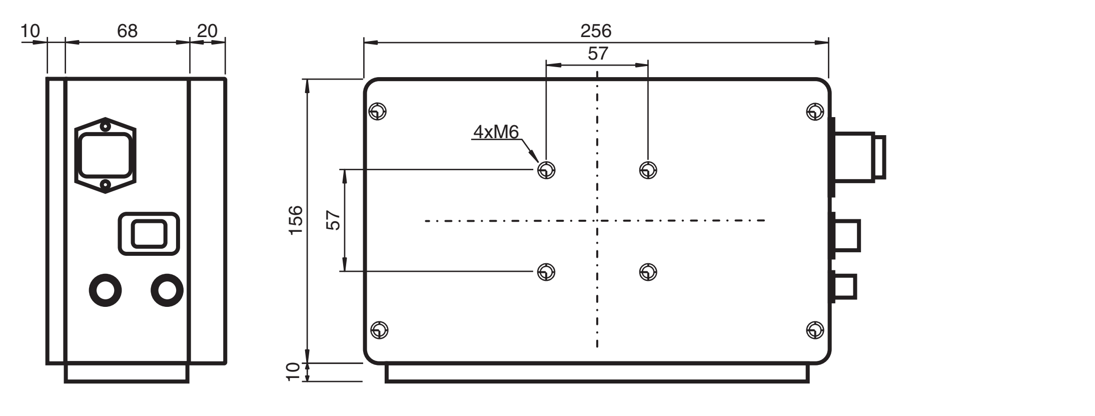


Figure 3.13 Dimensions of the OIT500 system

Mounting and Aligning the Device

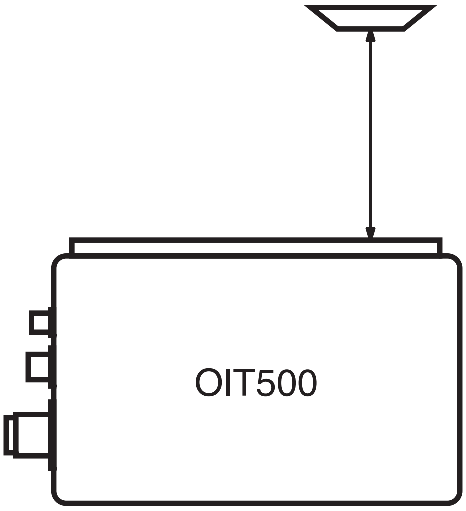
1. Mount the device to enable easy access to the sensor connections for configuration with the associated operating software.
2. Roughly position the device to face the code sheet.

3.2.4 Mounting the Code Sheet

Aligning/Adjusting Code Sheets

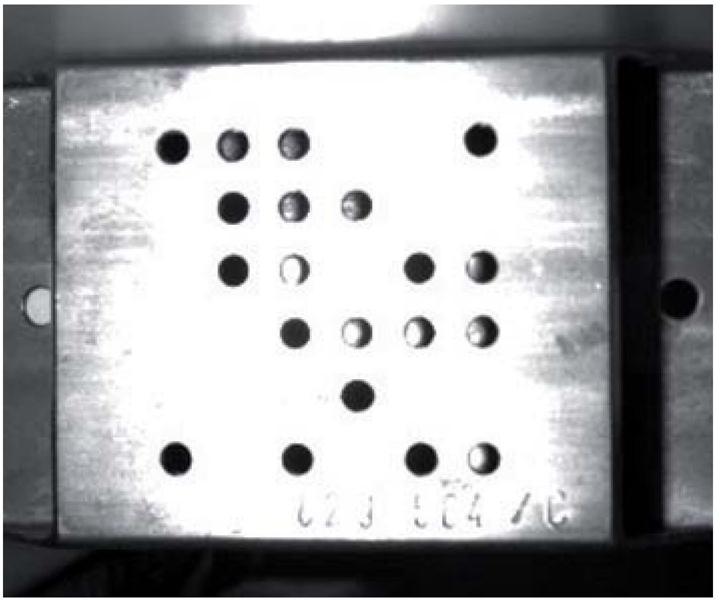
The OIT system is set to a reading distance of 380 mm by default.

Mount the code sheets parallel with the front panel, so that the lens faces the code sheet. The tilt angle must not exceed 10°.



Example!

In this image, the contrast between the holes and the metal is too low. If the tilt angle is too high, the light can be reflected through some holes, meaning the perforated matrix may not be detected correctly.



3.3 Establishing an Electrical Connection

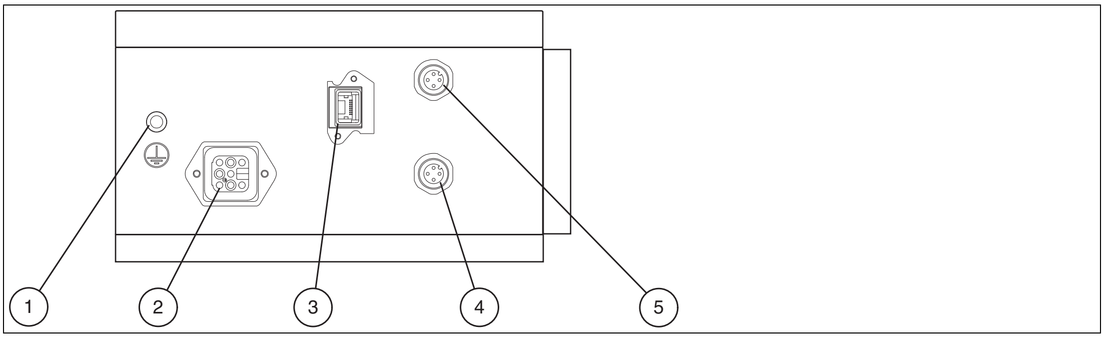


Figure 3.14 Connections

1. Ground connection
2. Power supply (Harting Han® 8D-F plug)
3. Network connection (RJ45 network socket, IP65)
4. Trigger connection (4-pin M12 socket)
5. Do not use connection (reserved)

Providing a Power Supply

To supply voltage to the OIT system, proceed as follows:

1. Insert the Harting Han Q 8/0-F connector into the plug provided on the side of the housing.
2. Pull the safety bracket over the connector until the bracket snaps into place.

   ↳ This ensures that the power cable cannot be pulled out inadvertently.

Connecting a Trigger Sensor

To connect a trigger sensor, proceed as follows.

Plug the 4-pin M12 plug into the socket provided for this purpose on the side of the housing. You can use Pepperl+Fuchs E2/A2 sensors with electrical outputs.

Creating a Network Connection

For the initial commissioning of the device, connect the device directly to the PC/laptop:

1. If you are using a standard RJ45 network connector, insert the RJ45 network connector into the network socket on the side of the housing until the plug snaps into place in the socket. To remove the RJ45 network connector, you will need a small slotted screwdriver to press down the plastic tab while pulling out the RJ45 network connector.
2. If you are using a PushPull RJ45 network connector, insert the PushPull RJ45 network connector into the network socket on the side of the housing until the black housing snaps into place in the socket. The network connection is secured with degree of protection IP65.
3. Use a crossover cable for a direct connection between the device and a PC/laptop.

Note!

Use a straight cable if you are operating the device on a network.

4. Check the network settings for the PC/laptop as described in the next chapter.

Note!

Network Connection with Degree of Protection IP65

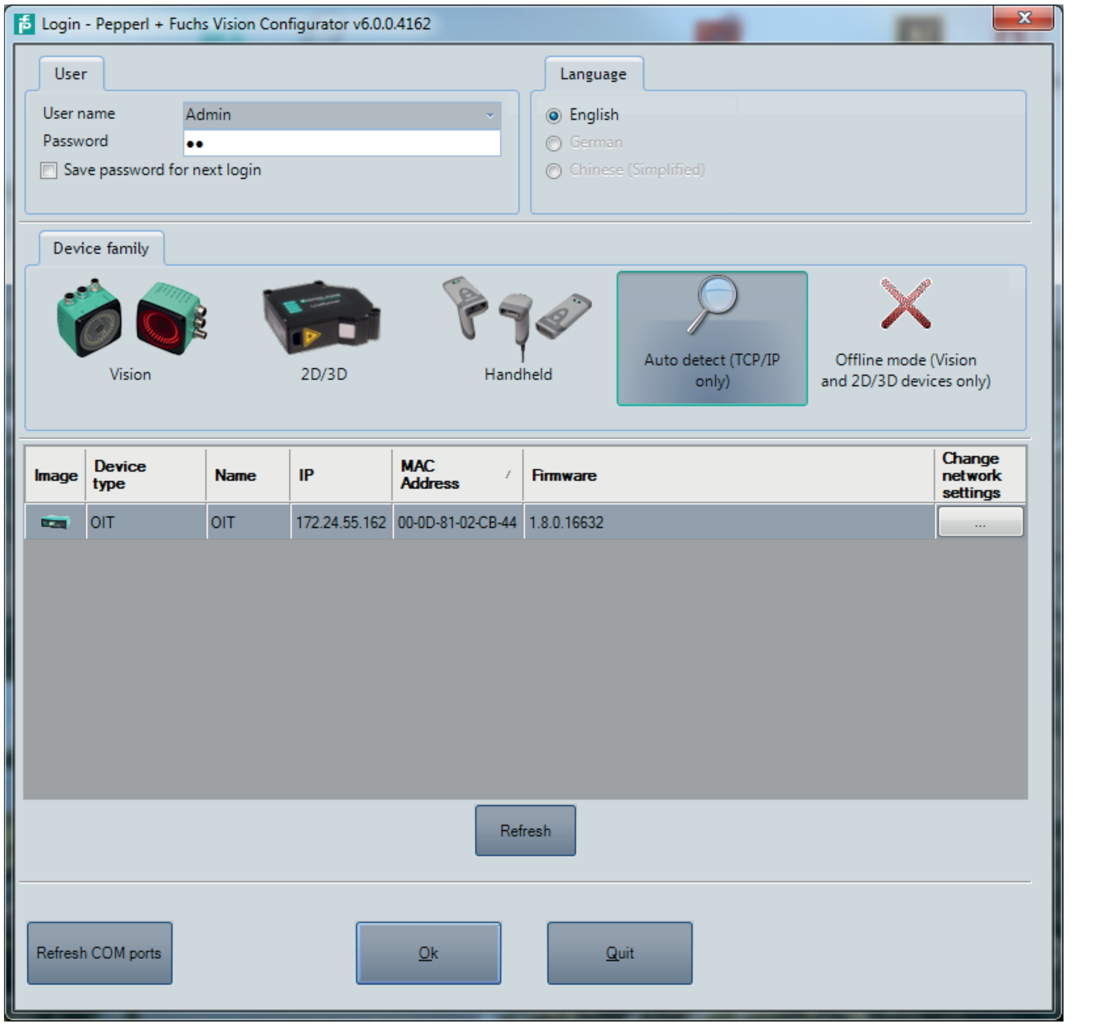
The network connection on the OIT has degree of protection IP65. To ensure the IP65 degree of protection, only use a network connector with degree of protection IP65. Do not use a standard RJ45 network connector.

3.4 Setting up Windows Network Communication between the Device and a PC/Laptop

The OIT system is delivered with a fixed IP address (192.168.2.5). To enable communication within the network, the network settings of your PC/laptop must be synchronized with the device and may need to be adjusted. To do so, proceed as follows.

Auto Detect Function

The **Auto detect** function is available in Vision Configurator. You can use this function to display all connected Ethernet TCP/IP devices. You can select the device you are looking for in the output window and read information such as the device's IP address. This function is also useful if you have inadvertently changed the IP address and no longer know it.


Login - Pepperl + Fuchs Vision Configurator v6.0.0.4162
User
Language
Device family
Vision
2D/3D
Handheld
Auto detect (TCP/IP only)
Offline mode (Vision and 2D/3D devices only)
Refresh
Refresh COM ports
Ok
Quit

User name: Admin
Password: ••
□ Save password for next login

● English
○ German
○ Chinese (Simplified)

Image	Device type	Name	IP	MAC Address	Firmware	Change network settings
	OIT	OIT	172.24.55.162	00-0D-81-02-CB-44	1.8.0.16632	...

Figure 3.15 Auto detect function in Vision Configurator

Once **Auto detect (TCP/IP only)** is selected, the software first displays an output window that is initially empty. When a device is detected, it appears in the output window. The following information is displayed for each device:

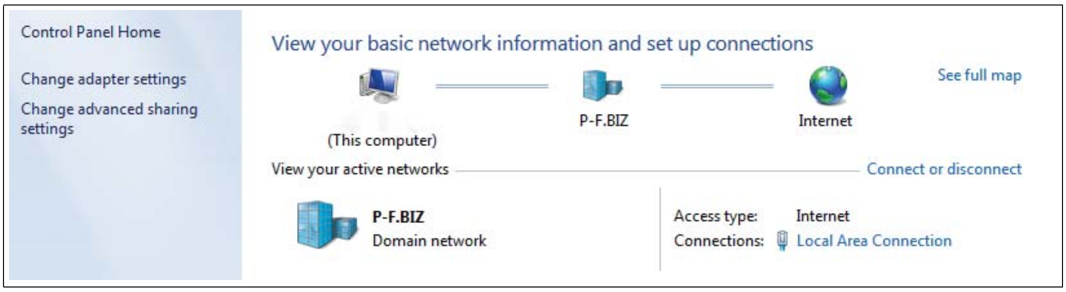
- An “Image" of the device
- The "Device type"
- The "Name" of the device
- The "IP" of the device

- The "MAC address" of the device
- And the "Firmware"

Setting the IP Address of the PC

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

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



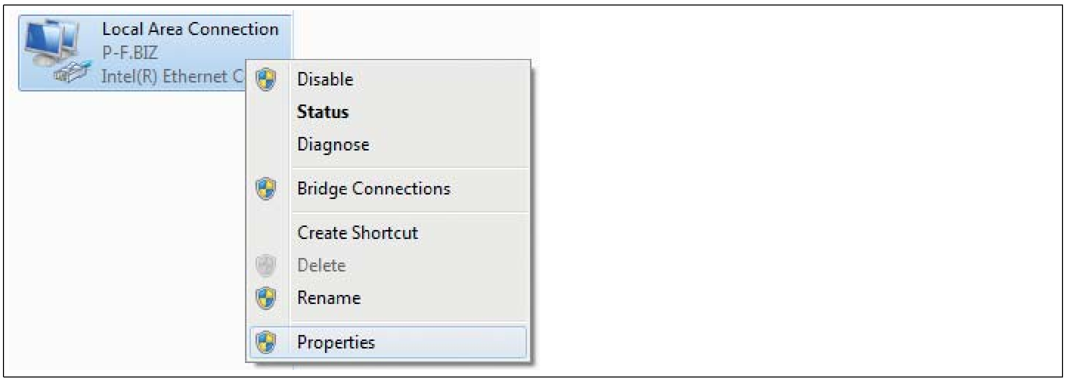
View your basic network information and set up connections

View your active networks

P-F.BIZ
Domain network

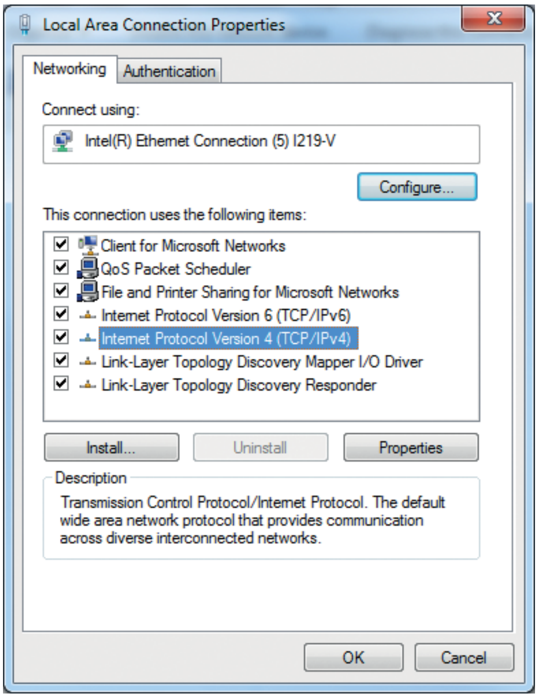
Access type: Internet
Connections: Local Area Connection

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



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

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



Local Area Connection Properties

Networking | Authentication

Connect using:

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

Configure...

This connection uses the following items:

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

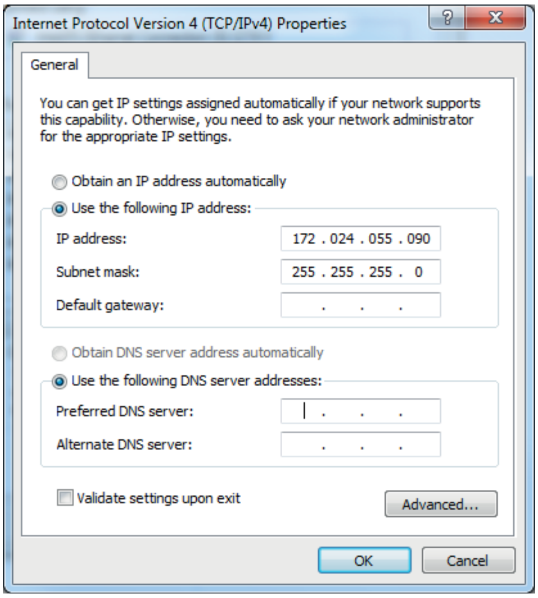
Install... | Uninstall | Properties

Description

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

OK | Cancel

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



Internet Protocol Version 4 (TCP/IPv4) Properties

General

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

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 172 . 024 . 055 . 090

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

☐ Validate settings upon exit

Advanced...

OK | Cancel

6. Select the "**General**" tab.
7. Select the input function "**Use the following IP address**."
8. Use the device's IP address that you found using the "Auto-detect function." In this example, enter the following IP address and subnet mask:

- **IP address: 172.24.55.90**
- **Subnet mask: 255.255.255.0**

Note!

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.

9. Click **OK**, and click **Cancel** in the next dialog.

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

Note!

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

Note!

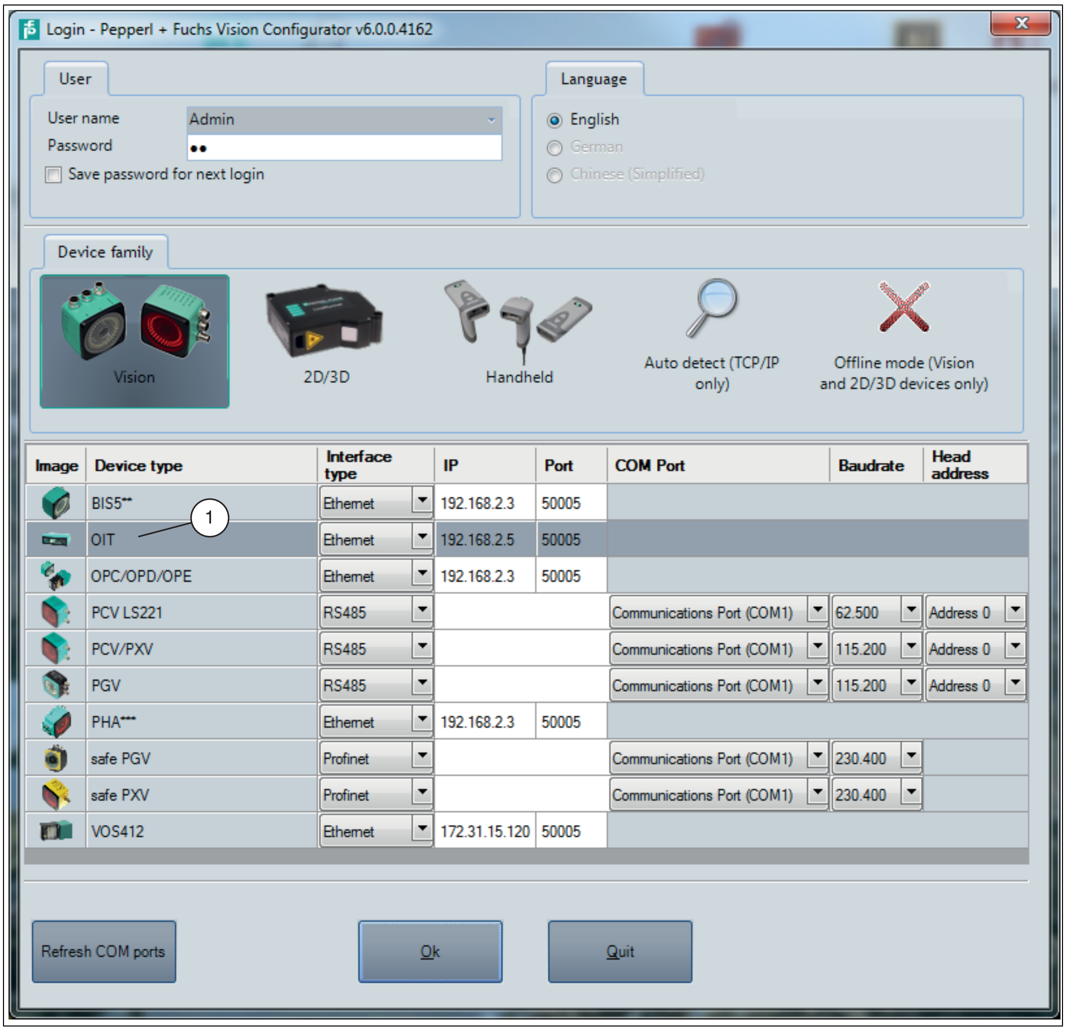
Documenting the network configuration

The OIT communicates via the TCP/IP protocol. If you have made changes to the network configuration, e.g. if you have changed the IP address, these changes must be recorded to ensure seamless communication.

3.5 Connecting the OIT System with Vision Configurator

Establishing a Connection

1. Start the Vision Configurator software.
2. Enter your user name and password.
3. Double-click on the device type **OIT** (1).



Login - Pepperl + Fuchs Vision Configurator v6.0.0.4162

User

User name: Admin

Password: ••

□ Save password for next login

Language

● English

○ German

○ Chinese (Simplified)

Device family
Vision
2D/3D
Handheld
Auto detect (TCP/IP only)
Offline mode (Vision and 2D/3D devices only)
1

Image	Device type	Interface type	IP	Port	COM Port	Baudrate	Head address
	BIS5**	Ethernet	192.168.2.3	50005			
	OIT	Ethernet	192.168.2.5	50005			
	OPC/OPD/OPE	Ethernet	192.168.2.3	50005			
	PCV LS221	RS485			Communications Port (COM1)	62.500	Address 0
	PCV/PXV	RS485			Communications Port (COM1)	115.200	Address 0
	PGV	RS485			Communications Port (COM1)	115.200	Address 0
	PHA***	Ethernet	192.168.2.3	50005			
	safe PGV	Profinet			Communications Port (COM1)	230.400	
	safe PXV	Profinet			Communications Port (COM1)	230.400	
	VOS412	Ethernet	172.31.15.120	50005			

Figure 3.16 Selecting the device

↳ The connection window opens.

4. Click the **Connect** button (1).

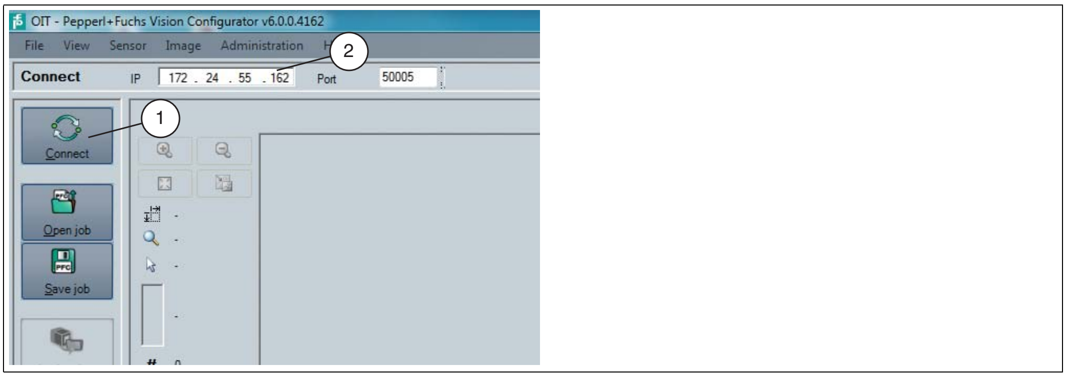

OIT - Pepperl+Fuchs Vision Configurator v6.0.0.4162
File View Sensor Image Administration H
Connect
IP 172 . 24 . 55 . 162
Port 50005
Connect
Open job
Save job
1
2

Figure 3.17 Connecting

↳ The connection to the device is established and the application screen opens.

Note!

In some cases, the IP address of the device may not be applied automatically. In this case, you must enter the device's IP address in the input field (2) manually and click the **Connect** button (1) again.

4 Parameterization

The factory default parameters are stored in the OIT memory. You can adjust the parameters using the Vision Configurator software or the control panel.

Configuration Parameters

Configuration ID	Parameter ID	Value	Description
System	EnableResultPort	0,1	Activates TCP/IP result output
	UDP	0,1	Activates UDP result output
	PortOut	1 – 65,536	TCP port
	PortOutUdp	1 – 65,536	UDP port
	SensorName	String	Name of the sensor
	ImageTransferActive	0,1 (default: 0)	Activates image transfer if the sensor is connected. 0 for RunTime, 1 to set up (with image display)
	ImageTransfer	ALL_IMG ERROR_IMG GOOD_IMG	Specify which images should be transferred to the end device
	BadImageTransfer	INSPECT_IMAGE INSPECT_IMAGE_2	Specify which image should be sent to the end device if the sensor has transferred two images in the case of a fault
Camera	ExposureTime	10 – 20,000 (default: 1500)	Exposure time in µs
	Gain	0 – 255 (default: 0)	Image gain
	StartLive	1	Starts continuous image transfer mode
	StopLive	1	Stops continuous image transfer mode
	CheckImages	0,1	If this is set, the sensor also starts code sheet detection in continuous image transfer mode
	DisplayPause	100 – 10,000	Delay time before the next image capture can be started, in ms
Command	TriggerStart	1	Trigger start
	GetLastImg	1	Loads the last captured image
	GetErrImg	1	Loads the next error image in a series of up to ten images
	SetDefaultParams	1	Loads the default parameters
	ResetCounters	1	Sets all internal counters to 0
Control panel	EdgeMode	TRIGGER_EDGE_POS TRIGGER_EDGE_NEG	Sets the trigger activation according to the rising or falling edge
	TriggerReleaseOption	0,1	If this is set, the sensor waits for an input signal when activating the trigger. If this option is disabled, the input signal of the trigger is ignored

Configuration ID	Parameter ID	Value	Description
Decoder	XStart	0 – 751	X coordinate of the decoder frame in which the OIT attempts to read a perforated matrix
	YStart	0 – 479	Y coordinate of the decoder frame in which the OIT attempts to read a perforated matrix
	RoiWidth	1 – 752	Width of the decoder frame (ROI)
	RoiHeight	1 – 480	Height of the decoder frame (ROI)
	DecoderPlate	INSPECTION_CB1_PLATE INSPECTION_CB3_PLATE	Activates the reading of CB1 code sheets Activates the reading of CB3 code sheets
	CaptureTwoImages	0,1	Activates the capture and evaluation of two images for one read operation. In the case of a faulty reading in the first image, the second image will be evaluated. Different exposure times increase the number of successful readings
	ReducedFlashtimePercent	1 – 100	The percentage of the exposure time of the second image compared to the first image
	SuppressDuplicates	0,1	Activate/deactivate suppression of the decoding of consecutive, identical codes
	GoodString	XML string	Specify which data string should be output after a successful reading
	BadString	XML string	Specify which data string should be output after a failed reading
	TimoutDecode	0...	Maximum decoding time before the reading is considered to have failed
	CodetypeCB3	8 – 12	Setting for data bits in the CB3 code sheet
	Orientation	ORIENTATION_0DEG ORIENTATION_180DEG ORIENTATION_0DEG_MIRRORED ORIENTATION_180DEG_MIRRORED	Determines the alignment of the code sheet in the sensor image
	CodetypeCB1	CODETYPE_6x6 CODETYPE_6x6_MIRRORED CODETYPE_5x5	Determines the sheet type for CB1
View	ShowCluster	0,1	Activates optical marking of the detected holes and displays the center point of the holes
	HoldCluster	0,1	When activated, the hole center points that are currently marked are retained if a new image is displayed

5 Commissioning

The OIT system is commissioned and configured using the **Vision Configurator** software.

Introduction to the Vision Configurator Operating Software

The Vision Configurator software has a user-friendly interface that makes it easy to operate the sensor. Standard functions include connecting to the sensor, configuring the operating parameters, saving data sets, displaying data, and diagnosing faults.

The following user roles with different authorizations are preset in Vision Configurator.

User Rights and Password

User rights	Description	Password
Default	View all information Create users at same level or below	No password required
User	View all information Sensor configuration Create users at same level or below	User
Admin **PFAdmin**	View all information Sensor configuration Create and delete users	Request the admin password from Pepperl+Fuchs

Note!

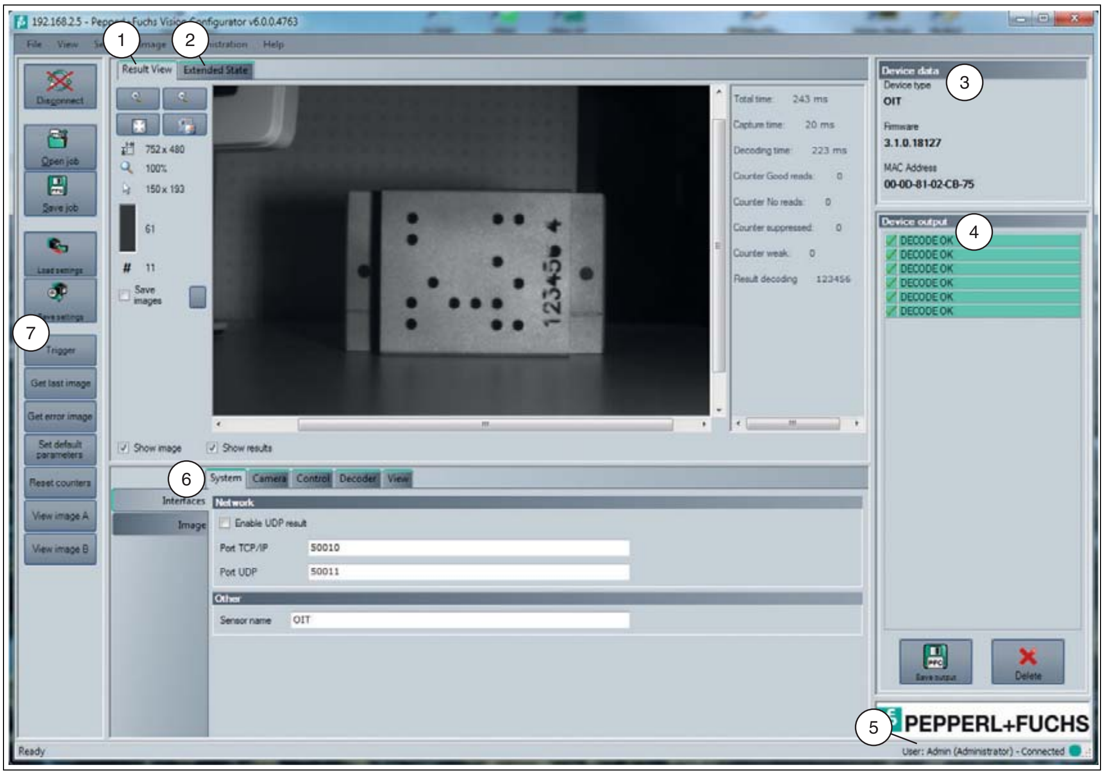
The latest version of the Vision Configurator software can be found online at https://www.pepperl-fuchs.com. The Vision Configurator manual outlines the properties of the operating software for multiple devices. You can also access this manual from our website.

5.1 Screen Layout

The application screen opens after you log in.

Note!

The individual functions are dependent on the selected user role.



☑ Show image ☑ Show results

Total time: 243 ms
Capture time: 20 ms
Decoding time: 223 ms
Counter Good reads: 0
Counter No reads: 0
Counter suppressed: 0
Counter weak: 0
Result decoding 123456

Device data
Device type
OIT
Firmware
3.1.0.18127
MAC Address
00-0D-81-02-CB-75

Device output
✓ DECODE OK
✓ DECODE OK
✓ DECODE OK
✓ DECODE OK
✓ DECODE OK
✓ DECODE OK

Network
□ Enable UDP result
Port TCP/IP 50010
Port UDP 50011

Other
Sensor name OIT

Ready
User: Admin (Administrator) - Connected

Figure 5.1 Application screen

1. The **Result View** displays the read images and offers basic editing tools. If the **Show results** option is enabled, additional information, such as the required decoding time, is displayed.
2. The **Extended State** area displays warning and status messages relating to the captured image.
3. The **Device data** area displays information about the connected sensor.
4. The **Device output** area displays various items of status information, such as whether a code could be read successfully.
5. The status bar shows information about the user who is logged in and the sensor connection status.
6. The parameterization area is split into several subareas and contains sensor-specific parameters.
7. The toolbar allows direct access to selected menu items.

5.2 Menu Bar

The menu bar contains a list of menu items. The functionality depends on the type of sensor that is connected and the permissions of the user logged in.

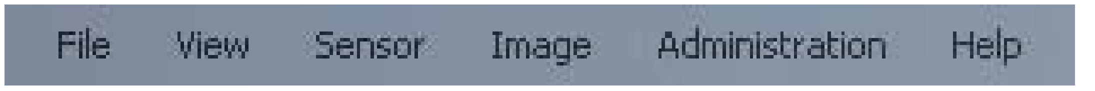


Figure 5.2 Menu Bar

5.2.1 *File* Menu

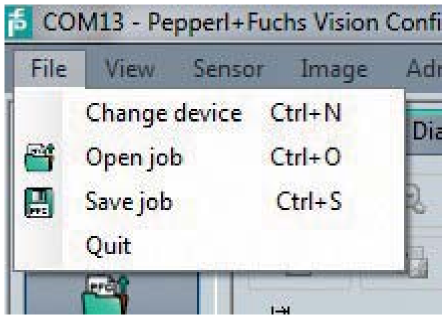


Figure 5.3 **File** Menu

Change device	Disconnects the device and returns to the Login dialog.
Open job	Loads a sensor configuration stored on the PC.
Save job	Saves the current sensor configuration on the PC.
Quit	Terminates the program.

Table 5.1 **File** Menu

5.2.2 *View* Menu

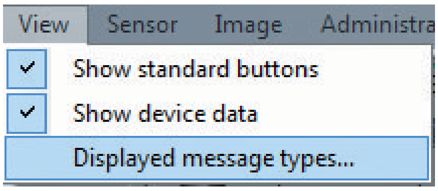


Figure 5.4 **View** Menu

Show standard buttons	Toggles the display of the buttons in the bar on the left on and off.
Show device data	Hides the display of the sensor data in the top right of the screen.
Displayed message types...	Opens a selection window in which the following display windows can be activated or deactivated: Info, Result OK, Result not OK, Warning, Error, Critical, Assert.

Table 5.2 **View** menu

5.2.3 *Sensor* Menu

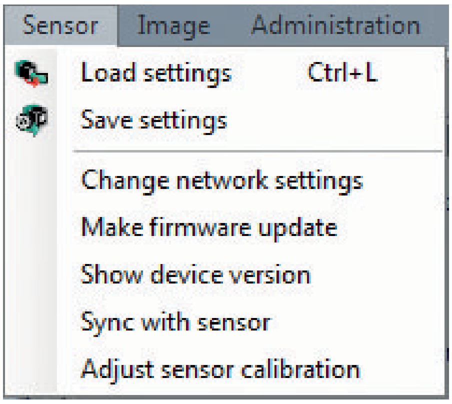


Figure 5.5 **Sensor** menu

Load settings	Loads the saved settings from the sensor
Save settings	Saves the settings to the sensor
Change network settings	Change the network settings. The settings window allows you to set the IP address, subnet mask, gateway address, and DHCP
Make firmware update	Performs firmware updates. This command should be used by experienced users only
Show device version	Displays the device version
Sync with sensor	Synchronization with the sensor
Adjust sensor calibration	Adjust the sensor calibration

Table 5.3 **Sensor** menu

Note!

Firmware Update

Once you have upgraded the firmware and **Update complete** is displayed, restart the sensor.

5.2.4 *Image* Menu

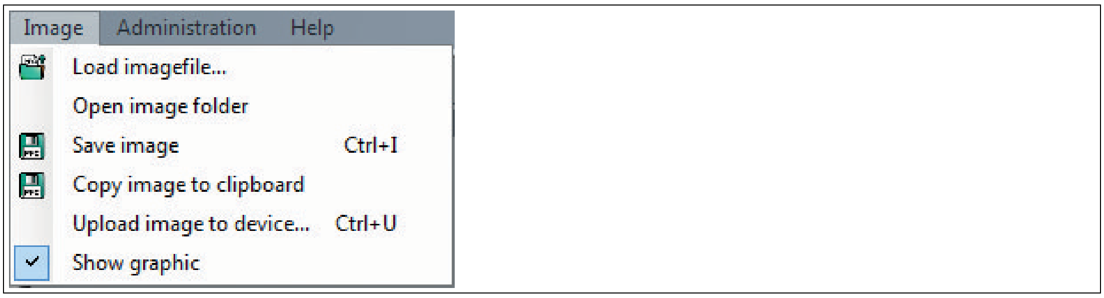


Image Administration Help

Load imagefile...

Open image folder

Save image Ctrl+I

Copy image to clipboard

Upload image to device... Ctrl+U

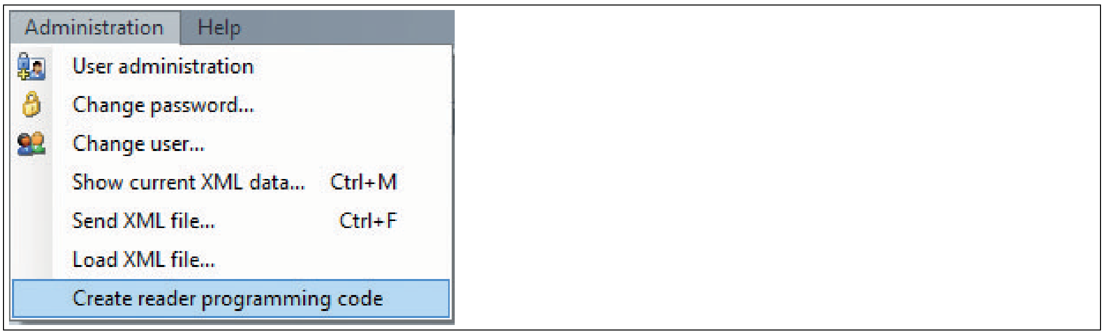
✓ Show graphic

Figure 5.6 **Image** menu

Load imagefile	Loads the image file
Open image folder	Opens the folder in which images are currently saved
Save image	Saves the image currently displayed on the PC
Copy image to clipboard	Loads an image file to the clipboard
Upload image to device	Uploads an image to the device
Show graphic	Switches display data sent from the sensor on and off in the image.

Table 5.4 **Image** menu

5.2.5 *Administration* Menu


Administration Help

User administration

Change password...

Change user...

Show current XML data... Ctrl+M

Send XML file... Ctrl+F

Load XML file...

Create reader programming code

Figure 5.7 **Administration** menu

User administration	Opens a window that shows all currently created users at the same authorization level or lower. New users at the same authorization level or lower can also be created and deleted here. In addition, a user password can be reset to the default password for the relevant user level.
Change password	Changes the current user's password.
Change user	The login screen opens and a different user and/or sensor can be selected.
Send XML file...	Saves the XML data on a computer.
Load XML file...	Loads XML data from a computer.
Create reader programming code	Creates a reader programming code

Table 5.5 **Administration** menu

5.2.6 *Help* Menu

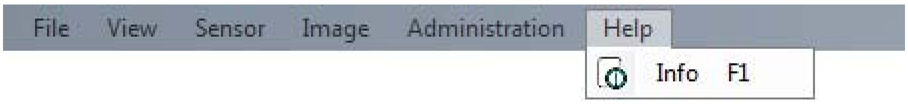


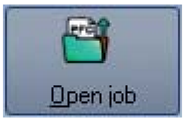
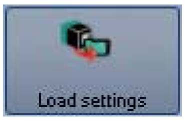
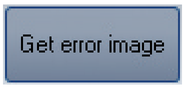
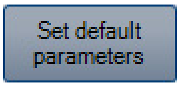
Figure 5.8 **Help** menu

Info	Displays information about Vision Configurator.

Table 5.6 **Help** menu

5.3 Toolbar

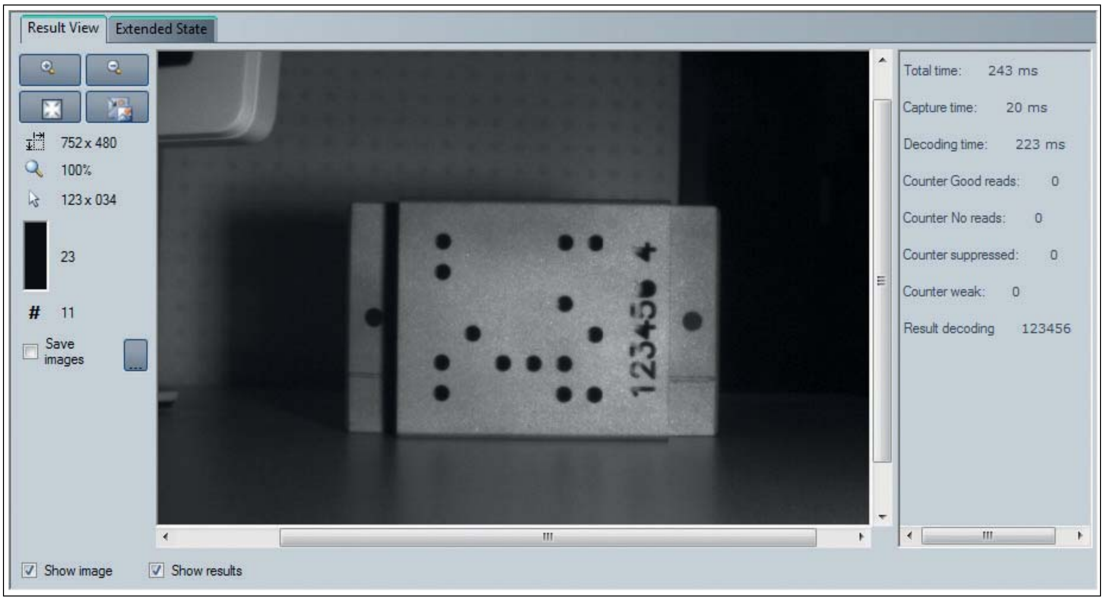
The toolbar contains various functions.

	Disconnects the connection between Vision Configurator and OIT.
	Opens locally-saved settings.
	Saves settings locally.
	Reads the current settings from the OIT.
	Saves the current settings to the OIT.
	Triggers an image capture.
	Sends the last image to Vision Configurator.
	Sends the last image from which no code could be read to Vision Configurator.
	Resets all settings to the factory default settings.
	Resets all counters to 0, such as the counters for successful and failed readings.

 	Sends the last image for image A to Vision Configurator.
 	Sends the last image for image B to Vision Configurator.

5.4 Result View

The **Result View** provides different options for displaying captured images. The **Show image** and **Show results** check boxes can be used to activate and deactivate the image display and result display.



```
Total time:          243 ms
Capture time:        20 ms
Decoding time:       223 ms
Counter Good reads:  0
Counter No reads:    0
Counter suppressed:  0
Counter weak:        0
Result decoding      123456
```

Figure 5.9 Image display (left) and result display (right)

Clicking the right mouse button or the context menu button calls up the following context menu:

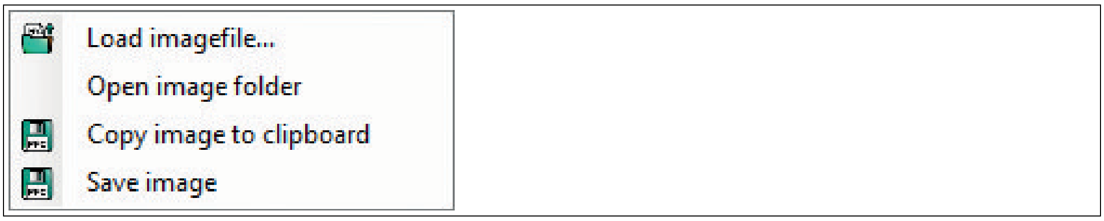


Figure 5.10 Context menu in the image display

Context Menu

Load image file...	Loads an image capture
Open image folder	Opens the storage location
Copy image to clipboard	Loads an image file to the clipboard
Save image	Saves the current image capture. You can choose whether the result display should be saved with the image capture

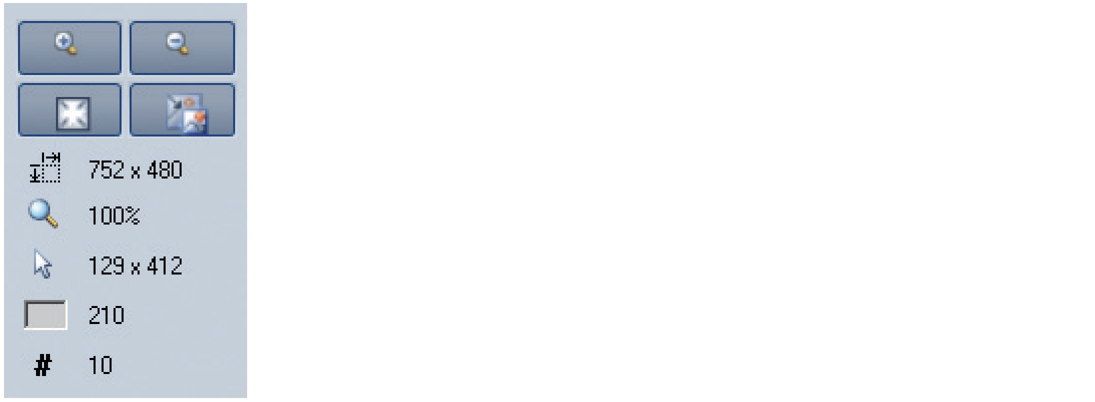


Figure 5.11 Tool bar in the image display

Magnifier +	Zooms into the image
Magnifier -	Zooms out of the image
Fit to window	Fits the image to the window
Original size	Displays the image in its original size
Size details	Shows the length and width of the image
Zoom factor	Shows the zoom factor
Position details	Shows the cursor position
Gray scale value details	Gray scale value of the pixel that the cursor is hovering over
Image number	Displays the image number

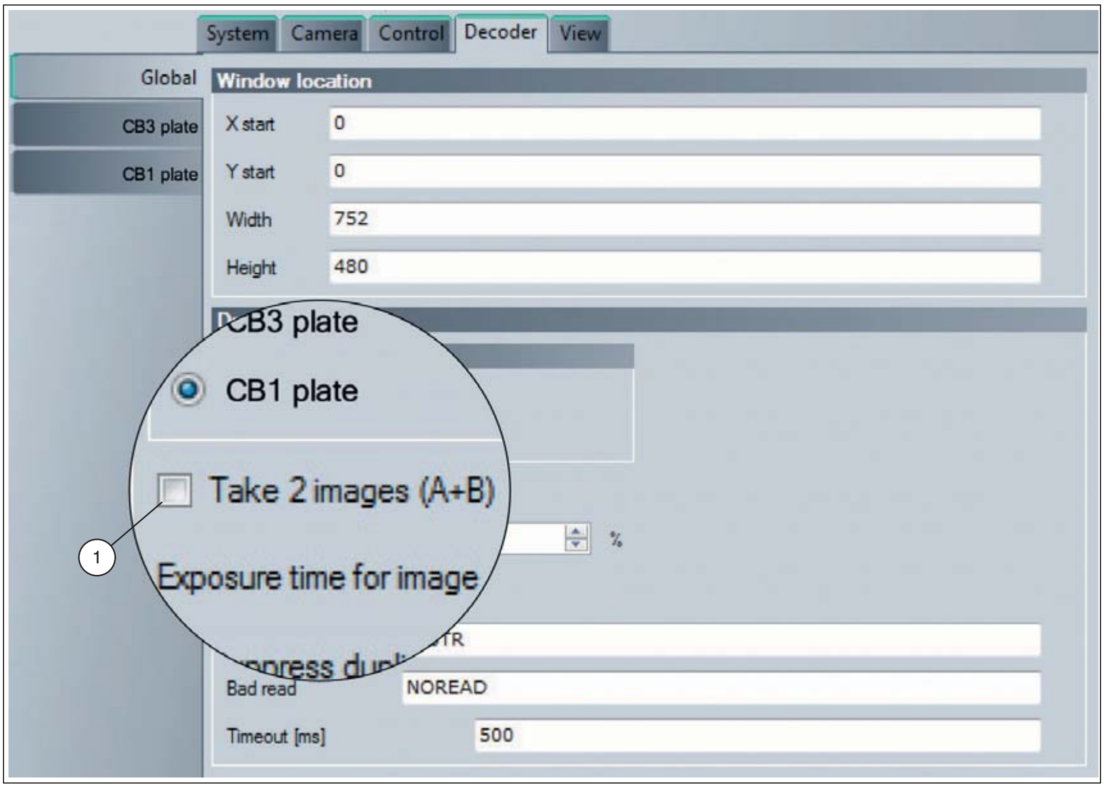
Tip

Alternatively, you can use the mouse wheel to zoom in and out of the image, and use the mouse to adjust the field of view.

5.5 Extended State

The **Extended State** tab contains two sections. Warning messages are displayed in the upper section of the window and the status is displayed in the lower section.

To enable subsequent evaluation of the warning and status messages, image captures must be set to single image mode. Therefore, always make sure that there is no check mark against the **Take 2 images (A+B)** option (1).



X start 0

Y start 0

Width 752

Height 480

CB3 plate

● CB1 plate

□ Take 2 images (A+B)

Exposure time for image

Bad read NOREAD

Timeout [ms] 500

Figure 5.12 Single image mode

The warning message is signaled via the respective status icon. As soon as an error occurs, the status indicator changes from gray to red (1). The warning message is linked to a weak reading, i.e., a warning message with a red status indicator is displayed at the same time as a weak message (weak counter increases by one).

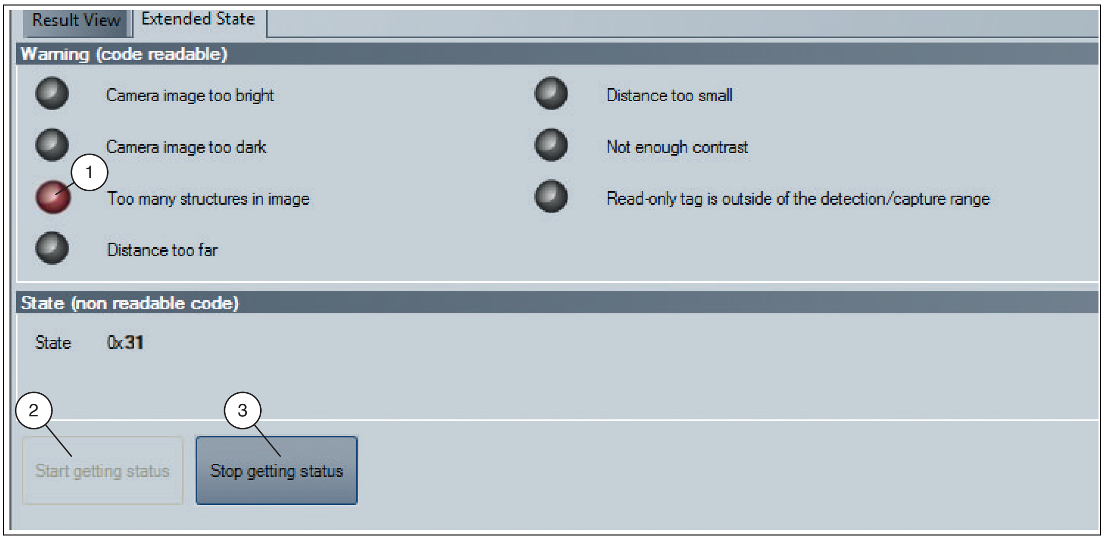


Figure 5.13 Extended State

You can also obtain a status message in the status window. The status message must be activated in advance. Click the **Start getting status** button (2) to activate the status message. If required, you can deactivate the status message by clicking the **Stop getting status** button (3).

Warning Message

Warning message	Description	Remedy
Camera image too bright	The camera image is too bright.	Set the right exposure time, see chapter 5.6.2.
Camera image too dark	The camera image is too dark.	Set the right exposure time, see chapter 5.6.2.
Too many structures in image	Too many shapes in the capture window that are similar to the hole pattern on the code sheet (e.g., screws).	Make sure there are no objects in the capture area that are a similar shape to the hole pattern on the code sheet.
Distance too far	The distance between the code sheet and the OIT system is too great.	Set the right operating distance, see chapter 3.2.1.
Distance too small	The distance between the code sheet and the OIT system is too small.	Set the right operating distance, see chapter 3.2.1.
Not enough contrast	There is insufficient contrast.	Set the right exposure time, see chapter 5.6.2.
Read-only tag is outside of the detection/capture range	The code sheet is at the edge of the reading window.	Move your code sheet so that it is within the reading window again, see chapter 3.2.1.

Status Message

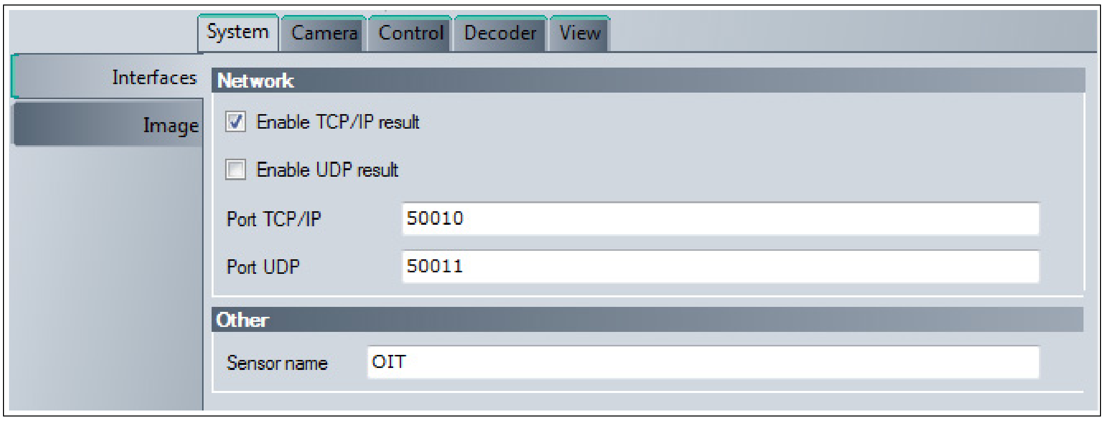
Status message	Description
0x01	No hole structures found
0x02	Not enough hole structures, first loop
0x03	Not enough hole structures, second loop
0x04	No corner marks found, check 1
0x05	Decoding not possible, check 1
0x06	No corner marks found, check 2
0x07	Poor contrast
0x08	No corner marks found, check 3
0x0A	No corner marks found, check 4
0x0B	Decoding not possible, check 2
0x0F	Checksum error
0x2A	Timeout
0x31	Poor read-only tag (too many possible hole structures)
0x32	Timeout while determining the corner points
0x33	Timeout while decoding
0x99	Maximum repeat read reached Double code reading
0x9A	Disconnection during the measuring process
0xFE	No usable data from the PLC

5.6 Configuration window

Various parameters are specified in the configuration window. The individual parameters depend on the current authorization level and are, therefore, not always all visible. Some features are available in different variants only. Depending on the parameters set, some fields will be grayed out.

5.6.1 **System** Tab

Interfaces Menu Item



☑ Enable TCP/IP result

□ Enable UDP result

Port TCP/IP 50010

Port UDP 50011

Sensor name OIT

Figure 5.14 **System** tab, **Interfaces** menu item

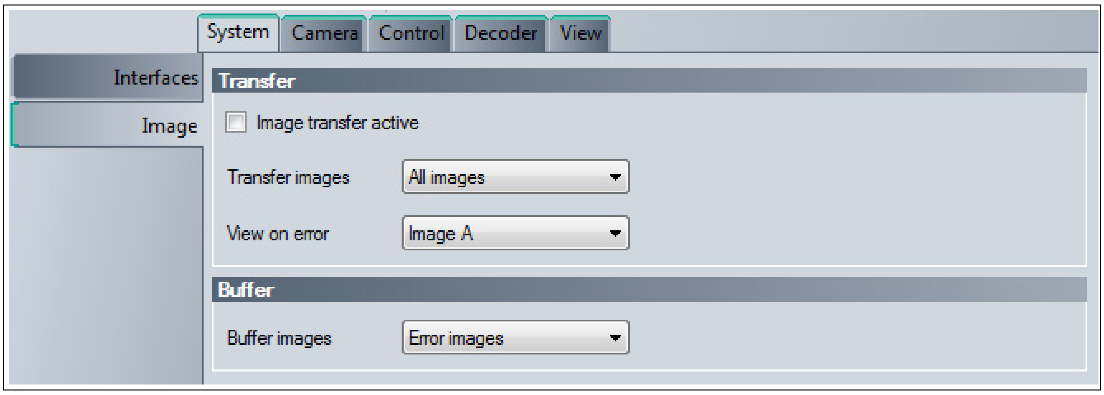
Network

Enable TCP/IP result	Activates the result output via TCP/IP
Enable UDP result	Activates the result output via UDP
TCP/IP port	Entry of the port for the TCP/IP connection
UDP port	Entry of the port for the UDP connection

Other

Sensor name	Enter a device name for the sensor. The device name can be used instead of the IP address to connect to the sensor

Image Menu Item



□ Image transfer active

Transfer images All images

View on error Image A

Buffer images Error images

Figure 5.15 **System** tab, **Image** menu item

Transfer

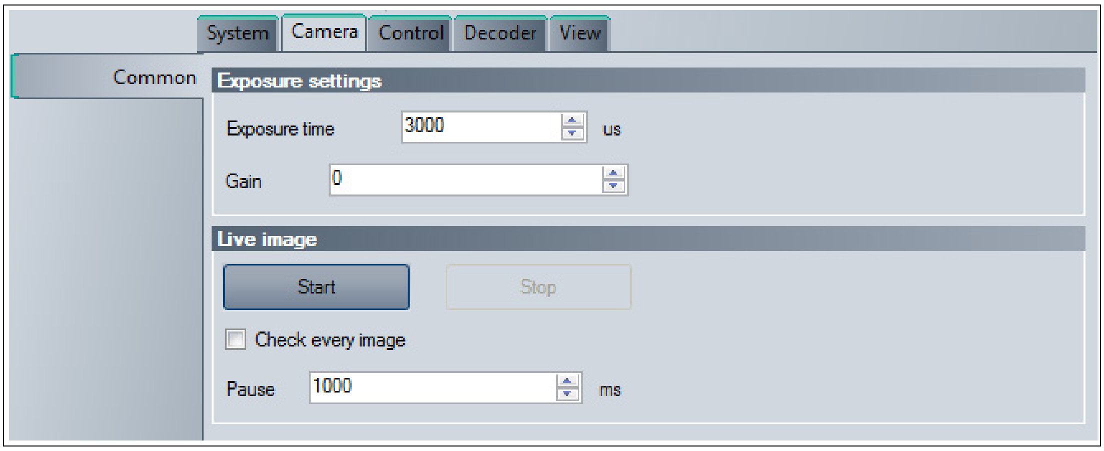
Image transfer active	Activates image transfer to Vision Configurator
Transfer images	Determines which images are to be transferred to Vision Configurator ■ **All images**: All images ■ **Error images**: Images from failed readings ■ **Good images**: Images from successful readings
View on error	Determines which image should be transferred to Vision Configurator after a failed reading. If the capture of two images for one read operation has been activated (see chapter 5.6.4), only one of the two images can be displayed in Vision Configurator. ■ **Image A**: image A is transferred ■ **Image B**: image B is transferred

Buffer

Buffer images	Specifies which images are to be saved. The memory offers sufficient space for approx. five ... ten images. ■ **All images**: All images ■ **Error images**: Images from failed readings ■ **Good images**: Images from successful readings

5.6.2 **Camera** Tab

Common Menu Item



Exposure time 3000 us

Gain 0

□ Check every image

Pause 1000 ms

Figure 5.16 **Camera** tab, **Common** menu item

Exposure Settings

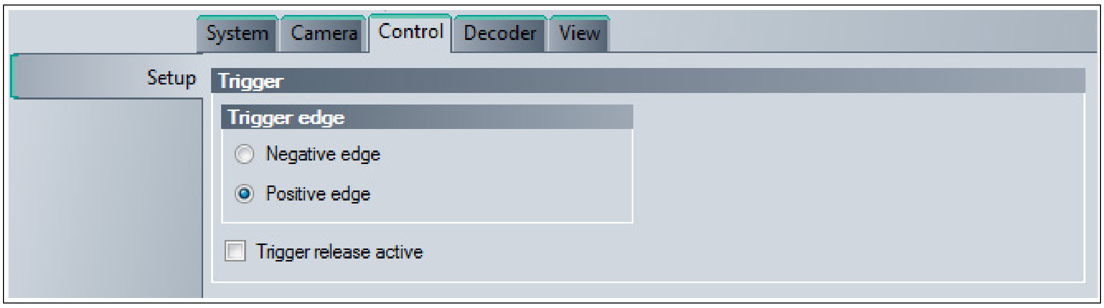
Exposure time	Exposure time setting in µs Min. = 10 µs, max. = 20,000 µs
Gain	Gain setting Min. = 0, max. = 255

Live Image

Start	Starts the live transfer to Vision Configurator
Stop	Stops the live transfer to Vision Configurator
Check every image	Activates the check for readable codes for each image that is transferred during the live transfer to Vision Configurator
Pause	Setting for the pause between two captures in live transfer in ms Min. = 100 ms, max. = 10,000 ms

5.6.3 **Control** Tab

Setup Menu Item



Trigger

Trigger edge

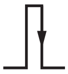
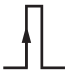
○ Negative edge

● Positive edge

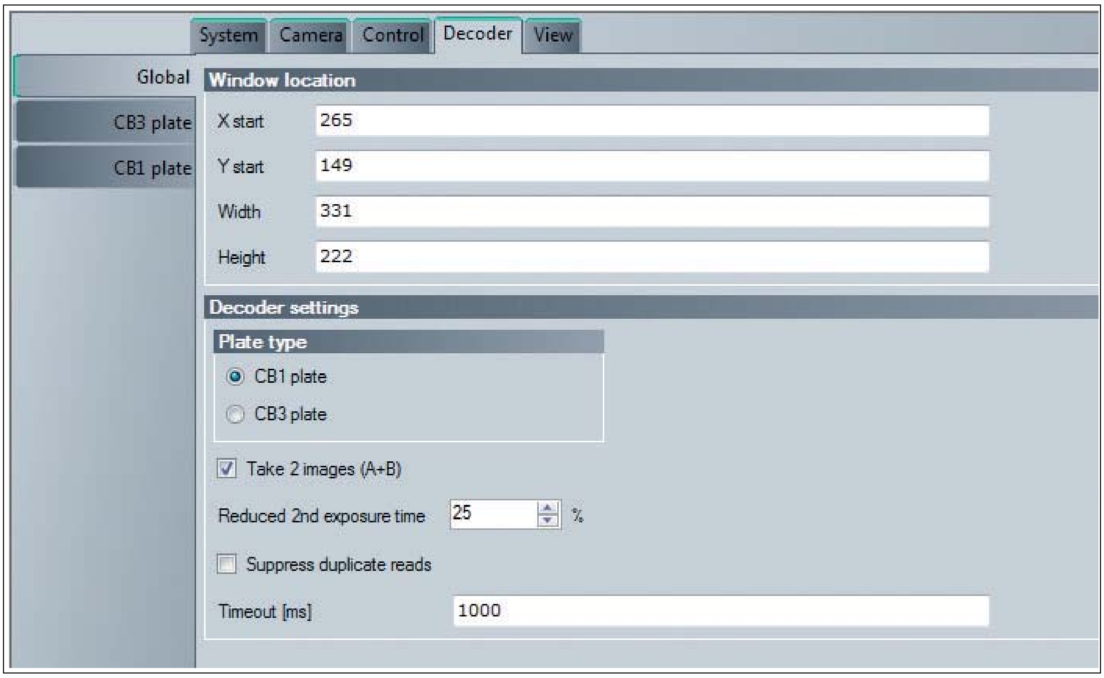
□ Trigger release active

Figure 5.17 **Control** tab, **Setup** menu item

Trigger

Trigger edge	🟩 **Negative edge**: The falling edge activates a trigger  🟩 **Positive edge**: The rising edge activates a trigger 
Trigger release active	The following function is optional: When enabled, a signal at the trigger input will only trigger a read attempt if the OIT has a high signal at a trigger release input.

5.6.4 **Decoder** Tab



Window location

X start 265

Y start 149

Width 331

Height 222

Decoder settings

Plate type

● CB1 plate

○ CB3 plate

☑ Take 2 images (A+B)

Reduced 2nd exposure time 25 %

□ Suppress duplicate reads

Timeout [ms] 1000

Figure 5.18 **Decoder** tab, **Global** menu item

Window Location

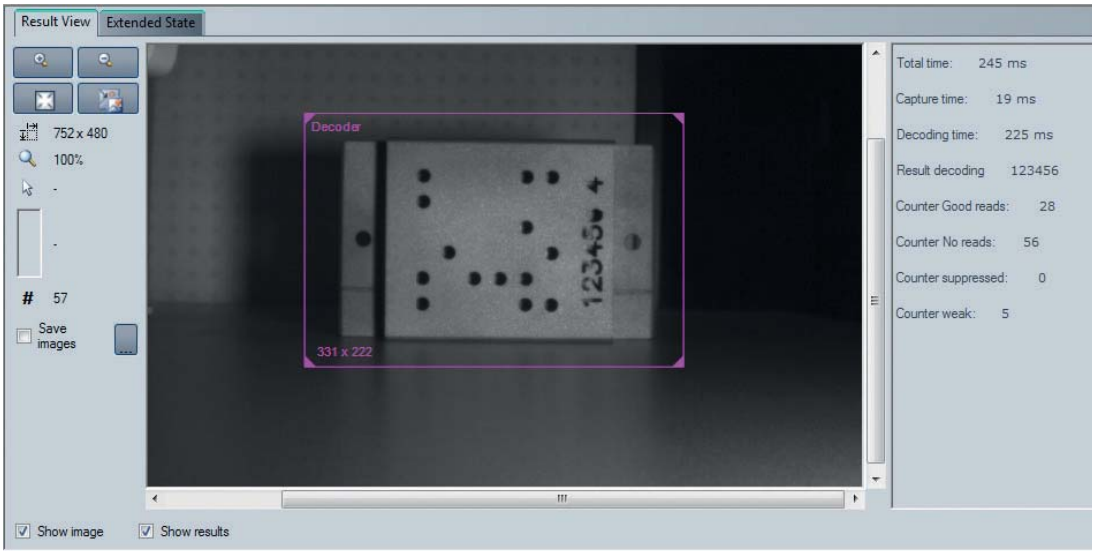
X start	X coordinate of the decoder frame in which the OIT attempts to read a perforated matrix
Y start	Y coordinate of the decoder frame in which the OIT attempts to read a perforated matrix
Width	Width of the decoder frame in which the OIT attempts to read a perforated matrix
Height	Height of the decoder frame in which the OIT attempts to read a perforated matrix

Decoder Settings

Plate type	Selects the type of code sheet to be read ■ **CB3 plate**: Activates the reading of CB3 code sheets ■ **CB1 plate**: Activates the reading of CB1 code sheets
Take 2 images (A+B)	Activates the capture and evaluation of two images (image A + image B) for one read operation. If no code can be read in the first image, the second image is also evaluated. Because you can set different exposure times for the two images, the number of successful readings can be substantially increased.
Reduced 2nd exposure time	Reduces the second exposure time, input in % The percentage refers to the exposure time for a normal capture that has been set in the **Camera** tab under the **Common** menu item.
Suppress duplicate reads	Activates or deactivates duplicate suppression, which prevents the same code being read several times in succession.
Timeout [ms]	Maximum duration for image capture time and decoding of the code in ms. If the operation times out, the reading is considered to have failed.

Tip

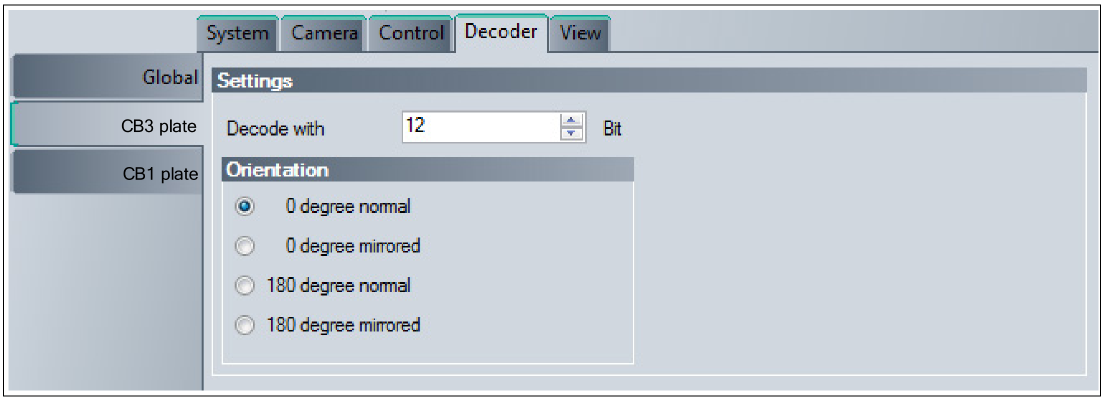
Alternatively, you can use your mouse in the **Result View** to expand, minimize, or move the decoder frame in which the OIT is attempting to read a perforated matrix.



Total time: 245 ms
Capture time: 19 ms
Decoding time: 225 ms
Result decoding 123456
Counter Good reads: 28
Counter No reads: 56
Counter suppressed: 0
Counter weak: 5

Figure 5.19 Decoder frame in **Result View**

CB3 plate Menu Item



Decode with 12 Bit

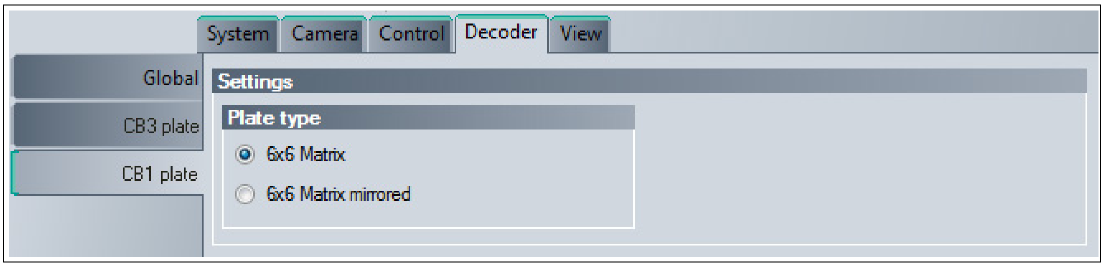
● 0 degree normal
○ 0 degree mirrored
○ 180 degree normal
○ 180 degree mirrored

Figure 5.20 **Decoder** tab, **CB3 plate** menu item

Settings

Decode with	Setting for data bits in the CB3 code sheet Min. = 8, max. = 12
Orientation	Setting for the alignment of the CB3 code sheet ■ **0 degree normal**: normal ■ **0 degree mirrored**: mirrored ■ **180 degree normal**: rotated by 180° ■ **180 degree mirrored**: rotated by 180° and mirrored

CB1 plate Menu Item



● 6x6 Matrix
○ 6x6 Matrix mirrored

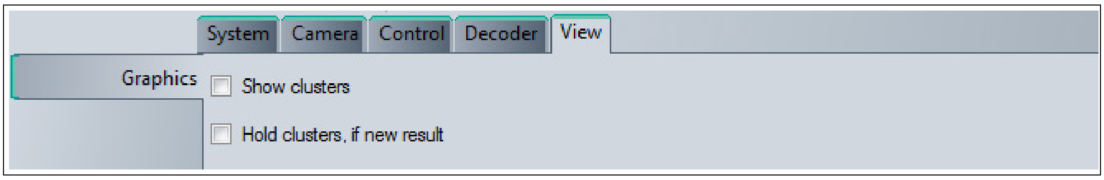
Figure 5.21 **Decoder** tab, **CB1 plate** menu item

Settings

Plate type	Setting for the perforated matrix of the CB1 code sheet 🟩 **6x6 matrix**: 6x6 perforated matrix 🟩 **6x6 matrix mirrored**: 6x6 perforated matrix, mirrored

5.6.5 **View** Tab

Graphics Menu Item



□ Show clusters
□ Hold clusters, if new result

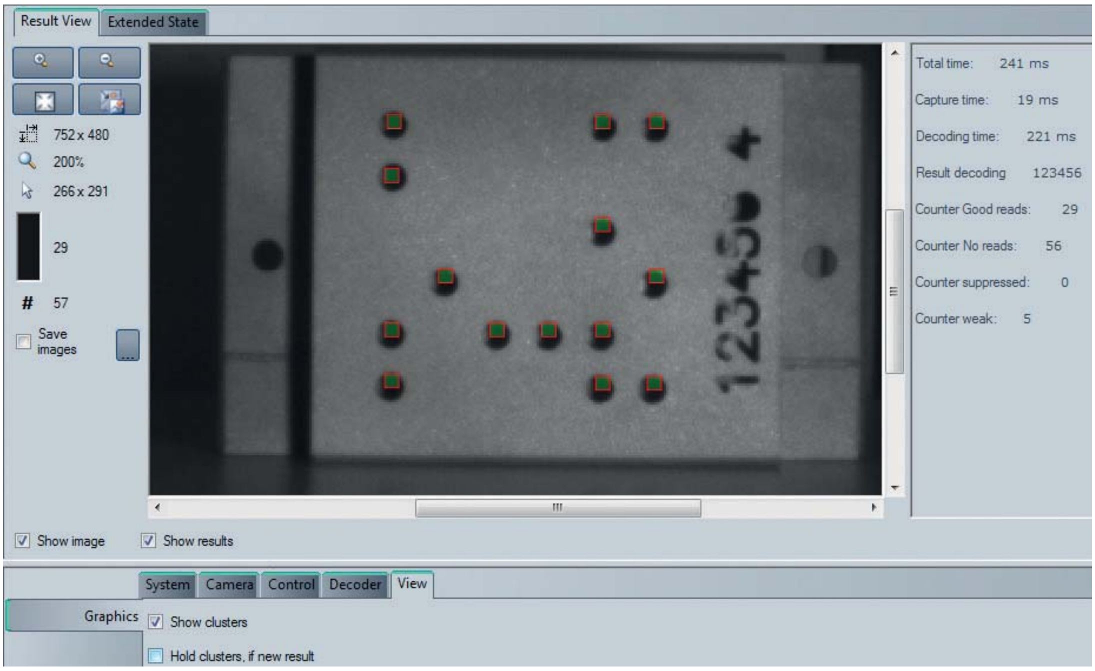
Figure 5.22 **View** tab, **Graphics** menu item

Graphics

Show clusters	Activates optical marking of the detected holes in the **Result View**
Hold clusters, if new result	If activated, the holes that are currently marked in the **Result View** will be retained when a new image is displayed.

Example!

The **Show clusters** option allows you to determine which holes in the code sheet have been detected correctly. Using this information, you can adjust the lighting or the code sheets, for example, until all the holes are detected correctly.



752 x 480

200%

266 x 291

29

57

□ Save images

Total time: 241 ms

Capture time: 19 ms

Decoding time: 221 ms

Result decoding 123456

Counter Good reads: 29

Counter No reads: 56

Counter suppressed: 0

Counter weak: 5

☑ Show image ☑ Show results

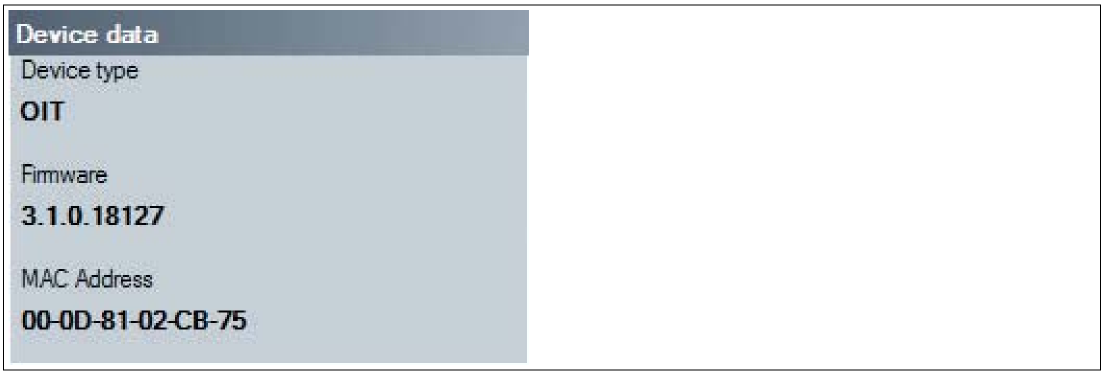
☑ Show clusters

□ Hold clusters, if new result

Figure 5.23 Optical marking of the detected holes

5.7 Device Data

This section shows the connected device type, the firmware version, and the MAC ID.



Device data

Device type

OIT

Firmware

3.1.0.18127

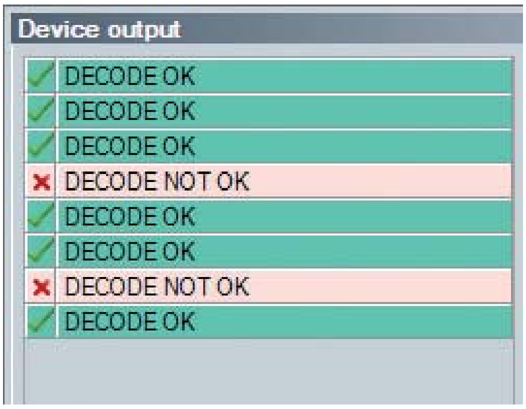
MAC Address

00-0D-81-02-CB-75

Figure 5.24 Sensor data

5.8 Device Output

This area displays an overview of the communication between Vision Configurator and the OIT system.

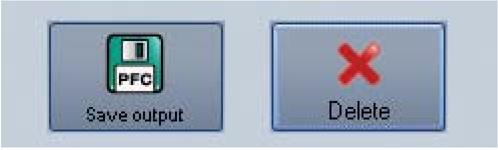


Device output

✓ DECODE OK
✓ DECODE OK
✓ DECODE OK
✗ DECODE NOT OK
✓ DECODE OK
✓ DECODE OK
✗ DECODE NOT OK
✓ DECODE OK

Figure 5.25 Device output

There are two buttons in the area at the bottom.



Save output	Saves the **Sensor Output** area in a text file.
Delete	Deletes the contents of the **Sensor Output** area.

6 Communicating with the OIT System

The following sections point out the different ways to communicate with the OIT system. Of note is the Easy Mode option, since this version requires the least prior knowledge.

- **TCP/IP communication:** With this option, communication between the OIT system and a PLC is communicated and logged in detail. Individual actions must be initiated separately see chapter 6.1.
- **TCP/IP communication with Easy Mode:** The simplest form of connection between a PLC and the OIT system. The connection is made through a simplified set of commands .

6.1 TCP/IP Communication with VSX Protocol

A **.NET 2.0-based** software interface is provided for easy integration with PC software. This software interface takes the form of DLLs and handles the communication with the sensor. Integrate the DLLs into the programming environment and run the programming lines indicated.

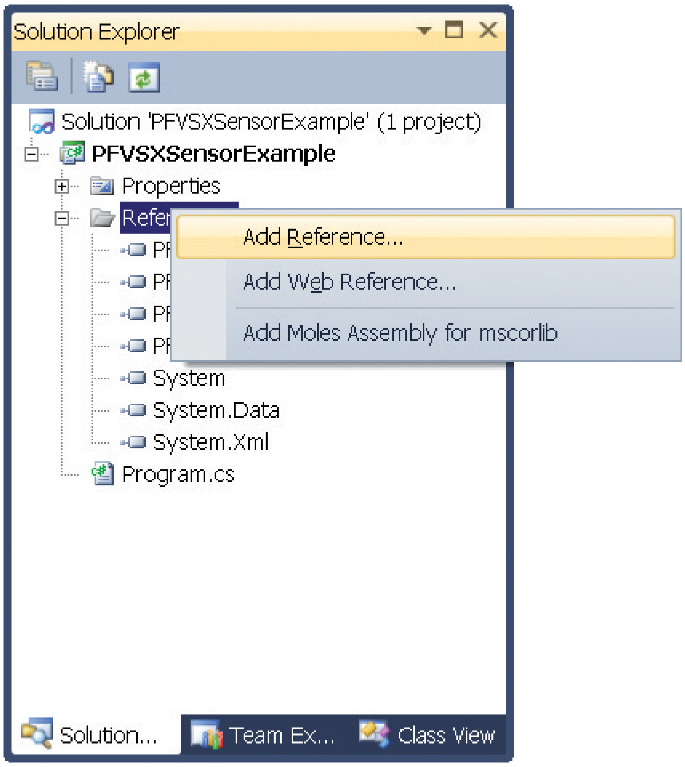


Figure 6.1 Integrating DLLs into the programming environment

All the examples relate to the Visual Studio 2010 programming environment and to the C# programming language.

Example!

Below is a sample program for integrating and triggering the sensor:

```
class Program
{
  static void Main(string[] args)
  {
    PF.Foundation.VsxFactory.PFVsxFactoryVCCustom sensor;
    sensor = new PF.Foundation.VsxFactory.PFVsxFactoryVCCustom();
    sensor.Connect("192.168.2.3", 50005);
    sensor.SetSpecificSingleParameter("Command", "TriggerStart",
"1");
    System.Threading.Thread.Sleep(1000);
    sensor.Disconnect();
  }
}
```

General

The library is used to support the creation of a graphic user interface for sensors that use the VSX protocol. The library connects to the sensor and handles the communication in accordance with the protocol. The user has functions for setting parameters on the sensor, retrieving parameter values from the sensor, and saving and loading whole parameter sets both locally and on the sensor. The user can also receive sensor images.

The library is implemented in C# and requires .NET 2.0 or higher as a minimum.

Make sure that the libraries supplied are in the project's execution folder.

Creating an Object

Create an object to access the library functions.

```
PFVsxFactoryVCCustom _vsxFactory =new PFVsxFactoryVCCustom();
```

Retrieving Parameter Data

Received parameter data is stored as a list in the sensor. Individual items of parameter data from this list can be retrieved using the following function:

```
string GetSpecificSingleParameter(string version, string configId,
string parameterId)
```

Displaying Modified Data

If data is being received by the sensor, this is indicated by the event:

```
event ParameterDataReceived(DataModifier modifier)
```

Library Functions

```
bool Connect(string ip,int port)
```

`ip`: IP of the connected sensor
`port`: Port of the connected sensor
Response: False if the connection could not be created, otherwise true
Opens a connection to a sensor with the IP and port specified.

```
void Disconnect()
```

Disconnects the open connection

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

Response: Returns the connection status

```
void SaveSensorSettings()
```

Saves the current parameter set in the sensor's memory.
After a confirmation has been received from the sensor, a `SaveDataOnSensorReceived` event is triggered.

```
void LoadSensorSettings()
```

Requests a parameter set saved with `SaveSensorSettings` from the sensor.
After a parameter set has been received from the sensor, a `ParameterDataReceived` event with `modifier = LOAD_Data` is triggered.
The data can then be retrieved via the `GetSingleParameter` method.

```
void GetNetworkSettings()
```

Requests the sensor's current connection settings from the sensor.
After this data has been received from the sensor, a GetNetworkReceived event is triggered. This provides the current IP address, network mask, and gateway for the sensor.

```
void GetLogMessages(bool on)
```

on: Switches log messages on
`off`: Switches log messages off
Defines whether or not the sensor sends log messages. If log messages are switched on, a `LogDataReceived` event is triggered each time a log message is received from the sensor.

```
void GetAllParametersFromSensor()
```

Requests the current parameter set from the sensor. After a parameter set has been received from the sensor, a `ParameterDataReceived` event with `modifier = None` is triggered. This indicates that the internal list has been updated with the parameter set. Individual parameters can then be retrieved via `GetSingleParameter.` abgefragt werden.

```
public bool ExistsParameter(string configId, string parameterId)
```

Used to query whether a particular parameter exists on the sensor
`configId`: Configuration ID of a parameter
`parameterId`: Parameter ID of a parameter
Response: True or false, depending on whether the parameter exists

```
string GetSpecificSingleParameter(ushort version, string configId,
string parameterId)
```

`configId`: Configuration ID of a parameter
`parameterId`: Parameter ID of a parameter
Response: The current value of the specified parameter
Retrieves the current value for a parameter from the sensor; details of the version, configId, and parameterId can be taken from the accompanying table. Returns the value of the parameter, or an empty string if the value was not retrieved.

```
void SetSpecificSingleParameter(ushort version, string configId,
string parameterId, string newValue)
```

`configId`: Configuration ID of a parameter
`parameterId`: Parameter ID of a parameter
`newValue`: New value for the parameter
Sets the value for a parameter to newValue; version, configId, and parameterId can be taken from the table in this case.

```
public bool ExistsResultParameter(ushort frameCounter, string
configId, string parameterId)
```

`frameCounter`: Number of the image associated with the result parameter
`configId`: Configuration ID of a parameter
`parameterId`: Parameter ID of a parameter
Response: True or false, depending on whether the parameter exists
Used to query whether a particular result parameter exists for a particular image.

```
string GetResultParameter(ushort frameCounter, string configId,
string parameterId)
```

`configId`: Configuration ID of a parameter
`parameterId`: Parameter ID of a parameter
Response: The current value of the specified parameter
Returns the value of an individual parameter defined by the configuration ID and parameter ID. The value is taken from an internal list and corresponds to the status last received from the sensor.
The receipt of result data for a particular image is indicated by the `event SensorResultDataReceived(ushort frameCounter).` angezeigt.

```
Bitmap GetImage(ushort frameCounter)
```

`frameCounter`: Number of a received image
Response: The image associated with the image number
Returns an image previously received from the sensor. The receipt of an image with a particular number is indicated by the `event ImageReceived(ushort frameCounter, Bitmap image).` angezeigt.

```
IList<ElementResult> GetResultList(ushort frameCounter)
```

`frameCounter`: Number of the image associated with the graphics
Response: List of overlay graphics for the specified image

```
IList<ElementShapeBase> GetShapes(ushort frameCounter)
```

`frameCounter`: Number of the image associated with the graphics
Response: List of overlay graphics for the specified image
Returns a list of graphics for an image overlay. The receipt of these graphics is indicated by the `event ShapeDataReceived(ushort frameCounter, List<IElement> shapeList).` angezeigt. The individual elements in the list are of the `ElementShapeBase`type.
This class has the following attributes:
`PointF ShapeLocation`: Coordinates of the top left corner of the graphic in the image
`Color ForeColor`: Color of the graphic
`string Type`: Type of graphic, either type="Rectangle" or type="Text". Depending on the type, the graphic can be parsed in `ElementShapeRectangle` or `ElementShapeText` and then has the following additional attributes:
`ElementShapeRectangle:`
Size Size: The size of the rectangular graphic
`ElementShapeText:`
stringText: The text in the text graphic

```
void ResetSensor()
```

Resets all the sensor's parameters to their factory default settings. After a confirmation has been received from the sensor, a `ParameterDataReceived` event with `modifier = LOAD_DEFAULT_DATA` is triggered.

```
void SaveSettingsToFile(string filename)
```

Saves the current parameter set to the specified file. The current parameter set is retrieved from the sensor and saved after receipt. After a successful save, a `SaveDataOnHdd` event is triggered. If an error occurs during the save, an `InternalError` event with `ErrorType = SAVE_FILE_ERROR` is triggered.
`filename:` Valid path and file name

```
bool SetAllParameters(string filename)
```

Loads a parameter set from the specified file and sends the parameters to the sensor. After the sensor has acknowledged receipt of the parameter set, a `ParameterDataReceived` event with `modifier = NONE` is triggered. If the data set cannot be loaded correctly, an `InternalError` event with `ErrorType = LOAD_FILE_ERROR` is triggered.
`filename`: Valid path and file name

```
void SetNetworkSettings(string·ipAddress, string networkMask, string gateway)
```

`ipAddress`: Valid IP address
`networkMask`: Valid network mask
`gateway`: Valid gateway
Converts the network parameters on the sensor. The connection to the sensor is then disconnected and a `DisconnectReceived` event is triggered. With `Connect` the connection must be reestablished with the new IP.

```
void SendImage(Bitmap image)
```

Sends an image to the sensor. This function is not supported by all device types.
`image`: Image to be sent

```
void SendVsxFile(string filepath)
```

`filepath`: Path and file name of a valid VSX file
Sends the content of a file to the sensor. This must comply with the VSX standard. If the data cannot be loaded properly or does not correspond to the VSX syntax, an`InternalError` event with `ErrorType = LOAD_FILE_ERROR` is triggered.

```
string SensorName { get; }
```

Returns the name of the sensor.

```
float SensorVsxVersion { get; }
```

Returns the VSX version installed on the sensor.

Library Events

```
event ParameterDataReceived(DataModifier modifier)
```

This is always triggered if parameter data has been received from the sensor.
This is the case after the following function call:
`LoadSensorSettings·(MODIFIER = LOAD_DATA)`
`GetAllParametersFromSensor·(MODIFIER = NONE)`
`SetSingleParameter·(MODIFIER = NONE)`
`ResetSensor·(MODIFIER = LOAD_DEFAULT_DATA)`
`SetAllParameters·(MODIFIER = NONE)`
Whenever this event is triggered, the internal list of parameters has been updated and the individual parameter values should be retrieved with `GetSingleParameter.` abgefragt werden.

```
event SensorResultDataReceived(ushort framecounter)
```

This is triggered when result data is received from the sensor. The parameter is an image number that allows the data to be assigned to the corresponding image. The individual result data can be retrieved with `GetResultParameter.` abgefragt werden.

```
event SingleDataReceived(string configId, string parameterId, string value)
```

This is triggered after `SetSingleParameter` if this one value only has been changed on the sensor.

```
event AcceptReceived()
```

This is triggered after `SetSingleParameter` if the value previously set has been successfully changed on the sensor.

```
event SensorInformationDataReceived(string type, string version, string macAddress)
```

This is triggered when any data is received from the sensor. Details of the sensor type, its firmware version, and its MAC ID are transmitted.

```
event DisconnectReceived(string errorMessage)
```

This is triggered if the connection is not disconnected with `Disconnect.` getrennt wurde. A description of the reason for the lost connection is transmitted.

```
event SaveDataOnSensorReceived()
```

This is triggered after `SaveSensorSettings` has been called successfully.

```
event GetNetworkReceived(string ipAddress, string mask, string
gateway)
```

This is triggered after the data requested from the sensor with `GetNetworkSettings` has been received. The IP address, the network mask, and the sensor's gateway are transmitted.

```
event LogDataReceived(string logData, LogMessageTypes logType)
```

This is triggered when log data has been received from the sensor. The data and log data type are transmitted as parameters.

```
event ShapeDataReceived(ushort frameCounter)
```

This is triggered when graphics for the image overlay have been received. The image number specifies the image with which the graphics are associated. A list of all graphics can then be retrieved using the `GetShapes(ushort frameCounter)` function.

The following event is supported by laser triangulation sensors only:

```
event LineDataReceived(ushort frameCounter, LineMulti lines, ushort
status)
```

This is triggered when line data has been received from the sensor. In addition to the line data, the image number is issued so that the data can be assigned to the appropriate image. A status is issued that reflects the sensor's hardware inputs and outputs.

```
event ImageReceived(ushort frameCounter)
```

This is triggered when an image has been received from the sensor. The image number is transmitted as a parameter. The image can then be retrieved using the `GetImage(ushort frameCounter)` function.

```
event SaveDataOnHddReceived()
```

This is triggered when the current parameter set has been successfully saved to a file.

```
event ErrorReceived(string id, string name)
```

This is triggered when an error has occurred in the sensor software. The parameters contain more details about this error.

```
event InternalError(ErrorTypes errorType, string errorMessage)
```

This is triggered when an internal error has occurred. This happens in the following cases:

- When calling `GetSingleParameter (ErrorType = PARAMETER_NOT_FOUND)` if the required parameter has not been found in the internal list
- When calling `SetSingleParameter (ErrorType = PARAMETER_NOT_FOUND)` if the parameter to be set is not present
- When calling `SaveSettingsToFile (ErrorType = SAVE_FILE_ERROR)` if the parameter data could not be saved to a file
- When calling `LoadSettingsFromFile (ErrorType = LOAD_FILE_ERROR)` if the parameter file could not be loaded
- When calling `SetAllParameters (ErrorType = LOAD_FILE_ERROR )` if the parameter data could be loaded from the file or if the file contains invalid data
- In all cases where `ParameterDataReceived` or `SensorResultDataReceived` would be triggered but the sensor data received is incorrect `(ErrorType = INVALID_DATA_RECEIVED).`
- If incorrect log data is received from the sensor `(ErrorType = INVALID_DATA_RECEIVED)`
- When calling `SendVsxFile (ErrorType = LOAD_FILE_ERROR)` if the file cannot be loaded or does not correspond to the VSX syntax

The `errorMessage` parameter contains a more detailed description of the respective error.

```
enum·ErrorTypes{ INVALID_DATA_RECEIVED, SAVE_FILE_ERROR,
LOAD_FILE_ERROR, PARAMETER_NOT_FOUND }
```

Siehe event `InternalError`.

```
enum·DataModifier{ NONE, LOAD_DATA, LOAD_DEFAULT_DATA }
```

See event `ParameterDataReceived`.

```
enum LogMessageTypes { DEBUG, INFO, RESULT_OK, RESULT_NOT_OK,
WARNING, ERROR, CRITICAL, ASSERT }
```

See event `LogDataReceived`.

Result Parameters

There are two types of result parameters:

- Result data (PF.Foundation.Protocol.XML.ElementResult)
- Shape data (PF.Foundation.Protocol.XML.ElementShapeBase)

The shape data is drawn directly on the corresponding image. The data consists of geometric shapes (e.g. ElementShapeRectangle) or labels (e.g. ElementShapeText). It also contains information regarding the position and size.

Result data is purely text data. This data contains the range of results, decoding time, and the code type in any active window.

Result Data

Event	Parameter ID	Value	Description
Trigger	TimeTotal	ms	Total time between the trigger and the end of the decoding
	TimeCapture	ms	Image capture time
	TimeDecode	ms	Decoding time
	ResDecode	String	Decoding result
	CounterGood		Number of successful read operations
	CounterBad		Number of failed read operations
	CounterBadSuppressed		Number of suppressed read operations
	CounterWeak		Number of successful read operations

Table 6.1 The result data is transferred by the "SensorResultDataReceived" event. The "GetResultList" and "GetResultParameter" processes provide access to the result data.

Shape Data

Event	Parameter ID	Text	Description
ElementShapeText	ImageType	Good Error	Image type
	ErrImgListNr		Number of the faulty image
	ExposureTime	ms	Exposure time of the image
	Res		Result
	ImageNr		Image number
ElementShapeRectangle	none		The number of rectangles that display the position of the detected holes

6.2 TCP/IP Communication with Easy Mode

The TCP/IP protocol enables communication between the PLC and OIT system.

For communication in Easy Mode, the host system connects to the OIT system via **port address: 50100**. The OIT system initially waits for a trigger signal when starting communication in Easy Mode. There are two different trigger signals:

Trigger signal from an external trigger sensor (positive edge)

or

Trigger signal via the PLC telegram.

Once one of the two trigger signals has been triggered, the OIT system captures an image. After the image has been successfully captured, the OIT system decodes the data stored on the read-only tag and sends this data to the PLC. If the read-only tag was not successfully read, the OIT system sends a "read error" to the PLC. The OIT system then waits for another trigger signal.

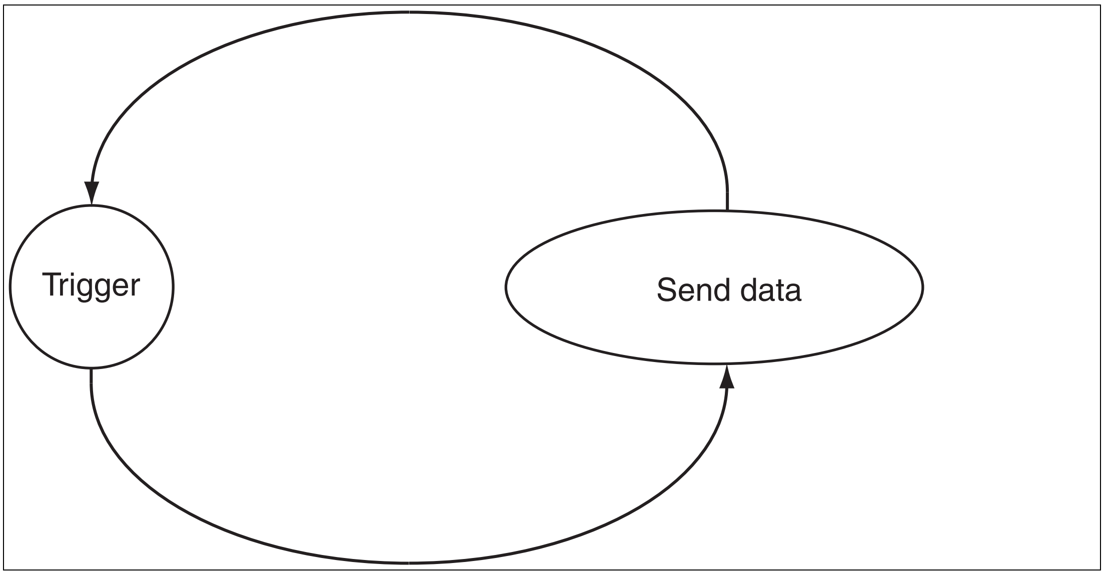


Figure 6.2 Easy Mode flow chart

Telegram structure:

The data telegram has a fixed length. The code read on the read-only tag is sent to the PLC in ASCII characters. If the code read on the read-only tag has less than six characters, leading zeros are added to the code. For example, the number 123 is sent as 000123.

Easy Mode byte format:

The outputs specified with 4 bytes are saved in the "little-endian" format and will also be output as such. For example, the number **0x04030201** is stored in the memory as **0x01 0x02 0x03 0x04** and is also output via Easy Mode as such.

Read Result Telegram

Byte 0	Start character	#(23 hex)
Byte 1–4	Last evaluation time, data in ms	
Byte 5–8	Number of good readings	
Byte 9–12	Number of bad readings	
Byte 13–16	Number of suppressed duplicate readings The counter is incremented if the box next to **Suppress duplicate reads** in Vision Configurator is checked. If you send a read command via Easy Mode, the response will contain a read code—even if it is a duplicate read. It is also counted as suppressed. Additionally in this case, the bad read counter is incremented and not the good read counter.	
Byte 17–20	Number of readings within the trip value range	
Byte 21–26	Read result	Result or NOREAD
Byte 27–30	Status	See status table
Byte 31	Checksum	Value x
Byte 32	End character 1	CR (0D hex)
Byte 33	End character 2	LF (0A hex)

Status Table for Bytes 27–30

Byte				Group description	Subgroup description
27	28	29	30		
bin	bin	bin	hex		
Bit 0					
	Bit 5			Function could not be started or is not running	Decoder not active
		Bit 0		OIT warnings (The read-only tag could still be read)	Camera image too bright
		Bit 1			Camera image too dark
		Bit 2			Too many structures in image
		Bit 3			The distance between the read-only tag and OIT system is too great
		Bit 4			The distance between the read-only tag and OIT system is too small
		Bit 5			Not enough contrast
		Bit 6			The read-only tag is outside of the detection/capture range
			0x01	OIT status (The read-only tag could no longer be read)	No hole structures found
			0x02		Not enough hole structures, first loop
			0x03		Not enough hole structures, second loop
			0x04		No check marks found, Check 1
			0x05		Decoding not possible, Check 1
			0x06		No check marks found, Check 2
			0x07		Poor contrast
			0x08		No check marks found, Check 3
			0x0A		No check marks found, Check 4
			0x0B		Decoding not possible, Check 2
			0x0F		Checksum error
			0x2A		Timeout
			0x31		Poor read-only tag (too many possible hole structures)
			0x32		Timeout while determining the corner points
			0x33		Timeout while decoding
			0x99		Maximum repeat read reached Double code-reading
			0x9A		Disconnection during the measuring process
			0xFE		No usable data from the PLC

Calculating the Checksum

The individual bytes of the data telegram (byte 0 to byte 30) are calculated with the XOR function. The resulting value is then transmitted as a checksum. The receiver of this data telegram can also calculate a checksum using the data and compare this checksum with the checksum transmitted by the sender. If the two checksums are not identical, a transmission error has occurred.

Definition x = unsigned byte

X =	Byte 0	XOR	Byte 1
X =	X	XOR	Byte 2
X =	X	XOR	Byte 3
...			
X =	X	XOR	Byte 29
X =	X	XOR	Byte 30

Byte 31 = checksum value X

Example telegram structure:

Response to the command **#R<CR><LF>** or to a positive edge at the trigger input in hex format. In the example here for a successful reading:

23 90 00 00 00 09 00 00 00 04 00 00 00 02 00 00 00 09 00 00 00 30 30 30 30 35 33 00 00 00 00 B3 0D 0A

Start character	23 (in ASCII: #)
Last evaluation time	90 00 00 00
Number of good readings	09 00 00 00
Number of bad readings	04 00 00 00
Number of suppressed readings	02 00 00 00
Number of readings within the trip value range	09 00 00 00
Read result	30 30 30 30 35 33 (in ASCII: 000053)
Status	00 00 00 00
Checksum	B3
End character 1	0D (in ASCII: CR)
End character 2	0A (in ASCII: LF)

Trigger Telegram (via Port 50100 for Software Triggering)

Byte 0	Start character	#(23 hex)
Byte 1	Command: read code	R (52 hex)
Byte 2	End character 1	CR (0D hex)
Byte 3	End character 2	LF (0A hex)

7 Troubleshooting

Note!

Do not repair, modify, or manipulate the device.

If there is a defect, the device must be repaired by Pepperl+Fuchs.

Fault Repair

Fault	Possible cause	Remedy
Communication with Vision Configurator is not possible	The IP address is configured incorrectly.	The factory default IP address is 172.24.55.162.
Image is not displayed in Vision Configurator	The transfer of image captures to Vision Configurator is not activated.	Enable the **Image transfer active** option in the **System** tab, under the **Image** menu item. See chapter 5.6.1
Code sheets cannot be read	The decoder is set to the wrong code sheet.	Select the correct type in the **Decoder** tab, under the **Global** menu item. See chapter 5.6.4
	The holes in the code sheet are located outside of the decoder frame in which the OIT is attempting to detect a perforated matrix.	Adjust the settings for the decoder frame in the **Decoder** tab, under the **Global** menu item. See chapter 5.6.4
	The image is too dark.	Change the settings for exposure and gain on the **Camera** tab, under the **Common** menu item. See chapter 5.6.2
	The lighting unit is set for the reading of a different code sheet.	Adjust the lighting unit.

Contacting the Service Center

1. Use Vision Configurator to save the image capture from a failed reading. See chapter 5.4 Save the image capture once with the result data and once without.
2. Note the firmware version of the OIT. See chapter 5.7
3. Contact the service center and have the saved image captures and the firmware version ready.

Disposing of device, packaging, and possibly contained batteries must be in compliance with the applicable laws and guidelines of the respective country.

FACTORY AUTOMATION – SENSING YOUR NEEDS

Worldwide Headquarters
Pepperl+Fuchs GmbH
68307 Mannheim · Germany
Tel. +49 621 776-0
E-mail: info@de.pepperl-fuchs.com

USA Headquarters
Pepperl+Fuchs Inc.
Twinsburg, Ohio 44087 · USA
Tel. +1 330 4253555
E-mail: sales@us.pepperl-fuchs.com

Asia Pacific Headquarters
Pepperl+Fuchs Pte Ltd.
Company Registration No. 199003130E
Singapore 139942
Tel. +65 67799091
E-mail: sales@sg.pepperl-fuchs.com

www.pepperl-fuchs.com

PEPPERL+FUCHS
SENSING YOUR NEEDS