OIT500-F113-B17-CB

High-Temperature Identification System

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

2023-10

2023-10

1 Introduction

1.1 Content of this Document

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

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

Note

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

Note

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

The documentation comprises the following parts:

- This document
- Datasheet

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

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

1.2 Target Group, Personnel

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

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

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

1.3 Symbols Used

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

Warning Messages

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

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

Danger!

This symbol indicates an imminent danger.

Non-observance will result in personal injury or death.

Warning!

This symbol indicates a possible fault or danger.

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

Caution!

This symbol indicates a possible fault.

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

Informative Symbols

Note

This symbol brings important information to your attention.

Action

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

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.

PROFINET®, PROFIBUS®: Trademarks of PROFIBUS Nutzerorganisation e.V. (PNO)

SIMATIC, TIA Portal: Trademarks of SIEMENS AG

2 Product Description

2.1 Functional Description

OIT500-F113-B17-CB (referred to as the OIT system from now on) uses infrared lighting and evaluates special code sheets that feature hole 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. For the OIT system you can use CB1, CB2 and CB3 code sheets.. 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 hole pattern punched into the code sheets features an ID that is detected and processed by the OIT system. The read result is forwarded to the controller via PROFINET or to a computer using 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 LED Indicator

The LED indicator shows all important system and status information.

The nine LEDs mounted on the side of the device can be used to read different information.

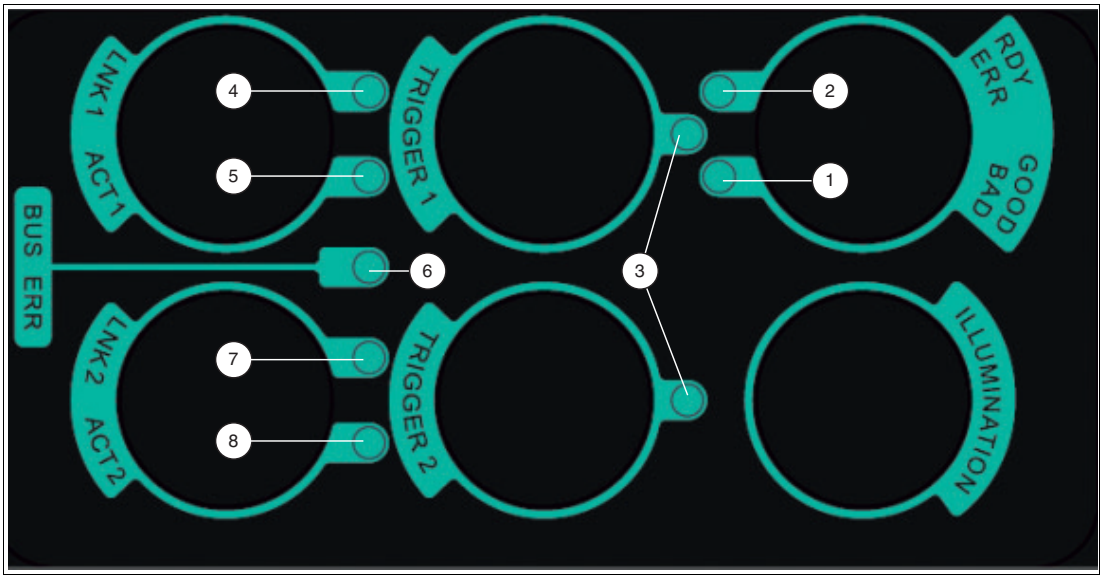


Figure 2.2 LED Indicator

Position	Description	Function
1	**GOOD BAD**	Reading successful (green) or failed (red)
2	**RDY ERR**	The LED flashes while initializing. During this time, no network connection to the OIT can be established. The LED lights up as soon as the OIT is ready for operation
3	**TRIGGER 1 TRIGGER 2**	Lights up yellow when a connected trigger sensor is activated
4	**LINK 1**	Lights up green when an Ethernet cable is connected
5	**ACT 1**	Flashes orange: Data transfer on the PROFINET channel active
6	**BUS ERR**	Flashes red if there is no PROFINET connection available
7	**LINK 2**	Lights up green when an Ethernet cable is connected
8	**ACT 2**	Flashes orange: Data transfer on the PROFINET channel active

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

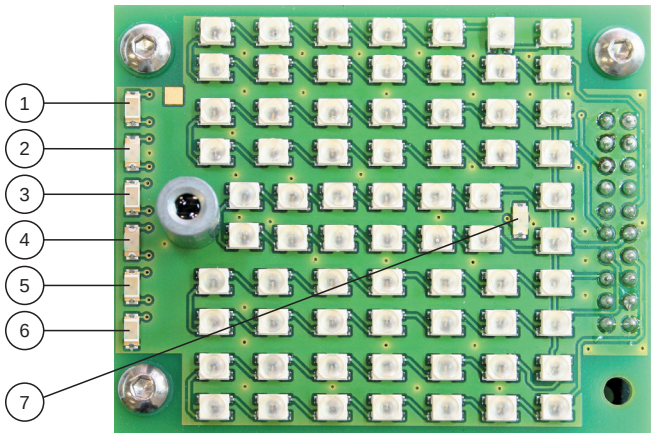


Figure 2.3 Lighting unit of the OIT system

Position	Description	Function
1	**Code OK**	Reading successful
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 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

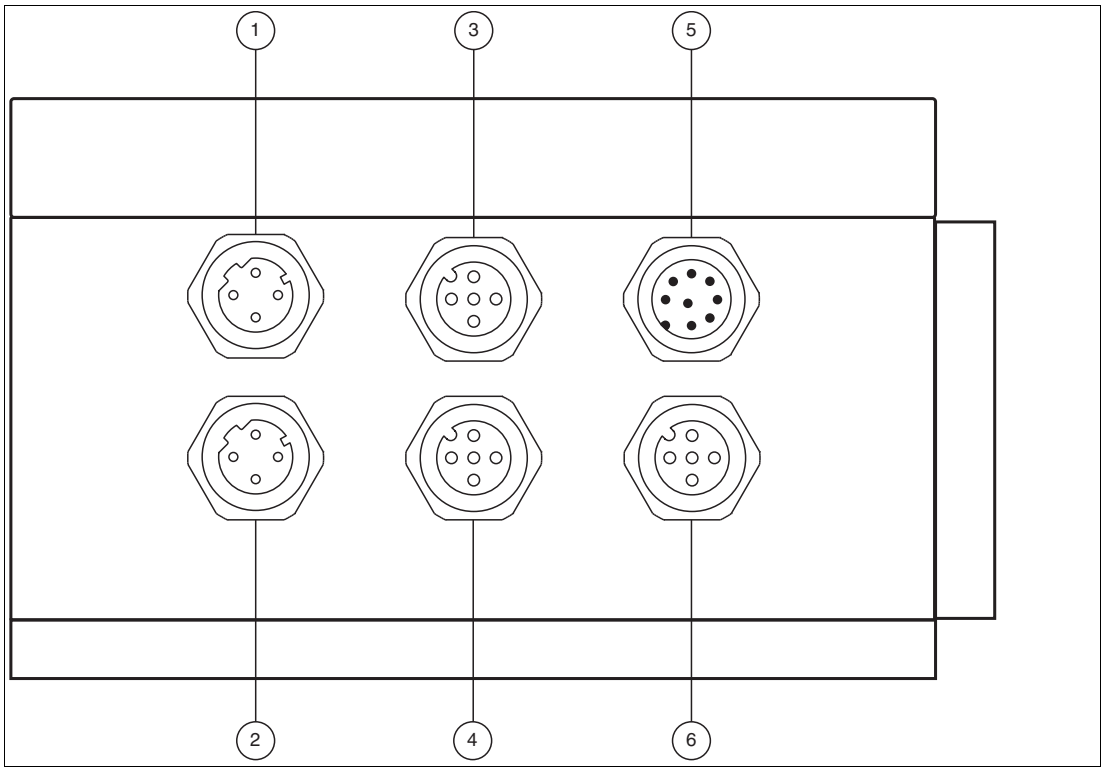


Figure 2.4 Connections

1. PROFINET/EtherNet TCP/IP connection 1
2. PROFINET/EtherNet TCP/IP connection 2
3. Trigger connection 1
4. Trigger connection 2
5. Power supply
6. Connection for external lighting

Pinout for PROFINET connection 1 and 2

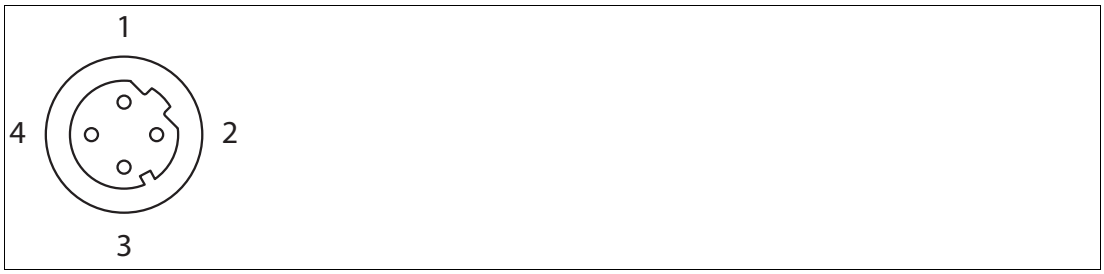


Figure 2.5 Pinout for PROFINET connection

1. Tx +
2. Rx +
3. Tx -
4. Rx -

2023-10

Pinout for Trigger Connection 1 and 2

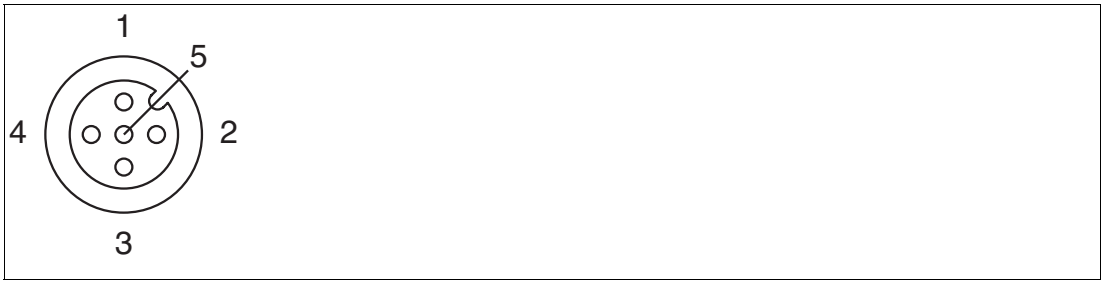


Figure 2.6 Pinout for Trigger Connection

1. 24 V power supply
2. Not assigned
3. Ground
4. Trigger signal
5. Not assigned

Pinout for power supply

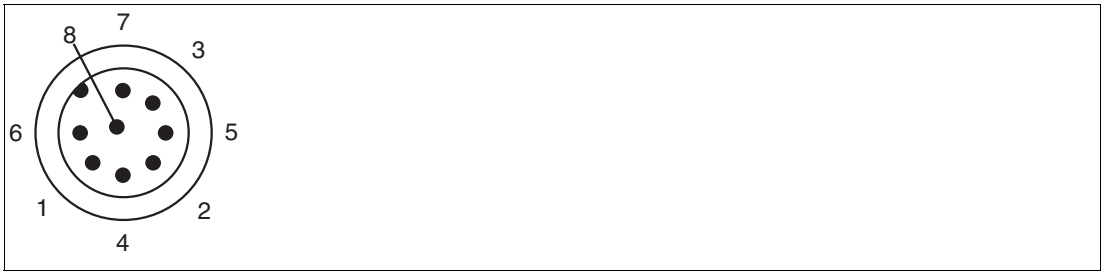


Figure 2.7 Pinout for power supply

1. I/O 1
2. 24 V power supply
3. Not assigned
4. Not assigned
5. I/O 2
6. I/O 3
7. Ground
8. I/O 4

Pinout for External Lighting

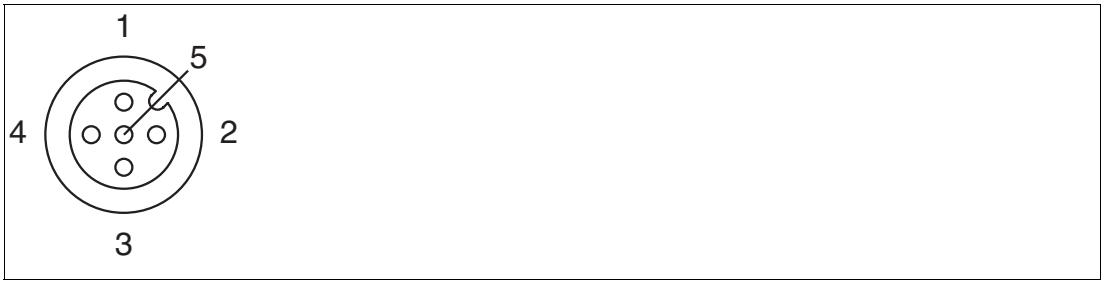


Figure 2.8 Pinout for external lighting

1. 24 V power supply
2. Not assigned
3. Ground
4. Lighting control
5. Not assigned

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-B17-CB
- Mounting base (preassembled 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

Description	Description
V19-G-2M-PUR-ABG	Single-ended female cordset, M12, 8-pin, shielded, PUR cable
V19-G-ABG-PG9	M12 single-ended female cordset, 8-pin, shielded, field-attachable

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

Model number	Description	Cable dia.
V1S-G-BK	4-pin M12 plug, straight	4 ... 6 mm
V1S-G-ABG-PG9	4-pin M12 plug, straight	5 ... 8 mm
V1S-G PG9	4-pin M12 plug, straight	6 ... 8 mm
V1S-G-Q2	4-pin M12 plug, straight	4 ... 8 mm
V1S-G-Q3	4-pin M12 plug, straight	3.5 ... 6 mm

Note

Other lengths on request.

2023-10

2.6.3 Network Cable

Note

Network Connection with Degree of Protection IP65

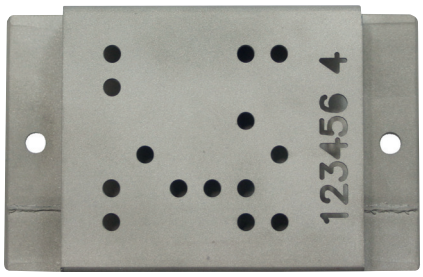
The network connection on the OIT has degree of protection IP65.

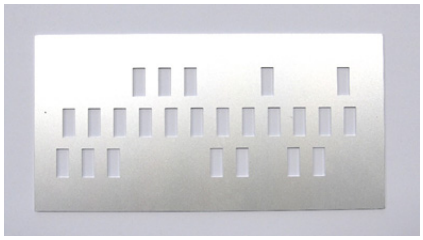
The OIT is connected to the network via a network connector.

Description	Description
V1SD-G-2M-PUR-ABG-V45-G	2 m cordset, M12 to RJ45 PUR cable, 4-pin, CAT5e
V1SD-G-5M-PUR-ABG-V45-G	5 m cordset, M12 to RJ45 PUR cable, 4-pin, CAT5e
V1SD-G-10M-PUR-ABG-V45-G	10 m cordset, M12 to RJ45 PUR cable, 4-pin, CAT5e
V1SD-G-2M-PUR-ABG-V1SD-G	2 m cordset, M12 to M12, PUR cable, 4-pin, CAT5e
V1SD-G-5M-PUR-ABG-V1SD-G	5 m cordset, M12 to M12, PUR cable, 4-pin, CAT5e
V1SD-G-10M-PUR-ABG-V1SD-G	10 m cordset, M12 to M12, PUR cable, 4-pin, CAT5e

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 the following code sheets for the OIT system:

Order designation	Code sheet	Description
OIC-C10V2A-CB1-xxxxxx-yyyyyy		xxxxxx: starting value yyyyyy: number The final value is determined as follows: Final value = start value + number - 1 Note: The code plate numbers are always increased by the value 1 from the start value to the final value.

Order designation	Code sheet	Description
OIC-C11V4A-CB2		Small read-only tag for optical high-temperature identification system, stainless steel. Value range: 4-digit number between 1 and 4095, puls a 6 bit check digit.
-		The CB3 code sheets are 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

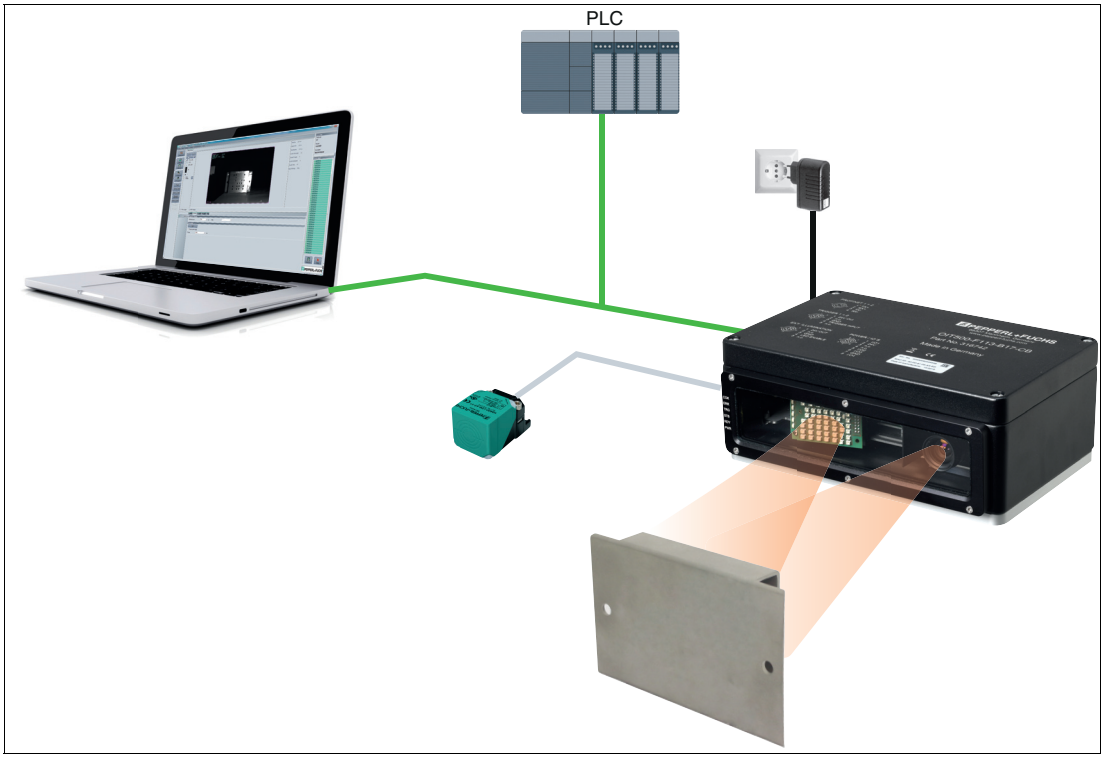


Figure 3.1 OIT system, structural principle

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:

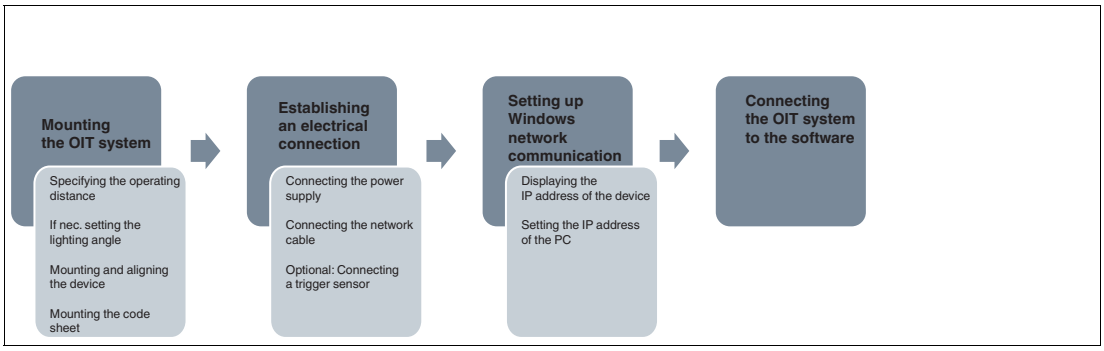


Figure 3.2 Schedule for initial commissioning

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.

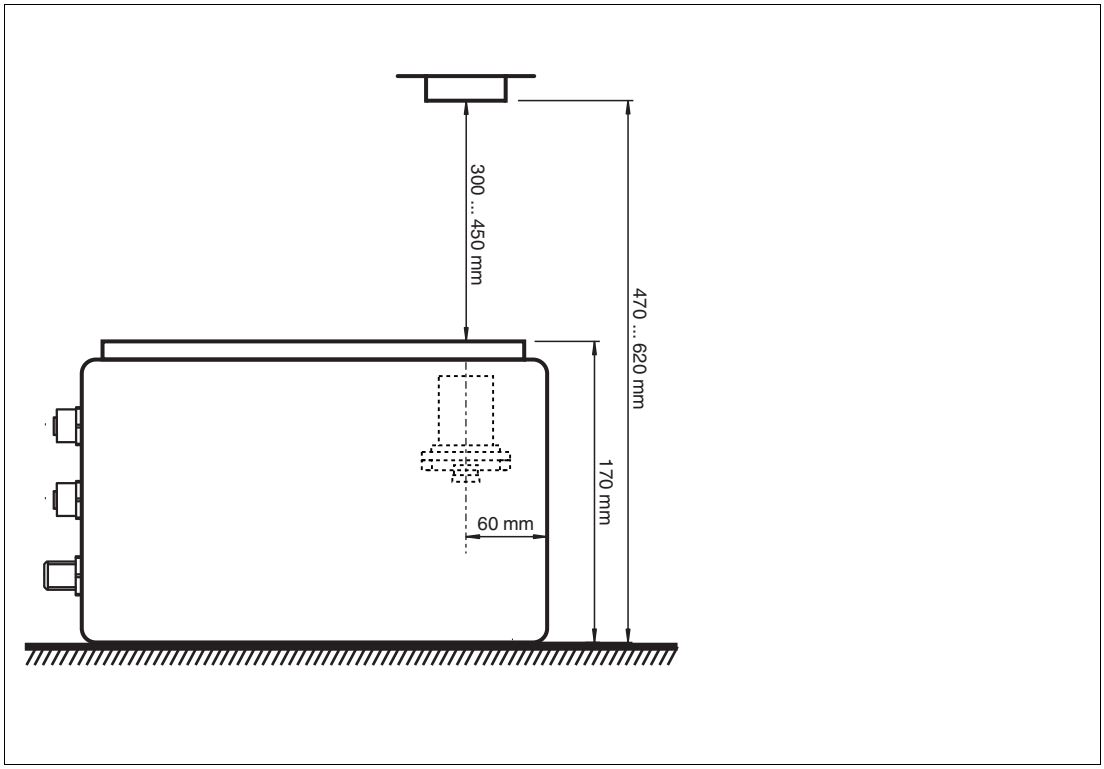


Figure 3.3 Operating distance (top view)

2023-10

PEPPERL+FUCHS

Permissible Code Sheet Displacement for CB1 Code Sheets

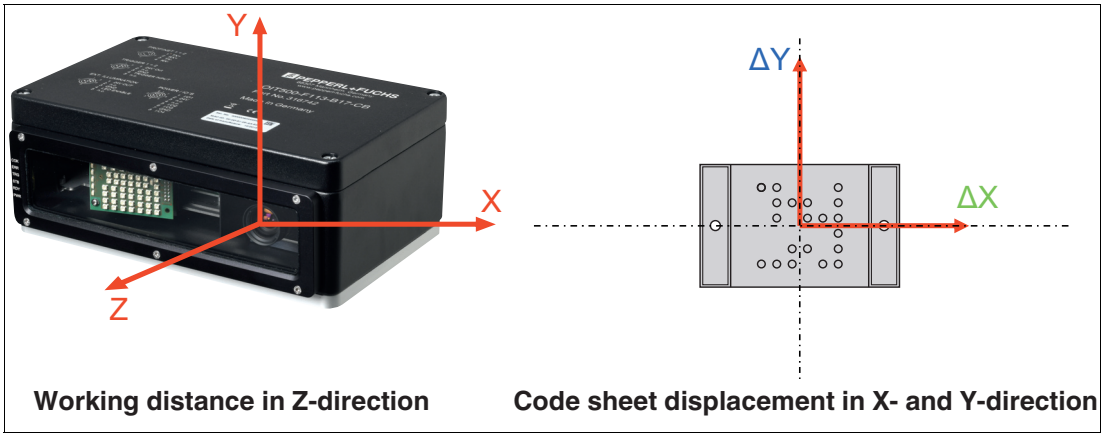


Figure 3.4 X/Y direction for the displacement of **OIC-xxxx-CB1** Code sheet.

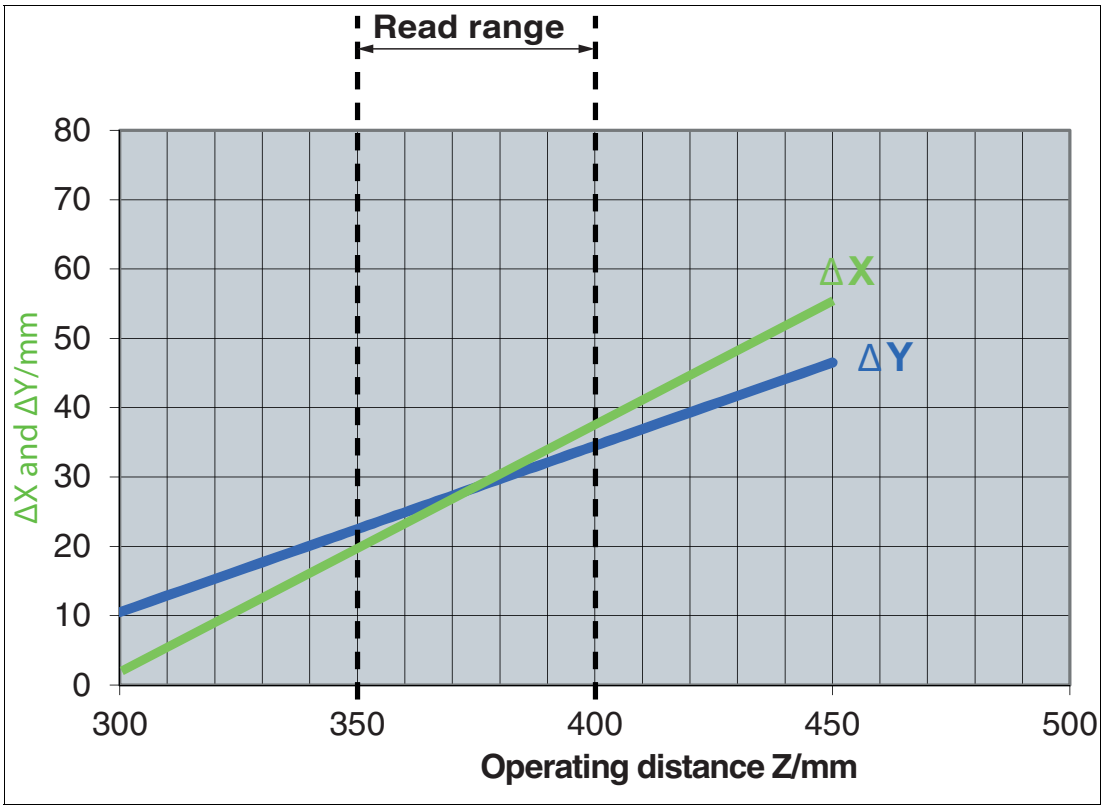
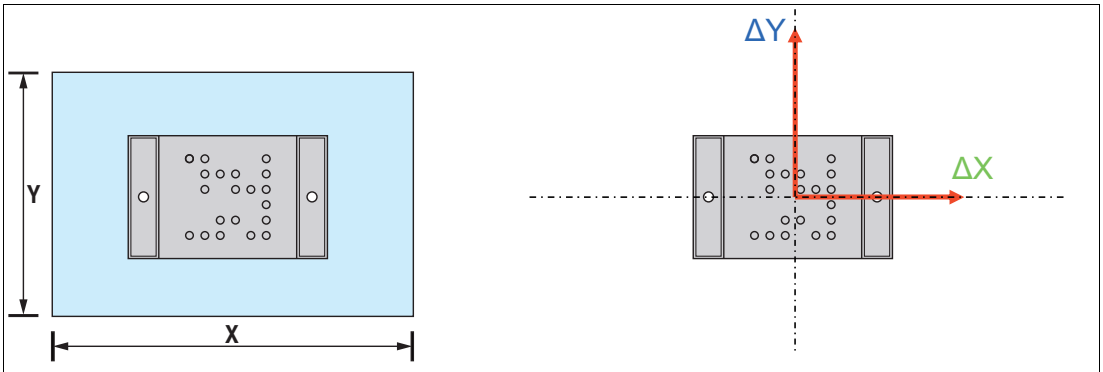


Figure 3.5 Permissible displacement of read-only tag ± ΔX and ± ΔY depending on reading distance Z

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.

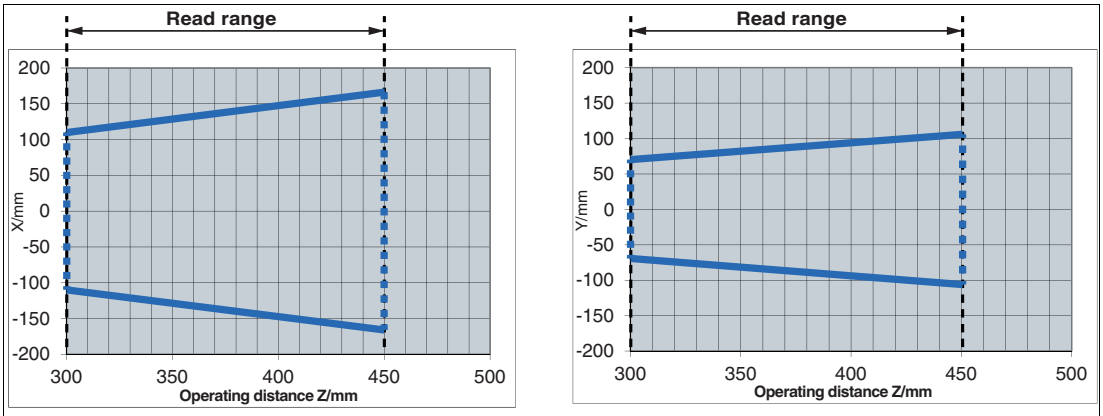
Permissible Reading Window Displacement for CB1 Code Sheets



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



Reading window in the X direction **Reading window in the Y direction**

Note

CB3 code sheets

There is the option to use the CB3 code sheet.

2023-10

PEPPERL+FUCHS

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.

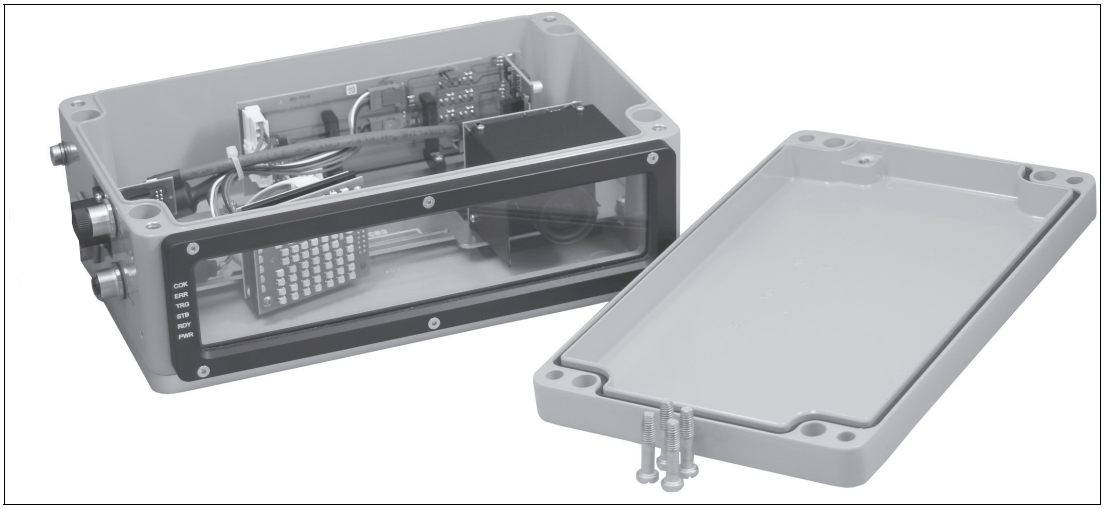


Figure 3.7 Remove the enclosure cover, basic diagram

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.

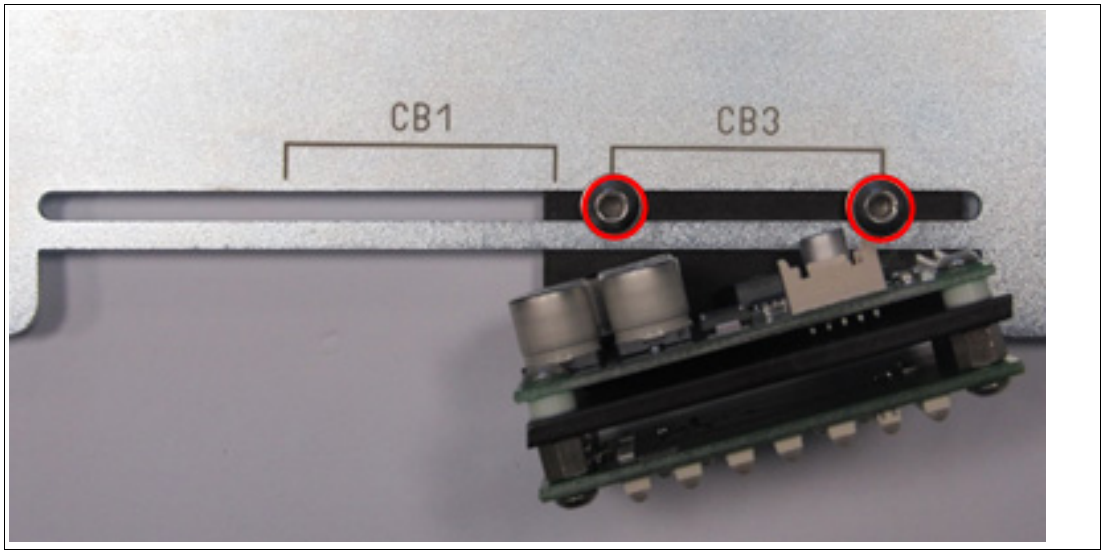


Figure 3.8 Positioning on the rail (CB3)

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).

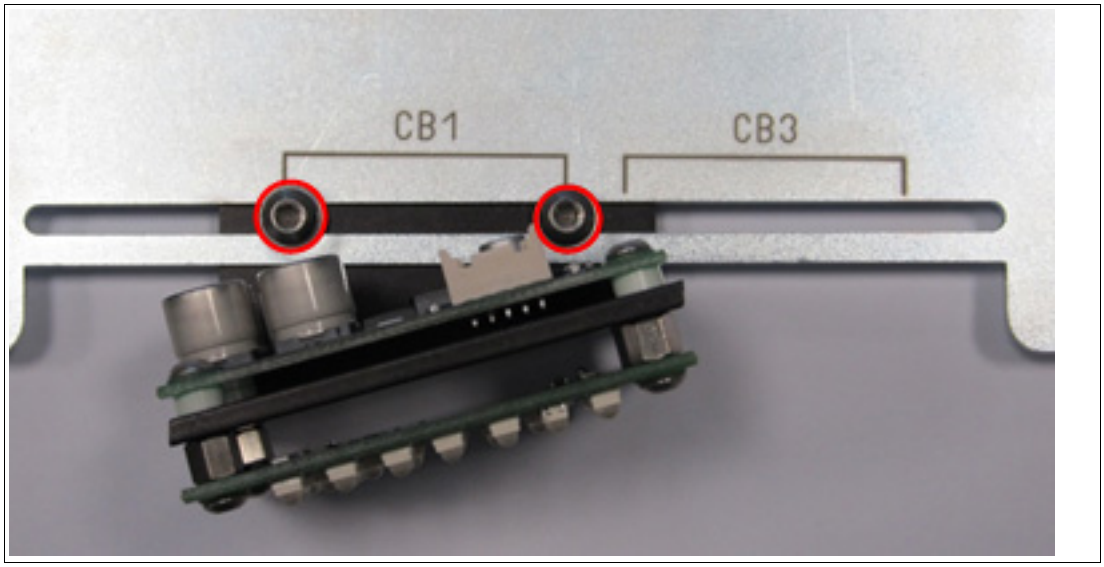


Figure 3.9 Positioning on the rail (CB1)

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:

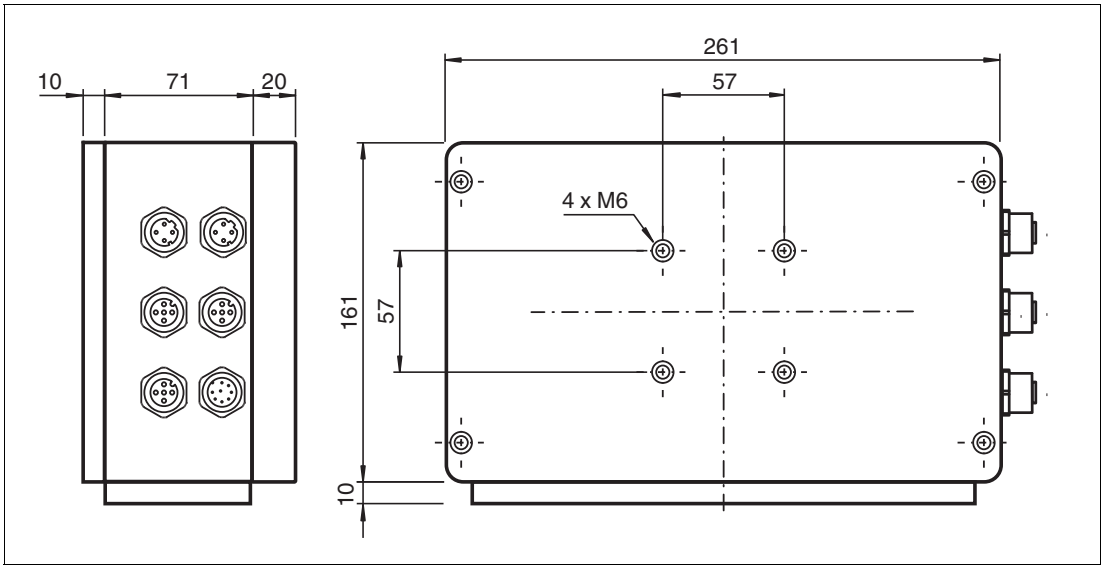


Figure 3.10 Dimensions of the OIT500 system

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.

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

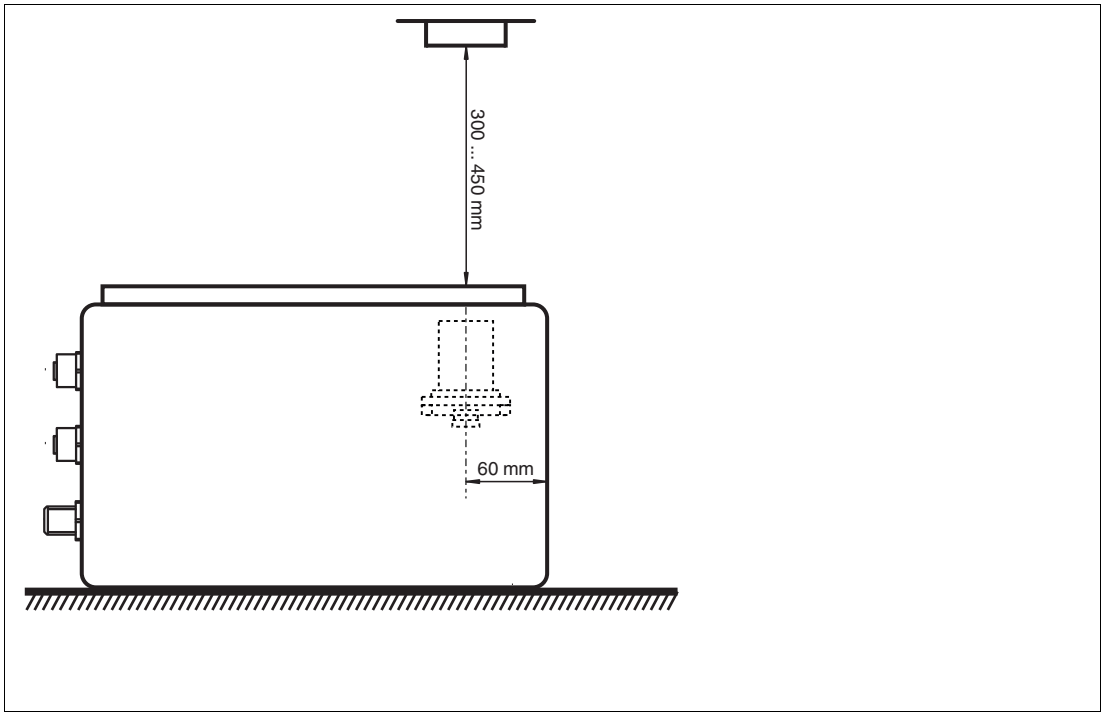


Figure 3.11 Aligning/adjusting code sheets

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.

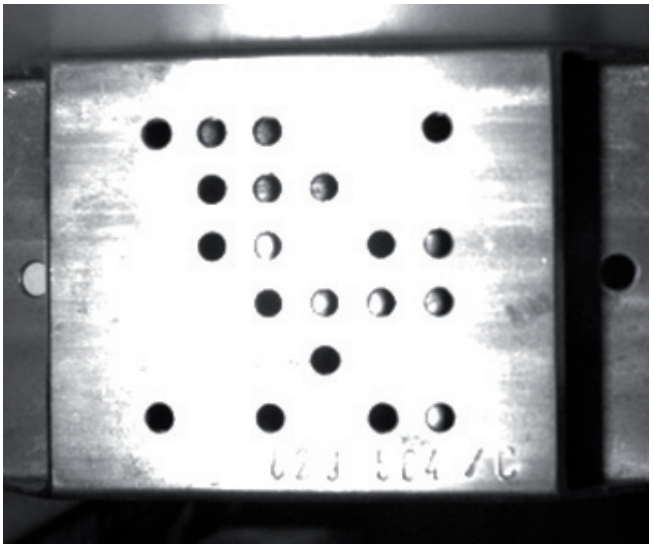


Figure 3.12 Overexposed code sheet

2023-10

3.3 Establishing an Electrical Connection

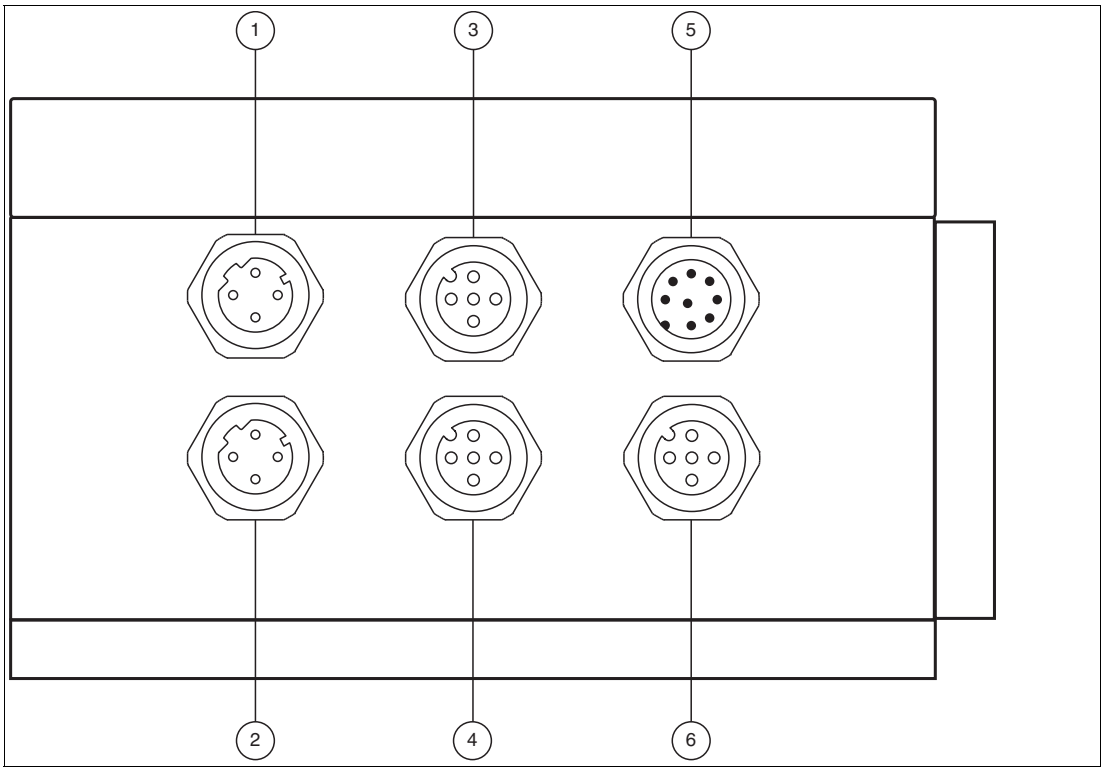


Figure 3.13 Connections

1. PROFINET/EtherNet TCP/IP connection 1
2. PROFINET/EtherNet TCP/IP connection 2
3. Trigger connection 1
4. Trigger connection 2
5. Power supply
6. Connection for external lighting

Providing a Power Supply

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

1. Insert the 8-pin M12 connector into the plug provided on the side of the housing.

Connecting a Trigger Sensor

To connect a trigger sensor, proceed as follows:

1. Plug the 5-pin M12 plug into the socket provided for this purpose on the side of the housing.

Connecting External Lighting

To connect an external lighting unit, proceed as follows:

1. Plug the 5-pin M12 plug into the socket provided for this purpose on the side of the housing.

Creating a PROFINET/EtherNet TCP/IP connection

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

1. Plug the 4-pin M12 plug (D-coded) into the socket provided for this purpose on the side of the housing.
2. Check the network settings for the PC/laptop as described in the next chapter.

Note

The device has an integrated switch. Additional Ethernet devices can be connected to the network via the second PROFINET connection.

When operating with PROFINET, the device supports PROFINET IO communication in Conformance Class B with 100 Mbit/s.

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.

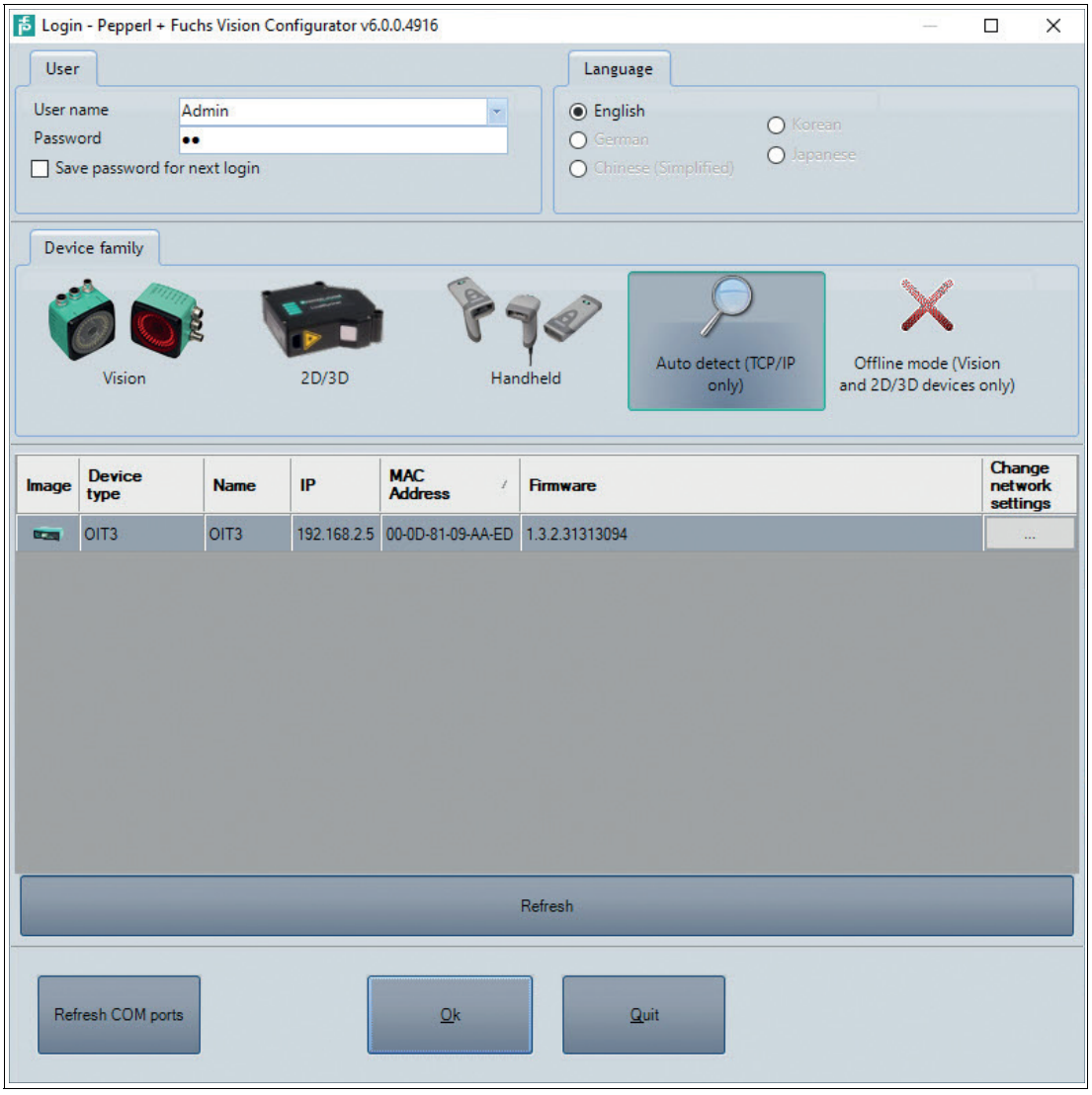


Figure 3.14 Auto detect function in Vision Configurator

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:

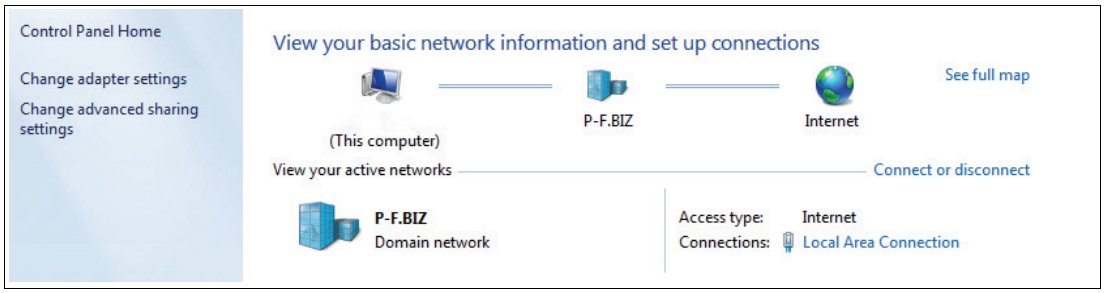
- An "Image" of the device
- The "Device type"
- The "Name" of the device (NOTE: Under Name the PROFINET name of the device is displayed).
- The "IP" of the device

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

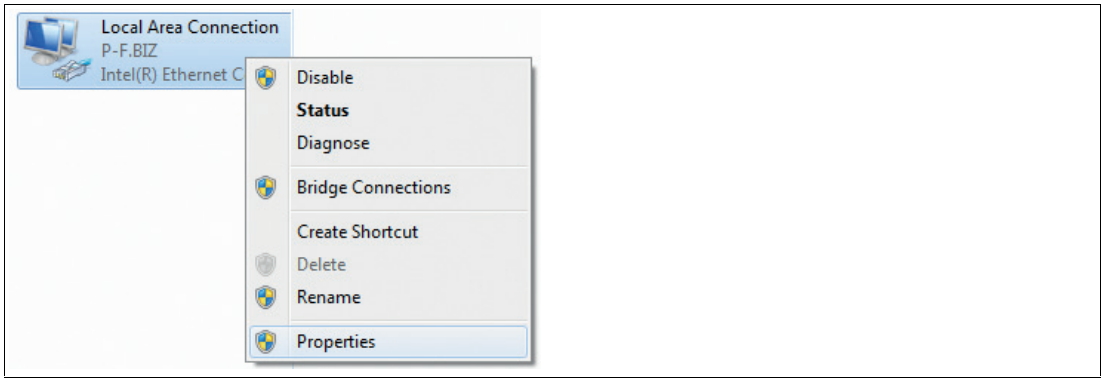
Setting the IP Address of the PC

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 and Sharing Center**."
3. Now click "**Change adapter settings**."

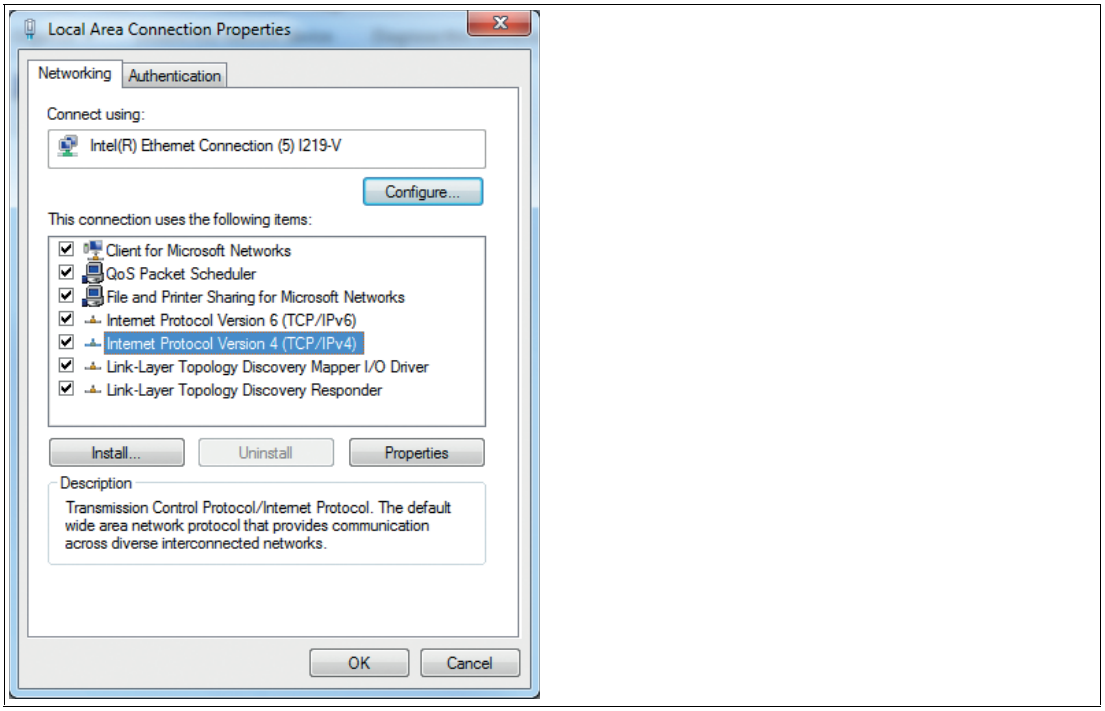


4. Select the required connection and right-click on your selection. In the selection window, select **Properties**.

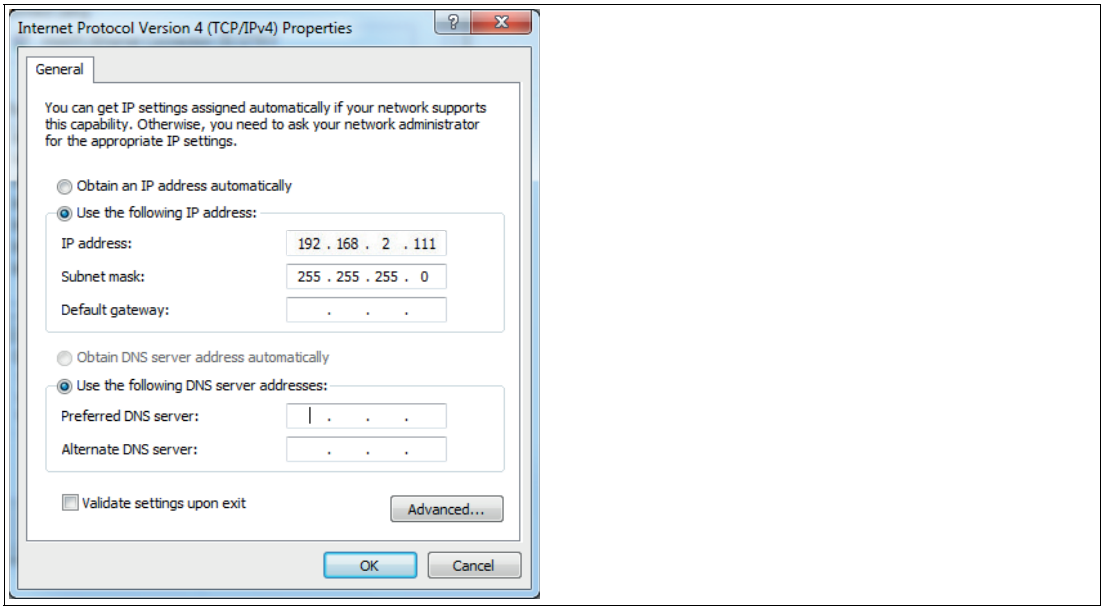


5. Double-click "**Internet Protocol Version 4 (TCP/IPv4)**."

2023-10



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



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: 192.168.2.111**
   - **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, see chapter 5.
3. Double-click on the device type **OIT** (1).

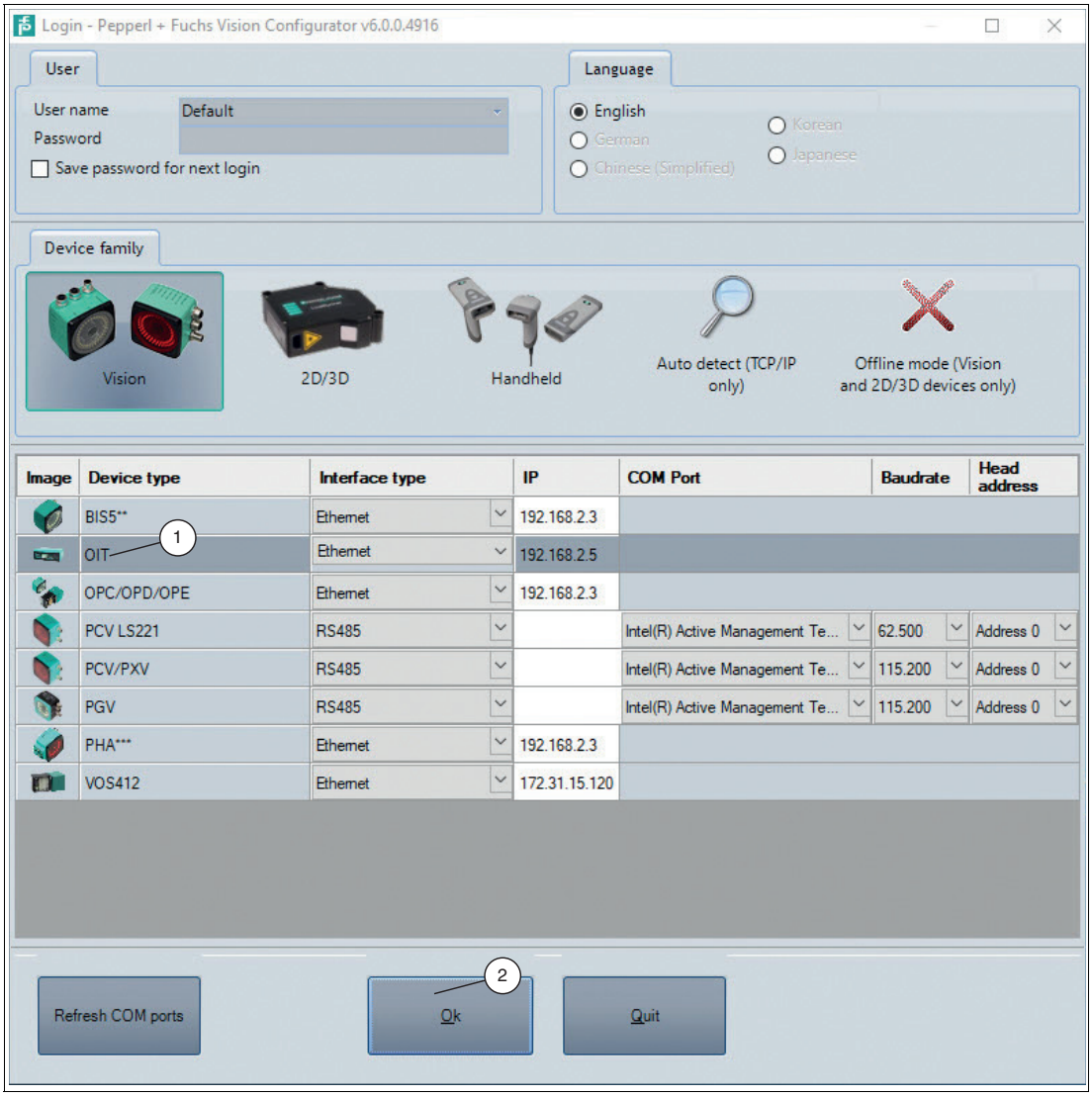


Figure 3.15 Selecting the device

4. Click on **Ok** (2) to confirm your selection.

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

4 Commissioning

4.1 Integrating the OIT System into the Network

Warning!

Risk of injury due to incorrect configuration

An error during the configuration of the device can override the fail-safe function, causing a danger to people and machinery.

- Ensure that the device is programmed exclusively by qualified personnel.
- Only put devices into operation after they have been configured correctly.

PROFINET Device Name

For the OIT system to be addressed as a participant on the PROFINET device, this OIT system requires a unique PROFINET device name.

PROFINET Address

In normal operation (data exchange mode), the IP address is usually assigned to the OIT system by the PROFINET controller (PLC). For this purpose, the OIT system has a device name that is used to address it.

Note

Various configuration tools are available for configuring your OIT system. As an example, this manual describes the configuration of a Siemens control panel with the OIT system using the TIA Portal V13. If you are using a programmable logic controller (PLC) from a different manufacturer, the process will be similar to the one described here.

Install the GSDML file

You require a **GSDML file** for the operation of the device. The GSDML file can be downloaded from our website: www.pepperl-fuchs.com. Simply enter the product name or item number in the Product/Keyword field and click "Search." Select your product from the list of search results. Click on the information you require in the product information list, for example, Software. A list of all available downloads is displayed.

1. Launch the "TIA Portal."

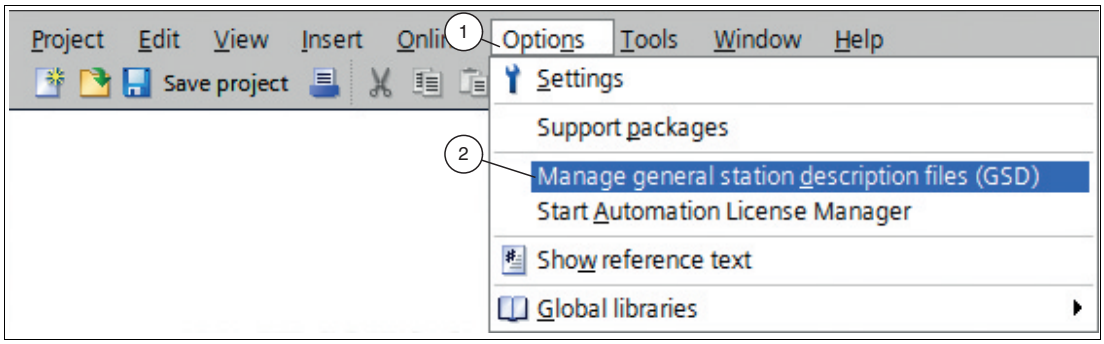


Figure 4.1 GSDML file

2. Under *"Options"* (1) in the menu bar, select the command *"Manage general station description files (GSD)"* (2).

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

2023-10

PEPPERL+FUCHS

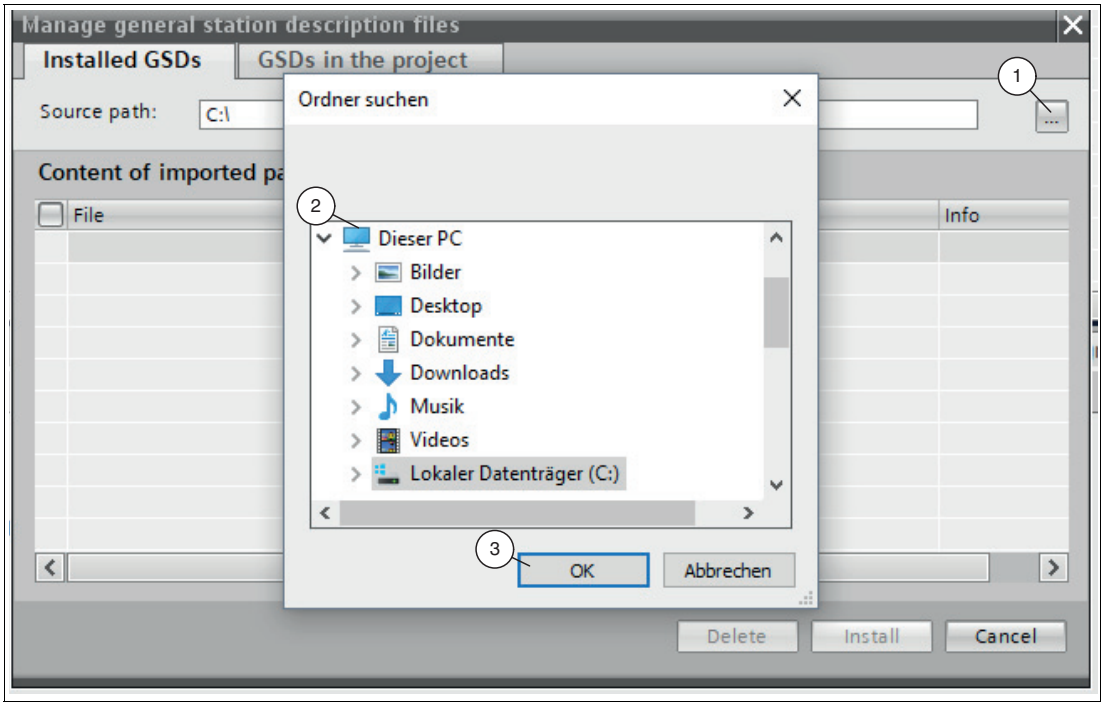


Figure 4.2 Search for GSDML file

3. Click the *"button with the three dots"* (1), which allows you to search for your GSDML file on the storage medium.
4. Select your GSDML file (2) and click *"OK"* (3) to confirm your selection.

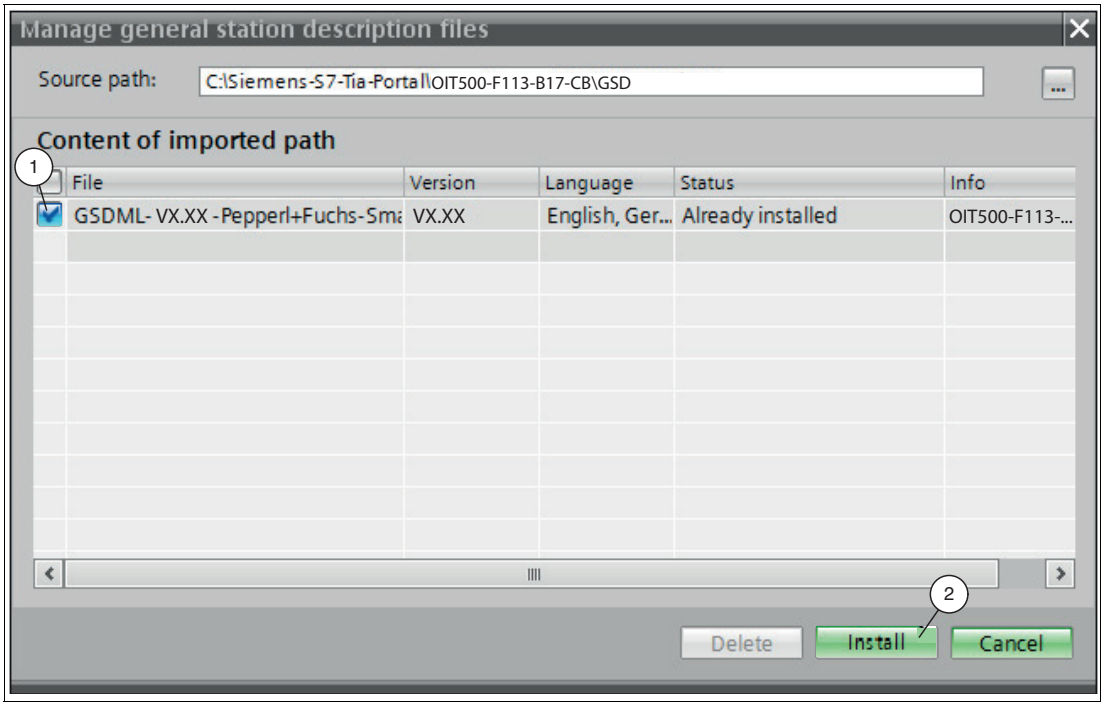


Figure 4.3 Install the GSDML file

5. Select the GSDML file to install by checking the box (1) to the left of the filename.
6. Click the *"Install"* button (2). The installation process may take a few minutes.

↳ 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. The project view opens in the work area without a selection being made.

Integrating a Device into the Project

1. Open the hardware catalog and browse through the tree structure until you see your device (1).

Note

The figures are provided to aid basic understanding and may deviate from the actual design.

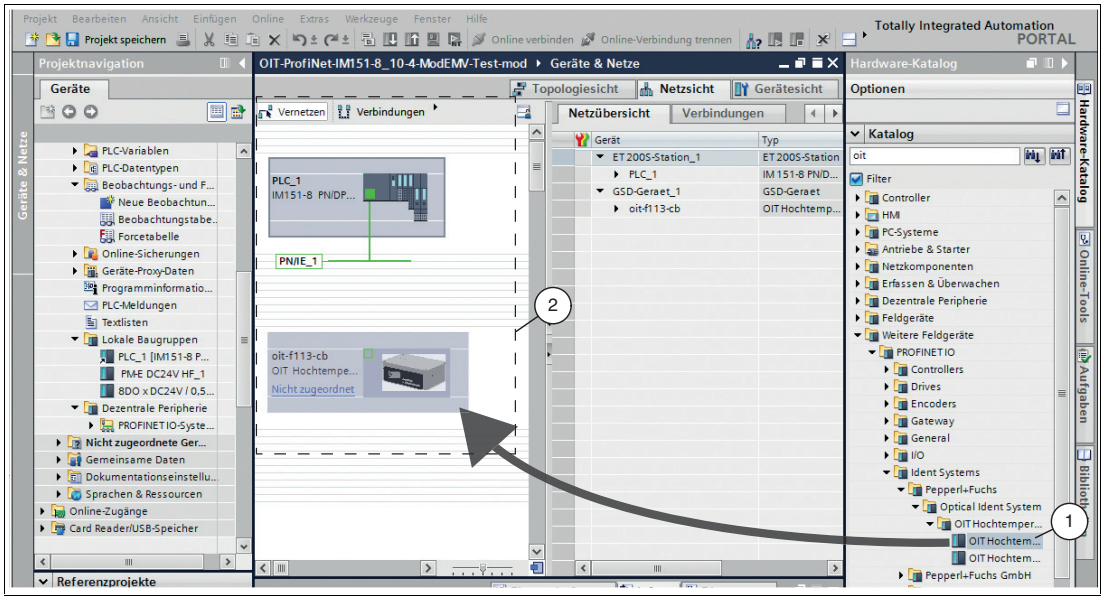


Figure 4.4 Integrating the OIT system into the project

2. Select your device (1) from the hardware catalog and drag this module into the network view (2).

↳ The device is displayed in the network view window (2).

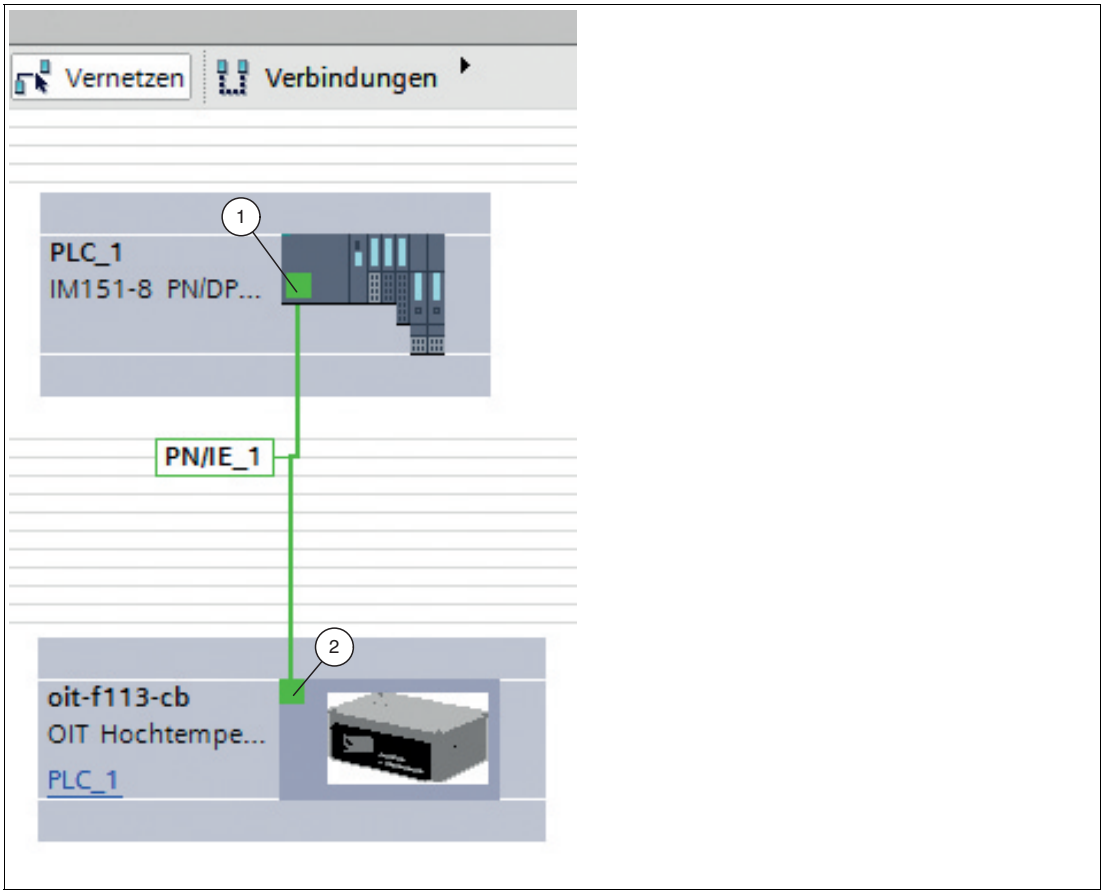


Figure 4.5 Connecting the device with the control panel

3. To connect the device with the control panel, move the mouse to the PROFINET interface that is highlighted in green in the control panel (1). Click the left mouse button and drag the line shown to the PROFINET interface on the device (2). Once there, release the left mouse button again.

↳ The device is now connected to the control panel. A PROFINET subsystem is created.

Integrating Modules into the Project

Multiple modules are available in the hardware catalog.

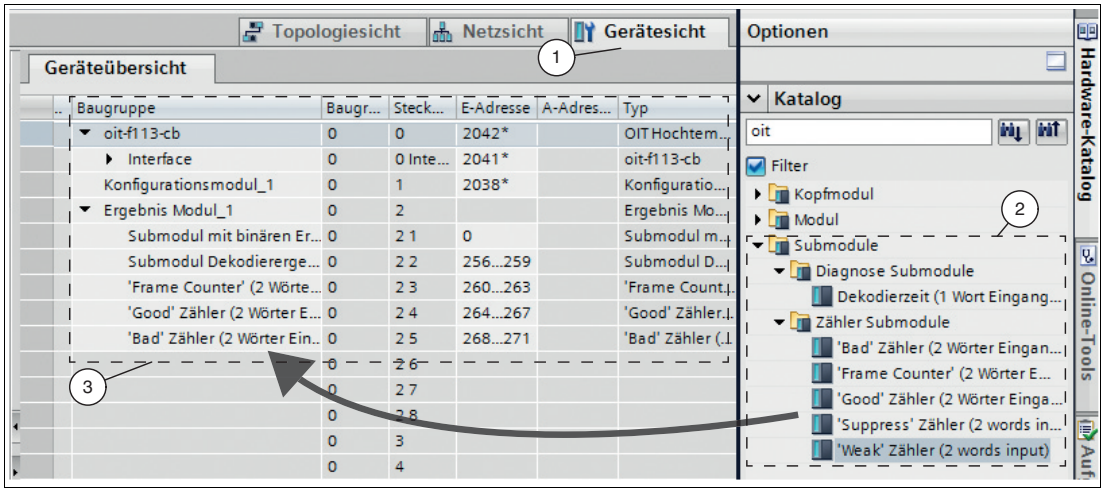


Figure 4.6 Adding a module

1. Switch to the *"Device view"* tab (1) in the *"Device and networks"* window.
2. Open the hardware catalog. Select your submodule (2) from the hardware catalog and drag it into the device view (3).

   ↳ The selected modules are automatically assigned an I and O address range.

Compiling a Project Configuration

The project configuration must be compiled before it is transferred to the control panel. During compilation, the project configuration is converted such that it can be read by the control panel.

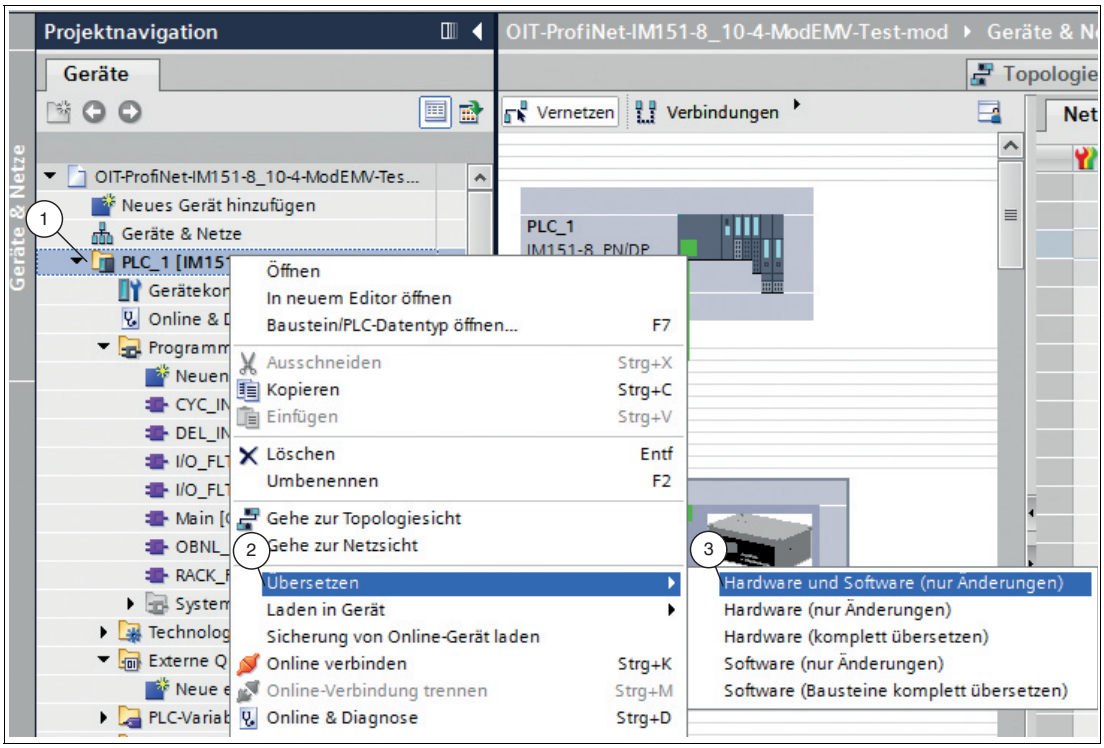


Figure 4.7 Compiling a project configuration

1. In the project tree, right-click on the control panel and select "PLC_1*" (1) and select "Compile" (2) > "Hardware and software (changes only)" (3) in the context menu.

   ↳ The compilation process starts and can be monitored in the inspection window via tabs "Info > Compile." The compilation must be completed without errors to be able to transfer a functional program to the control panel.

Loading the project configuration into the control panel

To load the project configuration into the control panel, proceed as follows:

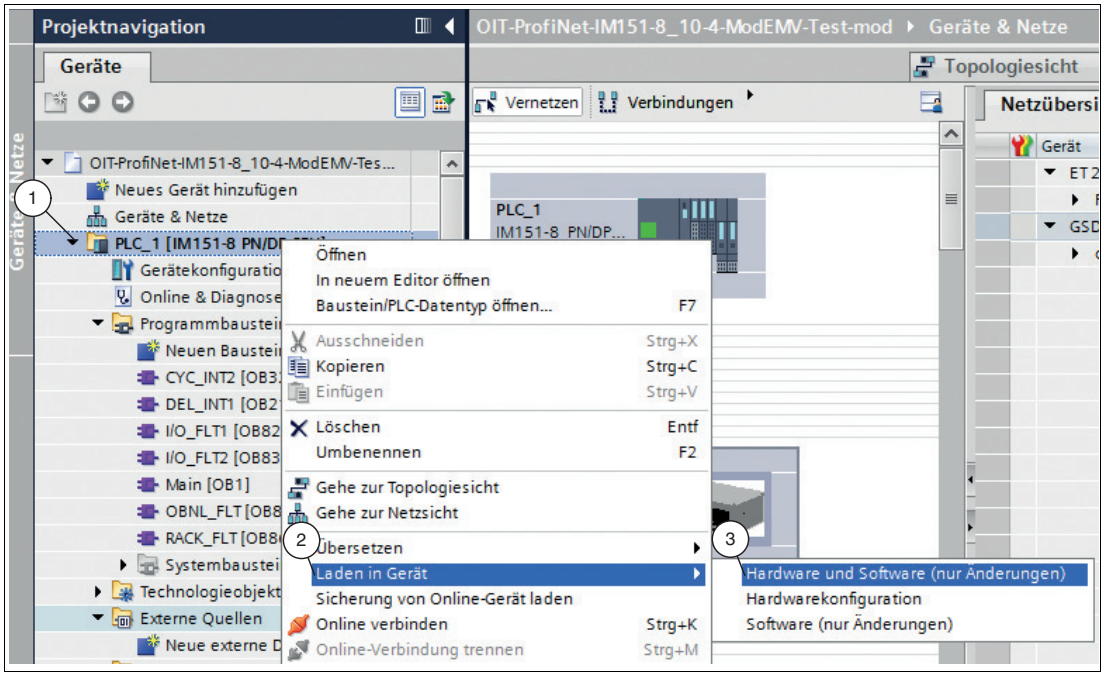


Figure 4.8 Loading project configuration

1. In the project tree, right-click on the control panel and select "PLC_1*" (1) and select "Load into device" (2) > "Hardware and software (changes only)" (3) in the context menu.

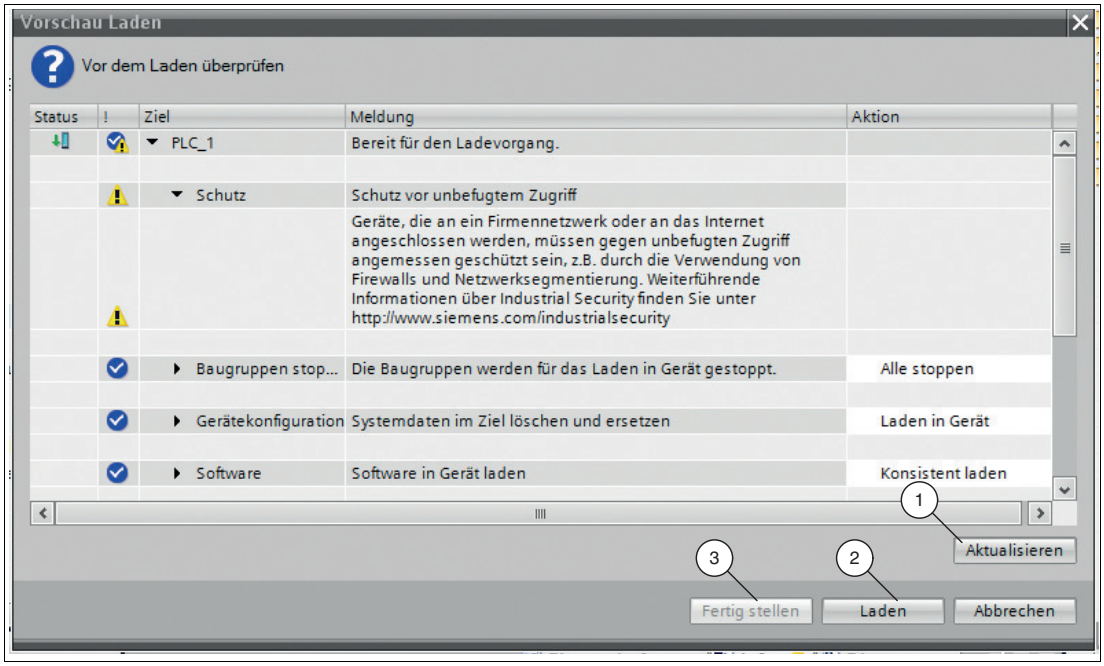


Figure 4.9 Loading the preview

↳ The "Load preview" dialog box opens.

2. Check the messages in the dialog box. If all the conditions are met, proceed to the next step. If not, correct the errors and repeat the loading procedure by clicking the "Refresh" button (1).

2023-10

3. Click the "Load" button (2).

   ↳ The project configuration is loaded into the control panel.

4. When the transfer is complete, click the "Finish" button (3).

Creating Tag Tables

A tag table must be created to be able to watch or control measured values. In the tag table, you can assign tags to input addresses or output addresses and allocate tag names. The names are valid throughout the control panel. To create a tag table, proceed as follows:

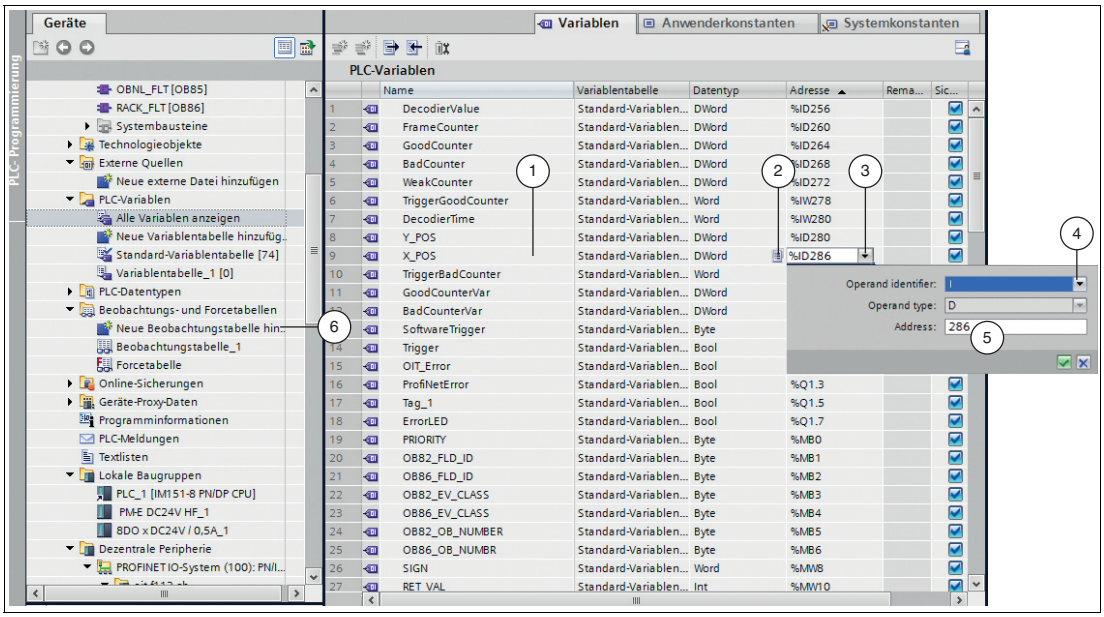


Figure 4.10 Creating a tag table

1. In the project tree, click on **PLC tags** and double-click on **Add new tag table** (6).
2. Enter a tag name in the field in the **Name** column (1).
3. Select the appropriate data type from the drop-down list in the **Data type** column (2).
4. Click on the drop-down list (3) in the field in the **Address** column.

   ↳ A dialog box opens in which you can define the properties and the address range for the tag.

5. From the drop-down list in the **Operand identifier** field (4), select whether it is an input channel, an output channel, or a marker:
   - I: input channel
   - O: output channel
   - M: marker
6. From the drop-down list in the **Operand type** field, select the data type for address assignment:
   - X: 1 bit
   - B: 1 byte (8 bits)
   - W: 1 word (16 bits)
   - D: 1 double word (32 bits)
7. Select an address range for the tag by entering a value in the **Address** field (5).

Note

Address Range

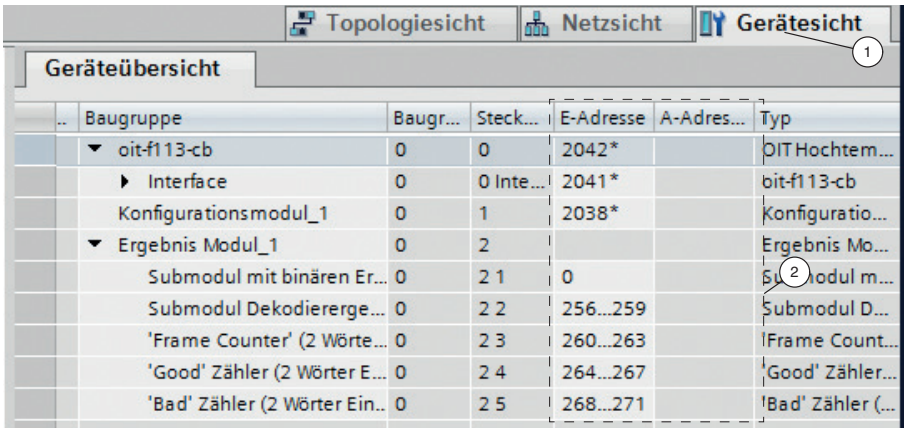


Figure 4.11 Address range

The address designations in this example are project-dependent and may differ from your actual address assignment. The memory range is automatically assigned by the TIA Portal when the modules are integrated into the project. This memory range determines the address range in the tag table.

Switch to the device configuration then to the OIT system (**Device view** (1)) to copy the address range of the OIT system (2) to the address range (**Address**) field of the tag table.

2023-10

Creating an Observation Table and Monitoring Tags

Note

To use observation tables, activate online mode.

To create an observation table and monitor measured values, proceed as follows:

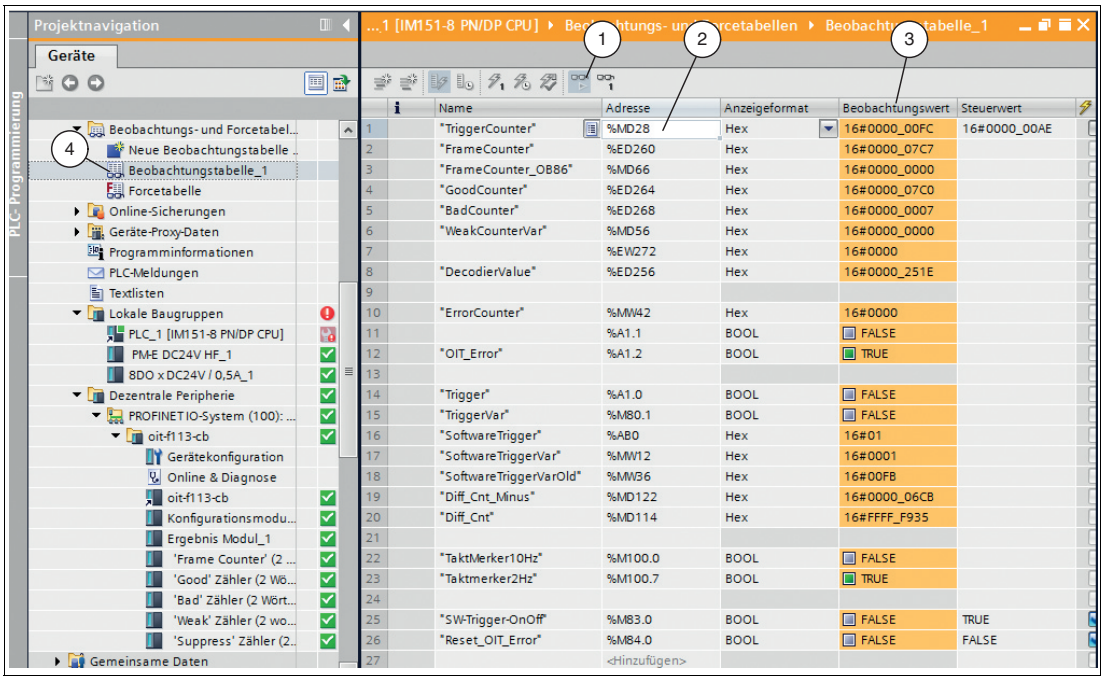


Figure 4.12 Creating an observation table

1. In the project tree, click on **Watch and force tables** and double-click on **Add new watch table** (4).
2. Click the field in the **Address** column (2).

   ↳ A list of addresses that you have defined in the tag table is displayed.
3. From the list, select the address that you want to watch.
4. Click on the watch icon (1) to watch the values.

   ↳ The current measured value is displayed in the field in the **Monitor value** column (3).

4.2 Parameterization

The factory default parameters are entered in the OIT memory. You can adjust the parameters using the "TIA Portal" configuration software. The OIT system can also be parameterized via the "Vision Configurator," see chapter 5.

Opening the Parameter Table

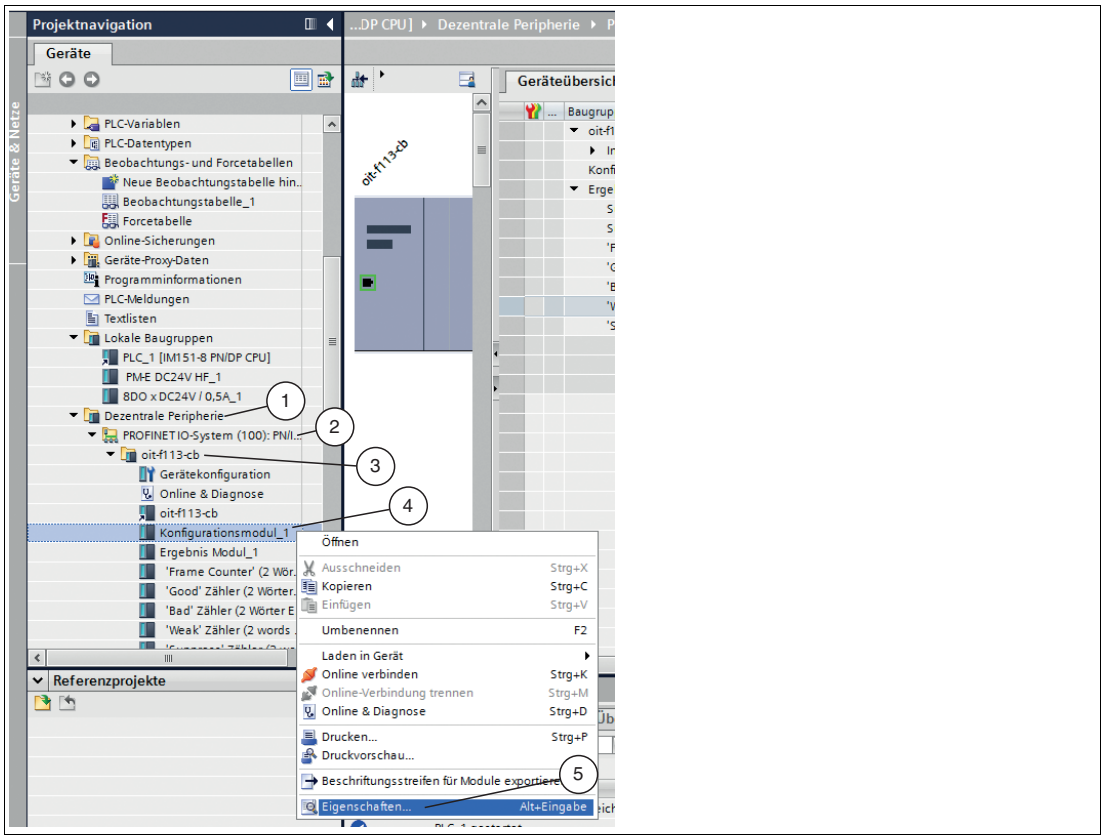


Figure 4.13 Calling up the parameter window

1. In project navigation, right-click on the control panel and select "distributed peripherals" (1) > "PROFINET IO-System (100)" (2) > "oit-f113-cb" (3) > "Configuration module_1" (4).

   ↳ The context menu opens.

2. Select "Properties" (5) from the context menu.

   ↳ The "Configuration module_1" window opens.

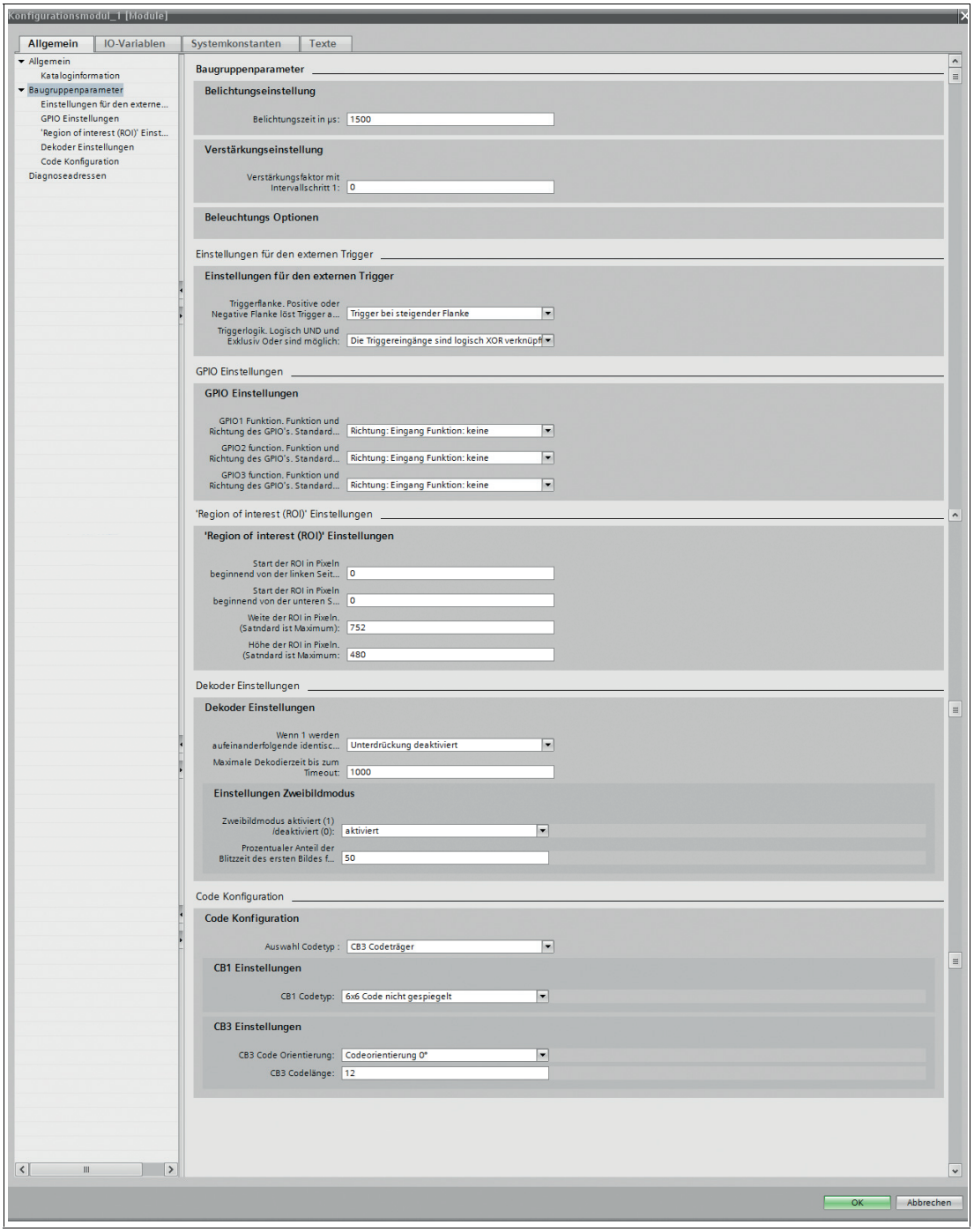


Figure 4.14 Configuration module_1

3. The "module parameters" can be accessed from the "General" tab. You can parameterize your data here. The table below shows you which parameters you can adjust.

Global Primary Data

The global primary data allows you to parameterize the OIT system using PROFINET. Global primary data is always transferred to the OIT system in full.

Configuration Parameters

Module parameters	Parameter ID	Data type	Value	Description
Exposure time	ExposureTime	32 bit	0 ... 20,000 (default: 1500)	Exposure time in µs
Gain setting	Gain	8 bit	0 ... 255 (default: 0)	Image gain
Set external triggers	TRIGGER_EDGE	bool	0 = rising edge (default: 0) 1 = falling edge	Sets the trigger activation according to the rising or falling edge
	TRIGGER_LOGIC	8 bit	0 ... 1 (default: 0) 0 = OR link 1 = AND link	Determines the function of the two triggers.
ROI Settings	XStart	16 bit	0 ... 752 (default: 0)	X coordinate of the decoder frame in which the OIT attempts to read a perforated matrix
	YStart	16 bit	0 ... 480 (default: 0)	Y coordinate of the decoder frame in which the OIT attempts to read a perforated matrix
	RoiWidth	16 bit	0 ... 752 (default: 752)	Width of the decoder frame (ROI)
	RoiHeight	16 bit	0 ... 480 (default: 480)	Height of the decoder frame (ROI)
Decode settings	SuppressDuplicates	bool	0 ... 1 (default: 0) 0 = deactivated 1 = activated	Activate/deactivate suppression of the decoding of consecutive, identical codes
	TimeoutDecode	16 bit	0 ... 65535 (default: 1000)	Maximum decoding time in ms
Set two-image mode	CaptureTwoImages	bool	0 ... 1 (default: 1) 0 = deactivated 1 = activated	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
	ReducedFlashtime-Percent	8 bit	0 ... 100 (default: 50)	The percentage of the exposure time of the second image compared to the first image
Code configuration	DecoderPlate	8 bit	0 ... 1 (default: 0) 0 = CB3 code sheets 1 = CB1 code sheets	Value = 0: activates the reading of CB3 code sheets Value = 1: activates the reading of CB1 code sheets

2023-10

Module parameters	Parameter ID	Data type	Value	Description
CB1 settings	CodetypeCB1	8 bit	0 ... 1 (default: 0) 0 = 6x6 perforated matrix 1 = 6x6 perforated matrix, mirrored	Determines the sheet type for CB1
CB3 settings	CB3 orientation	8 bit	0 ... 3 (default: 0) 0 = 0: normal 1 = 0: mirrored 2 = 180: rotated by 180° 3 = 180: rotated by 180° and mirrored	Determines the alignment of the code sheet in the sensor image
	CodetypeCB3	8 bit	8 ... 12 (default: 12)	Setting for data bits in the CB3 code sheet

5 Parameterization Using Vision Configurator

The OIT system can be parameterized 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.

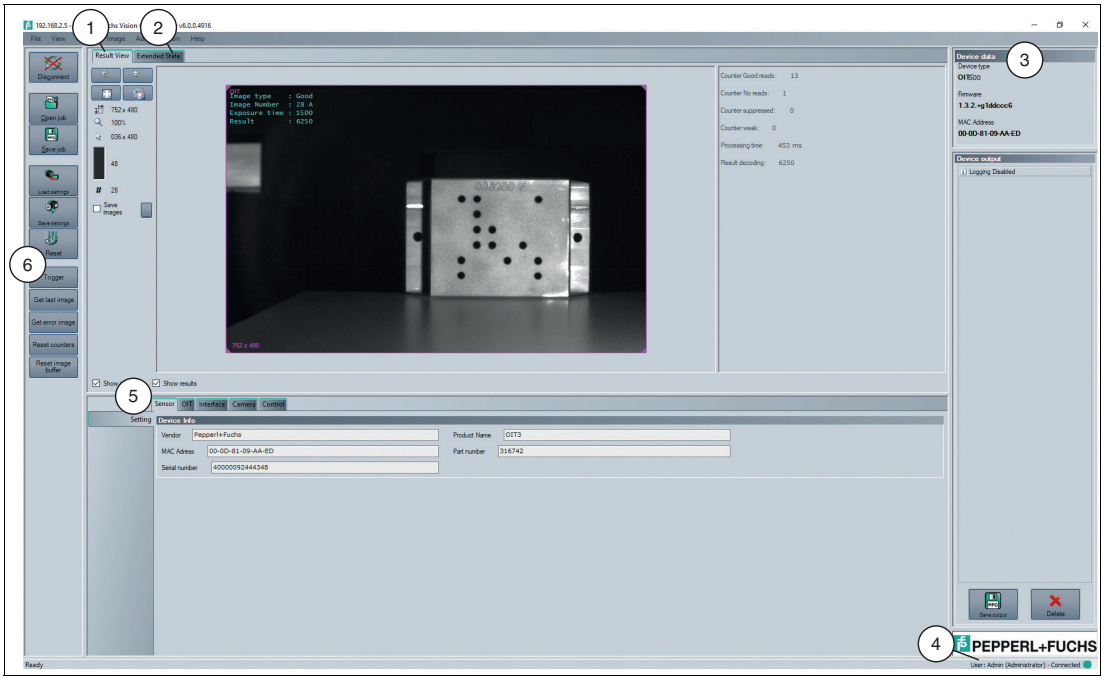


Figure 5.1 Application screen

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 status bar shows information about the user who is logged in and the sensor connection status.
5. The parameterization area is split into several subareas and contains sensor-specific parameters.
6. 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

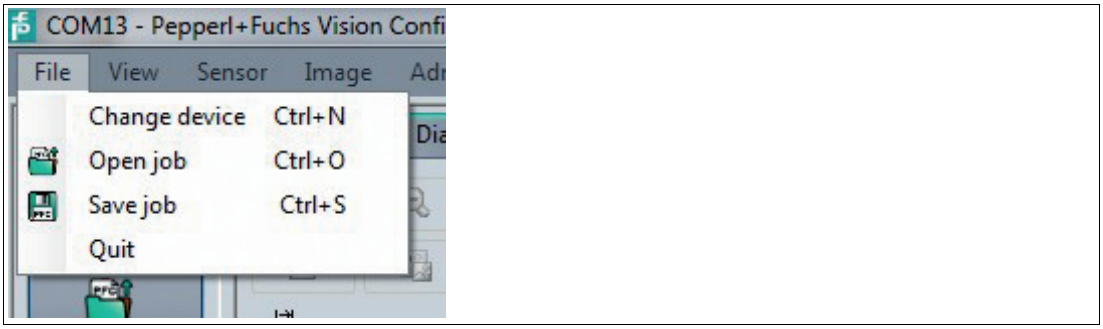


Figure 5.3 **File** Menu

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

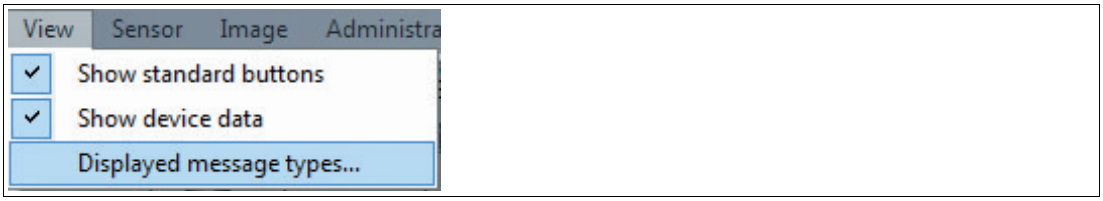


Figure 5.4 **View** Menu

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

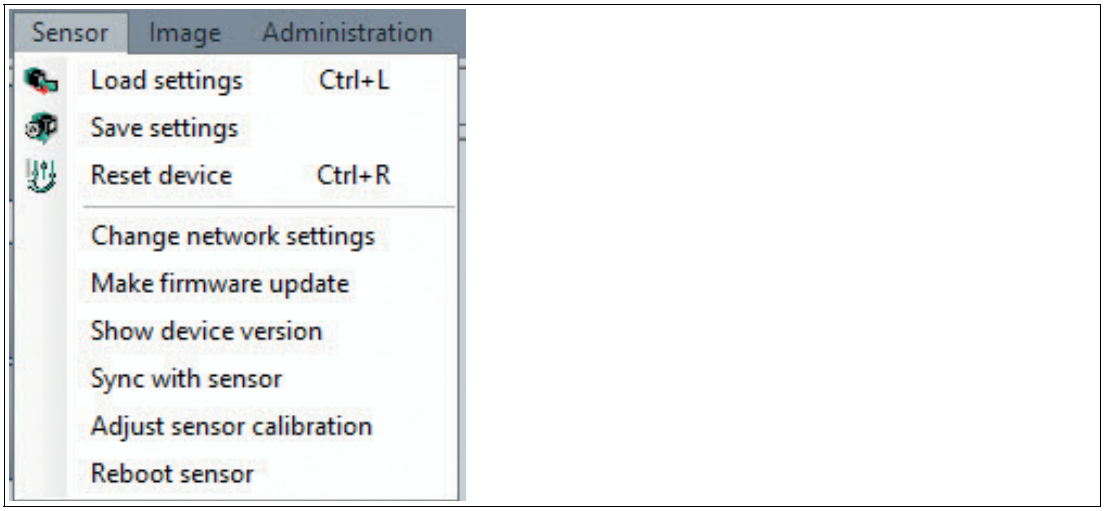


Figure 5.5 **Sensor** menu

Load settings	Loads the saved settings from the sensor
Save settings	Saves the settings to the sensor
Reset device	Resets the sensor to its default settings
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	Currently not supported
Reboot sensor	Restarts the sensor

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

Figure 5.6 **Image** menu

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
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

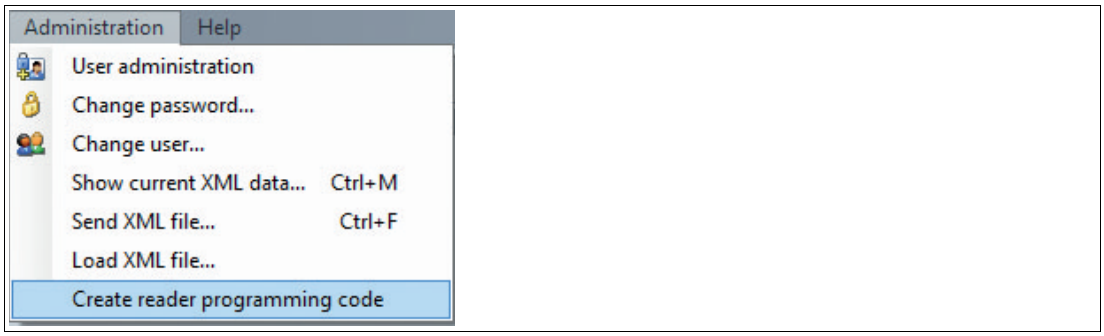


Figure 5.7 **Administration** menu

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	Not supported

Table 5.5 **Administration** menu

5.2.6 *Help* Menu

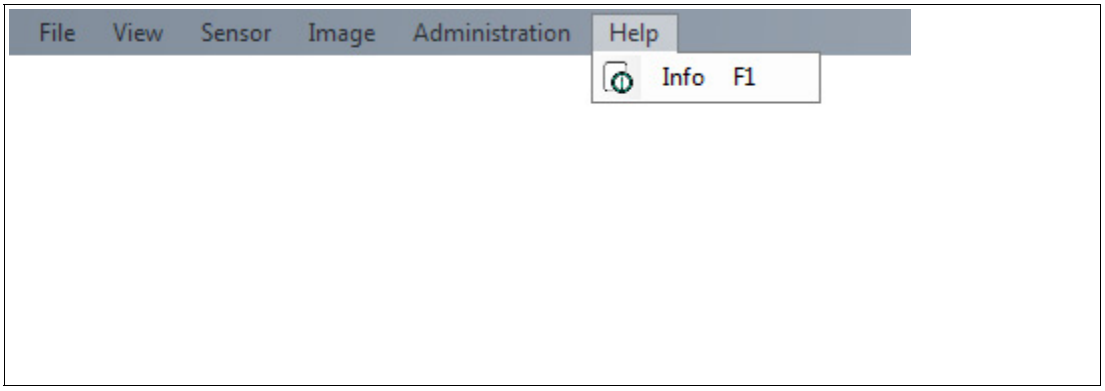


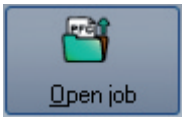
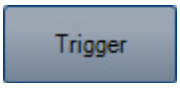
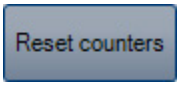
Figure 5.8 **Help** menu

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.
	Resets to factory default settings.
	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 counters to 0, such as the counters for successful and failed readings.
	Reset image memory.

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.

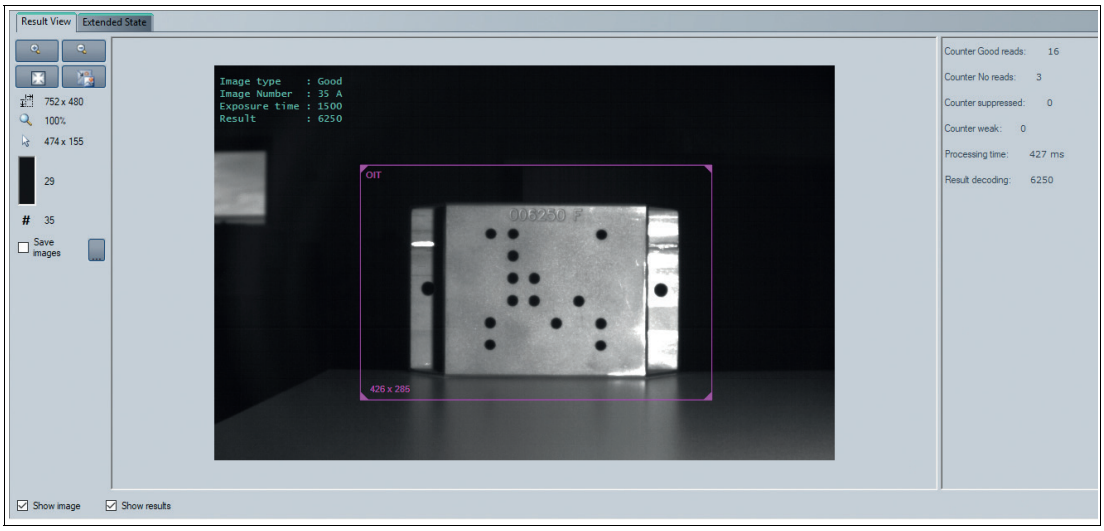


Figure 5.9 Image display (left) and result display (right)

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

Toolbar

The toolbar is located on the left side under the **Result View** tab. There are some useful functions in the toolbar which are used to further process recorded images. The following functions are available to you.

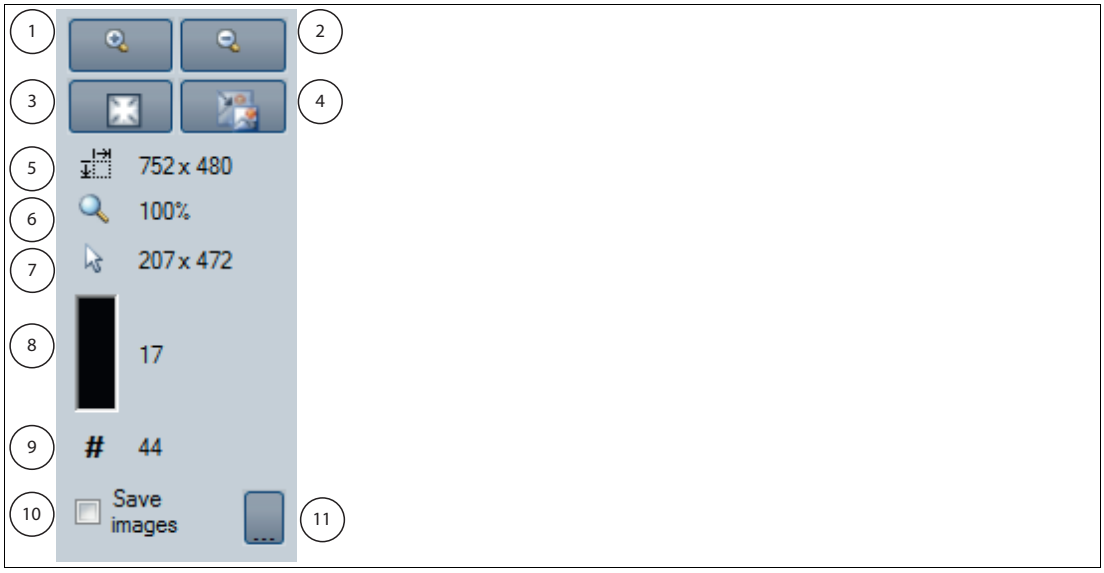


Figure 5.11 Toolbar

Position	Description	Function
1	Magnifier +	Maximize image
2	Magnifier -	Minimize image
3	Fit to window	Fit image size to the window
4	Original size	Set size of original image
5	Size details	Image size information field
6	Zoom factor	Zoom factor information field. Zoom factor of 100 % is the original image size
7	Position details	Shows the position of the mouse cursor
8	Gray scale value details	Gray scale value details for the pixel indicated by the mouse cursor
9	Image counter	Displays the current image number
10	Save image	Saves every image following transfer
11	Select path	Select path on the storage medium

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).

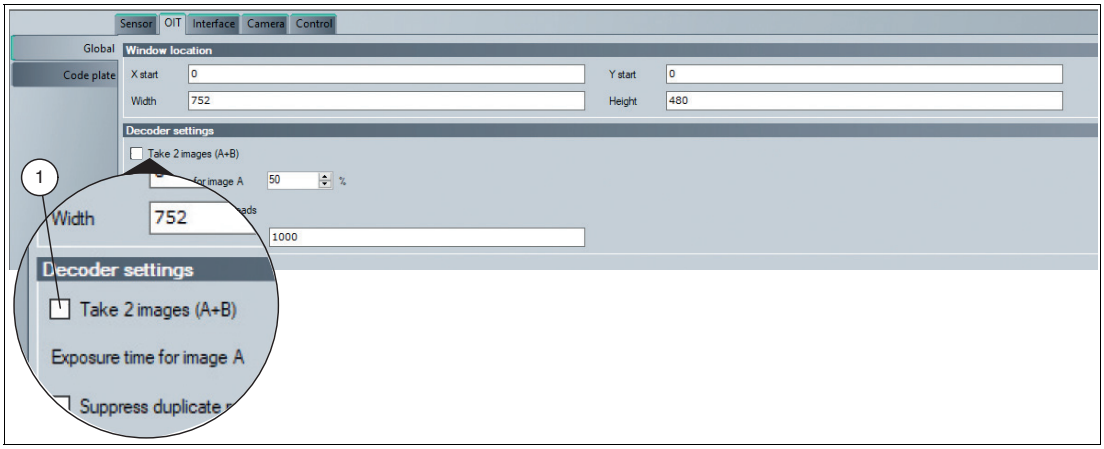


Figure 5.12 Single image mode

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).

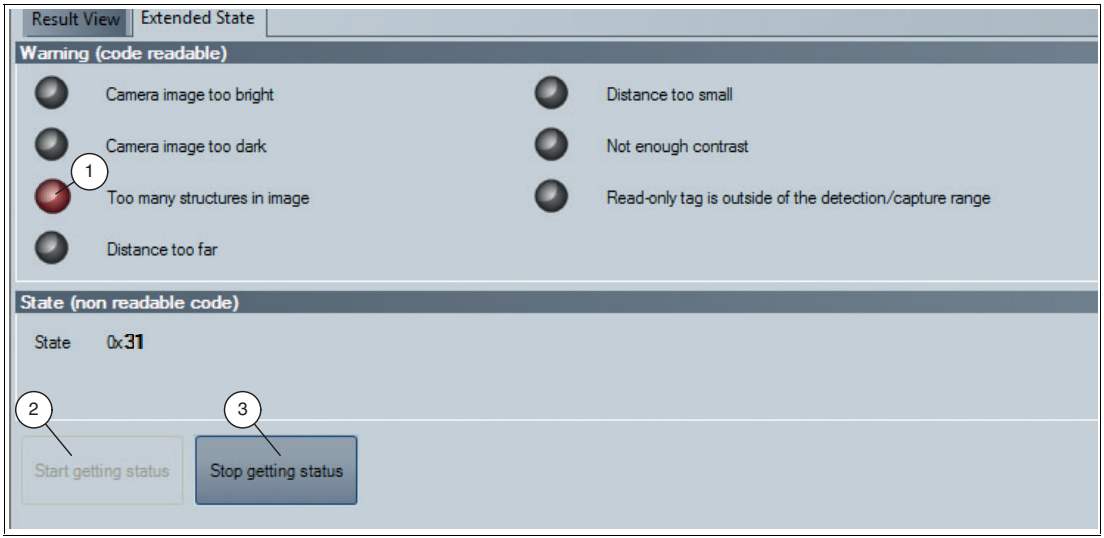


Figure 5.13 Extended State

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 5.6.4.
Camera image too dark	The camera image is too dark.	Set the right exposure time, see chapter 5.6.4.
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 5.6.4.
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 5.6.4.

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

2023-10

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 Sensor Tab

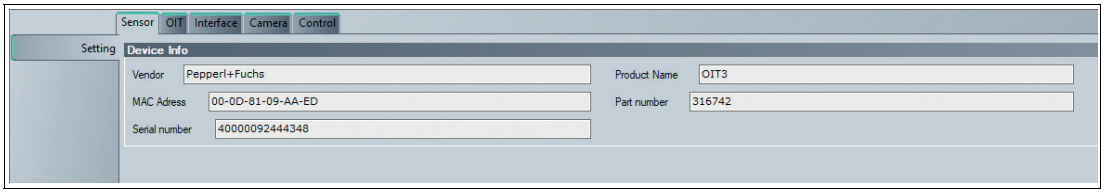


Figure 5.14 **System** tab **Setting** menu item

Device Info

Vendor	Manufacturer
MAC ID	MAC ID (Media Access Control Address)
Serial number	Serial number
Product Name	Product name
Part number	Model number

5.6.2 OIT Tab

Global Menu

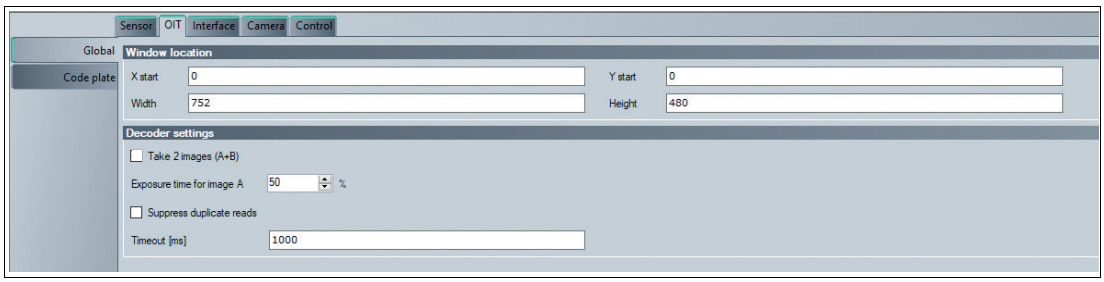


Figure 5.15 **OIT** tab **Global** menu item

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

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.
Exposure time for image A	Exposure time for image A, input in % The percentage refers to the exposure time for a normal capture that has been set in the **Camera** tab under the **Acquisition** 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.

2023-10

Code plate Menu

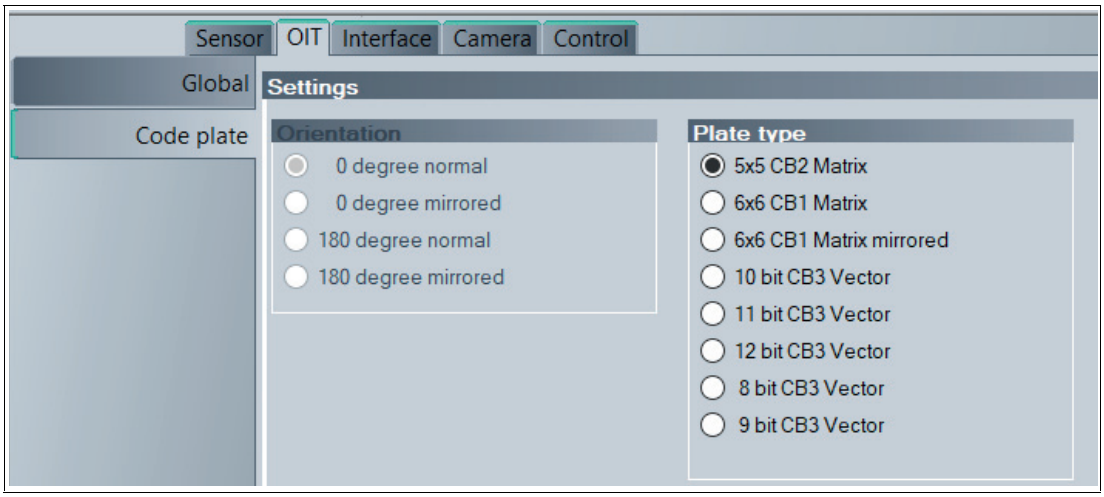


Figure 5.16 **OIT** tab **Code plate** menu item

Settings

Orientation	Setting for the alignment • **0 degree normal**: normal • **0 degree mirrored**: mirrored • **180 degree normal**: rotated by 180° • **180 degree mirrored**: rotated by 180° and mirrored
Plate type	Selection of the code sheet • **5x5 CB2 Matrix**: 5x5 perforated matrix of CB2 code sheet • **6x6 CB1 matrix**: 6x6 perforated matrix of CB1 code sheet • **6x6 matrix CB1 mirrored**: 6x6 perforated matrix (mirrored) of the CB1 code sheet • **10 bit CB3 vector**: 10 bits in the CB3 code sheet • **11 bit CB3 vector**: 11 bits in the CB3 code sheet • **12 bit CB3 vector**: 12 bits in the CB3 code sheet • **8 bit CB3 vector**: 8 bits in the CB3 code sheet • **9 bit CB3 vector**: 9 bits in the CB3 code sheet

5.6.3 **Interface** Tab

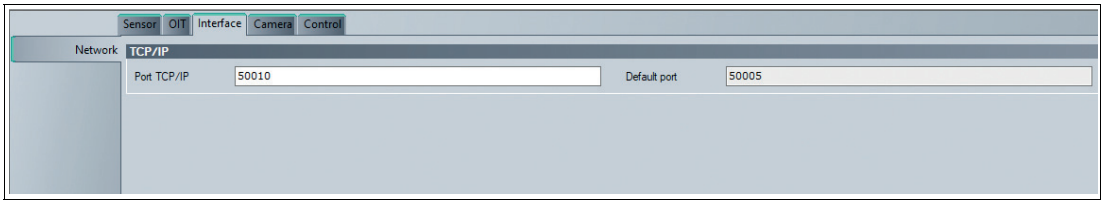


Figure 5.17 **Interface** tab **Network** menu item

TCP/IP

Port TCP/IP	Entry of the port for the TCP/IP connection
Default port	Standard connection

5.6.4 Camera Tab

Image Menu

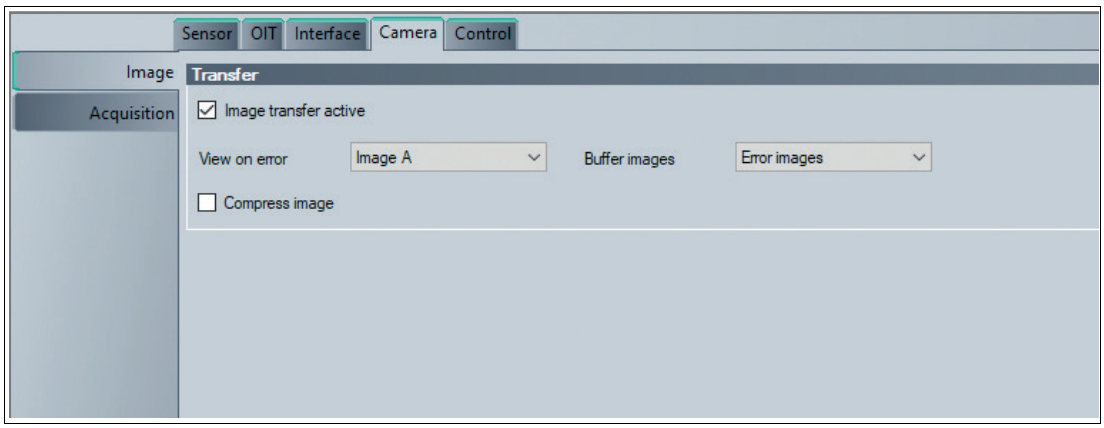


Figure 5.18 **Camera** tab **Transfer** menu item

Transfer

Image transfer active	Activates image transfer to Vision Configurator
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 5.6.2), 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 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
Compress image	Image compression selection field

Acquisition Menu

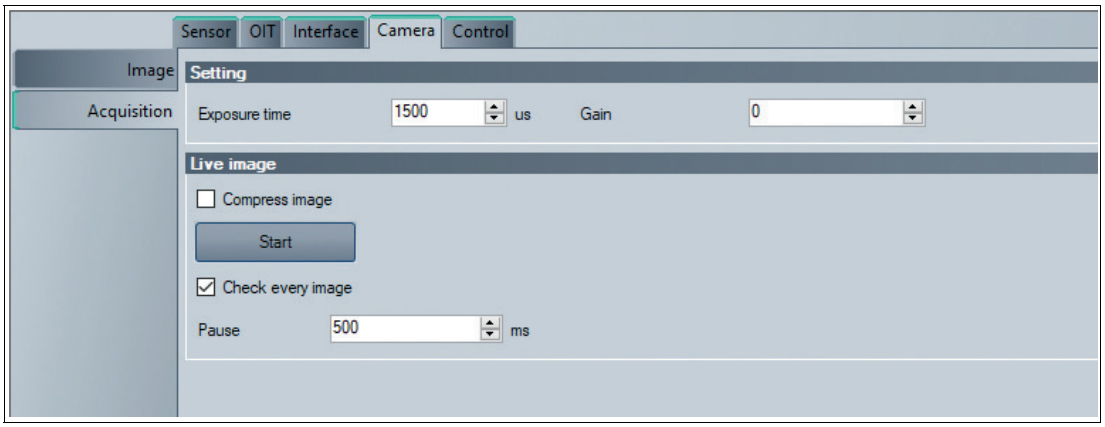


Figure 5.19 **Camera** tab **Setting** or **Live image** menu item

Positions

Exposure time	Exposure time setting in µs Min. = 10 µs, max. = 10,000 µs Default setting: 1500 µs
Gain	Gain setting Min. = 0, max. = 255 Default setting: 0

Tip

Recommended Settings

If possible, set a smaller "gain" (e.g. gain = 1) and set the image brightness via the exposure time. If this setting is not appropriate or the passing speed is too high, increase the "gain."

Live Image

Compress image	Image compression selection field
Start	Starts 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

5.6.5 Control Tab

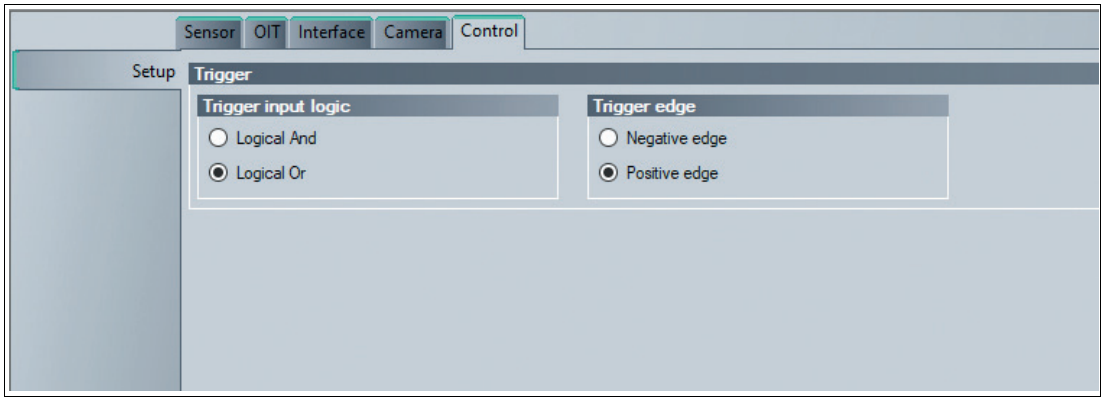


Figure 5.20 **Control** tab, **Setup** menu item

Trigger (inputs 1 and 2)

Trigger input logic	
Logical And	AND link: A trigger must be present so that the 2nd trigger sensor is activated.
Logical Or	OR link: A trigger must not be present so that the 2nd trigger sensor can be activated.
Trigger edge	
Negative edge	The falling edge activates a trigger 
Positive edge	The rising edge activates a trigger 

5.7 Device Data

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

Figure 5.21 Device data

5.8 Device Output

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

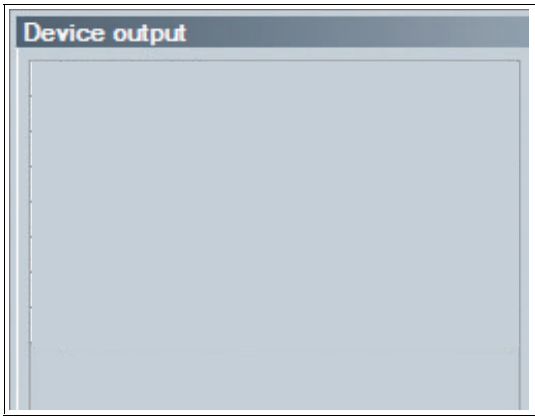
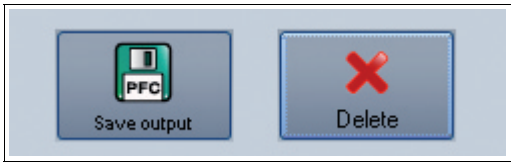


Figure 5.22 Device output

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.**.

6 Operation and communication

6.1 Communication via PROFINET

6.1.1 General Information on Communication via PROFINET

PROFINET is an open standard for industrial automation based on industrial Ethernet. PROFINET integrates information technology with established standards such as TCP/IP and XML in automation technology.

Within PROFINET, PROFINET IO is the communication concept for the construction of decentralized applications. This means that decentralized field devices are integrated through PROFINET IO. The familiar IO view of PROFIBUS DP is used where the usable data of the field devices is transferred to the controller process image in cycles. PROFINET IO is a device model consisting of slots and channels, which is based on the main features of PROFIBUS DP. The field device properties are written in a Generic Station Description Markup Language (GSDML) based on XML. PROFINET IO is engineered in the same way as has long been the case for system integrators of PROFIBUS DP. The decentralized field devices are assigned in the design of a controller.

PROFINET IO distinguishes between the following three device types:

- IO controller: Controller that executes the automation program.
- IO device: Decentrally assigned field device that is assigned to an IO controller.
- IO supervisor: Programming unit/PC with commissioning and diagnostic functions.

6.1.2 PROFINET Modules

This chapter contains the description of the PROFINET process data. All data should be processed in a single module and in a set of optional submodules to deliver consistent data.

Input Data

Response

Description	Data type	Description
Submodule with binary result data	Byte	Submodule contains result status 0 = bad 1 = good
Submodule decoding result	DWord	Submodule contains 32 bit input data with decoding result
"Frame Counter"	DWord	Submodule contains 32 bit input data with frame counter value. The image counter is increased if a new image is triggered.
"Good" counter	DWord	Submodule contains 32 bit input data with the value of the "Good" counter. The "Good" counter is increased if the read-only tag is successfully decoded.
"Bad" counter	DWord	Submodule contains 32 bit input data with the "Bad" counter value. The "Bad" counter value is increased if the decoding was not successful.
"Weak" counter	DWord	Submodule contains 32 bit input data with the "Weak" counter value. The "Weak" counter value is increased with each borderline decoding.

2023-10

Description	Data type	Description
"Suppress" counter	DWord	Submodule contains 32 bit input data with suppressed counter value. The "Suppress" counter is increased with each suppressed decoding of identical codes.
Decoding time	Word	Module contains 16 bit input data with the decoding time of the last successful decoding.

Warning

Bit	Description	Data type	Description
0	Warning information	Word	The camera image is too bright.
1			The camera image is too dark.
2			The distance between the code sheet and the OIT system is too great.
3			The distance between the code sheet and the OIT system is too small.
4			There is insufficient contrast.
5			The code sheet is at the edge of the reading window.
	Warning status	Byte	Value of the warning status. For diagnostic purposes. Value 0x0 signals status "OK."

Output Data

Software Trigger

Bit	Description	Data type	Description
0	Bitmask for trigger output value	Byte	Trigger flag. Changing the value starts a trigger.

7 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 .
- **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 .

7.1 TCP/IP Communication with VSX Protocol

A **.NET 3.5-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.

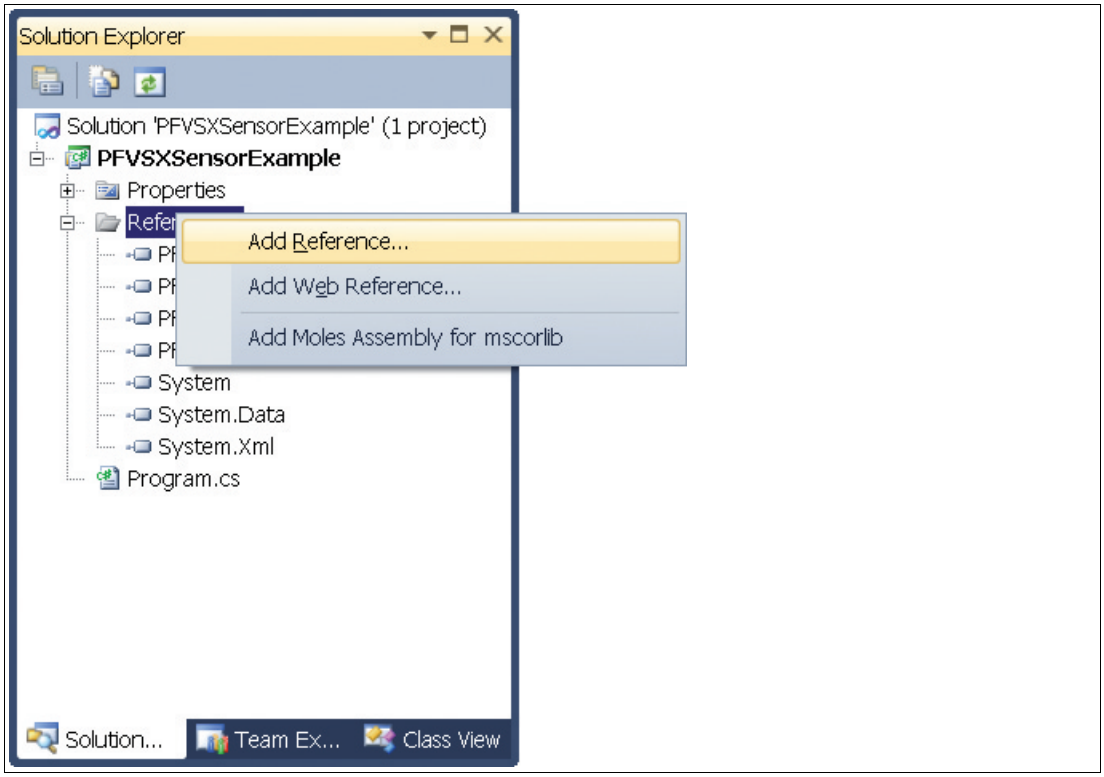


Figure 7.1 Integrating DLLs into the programming environment

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 3.5 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 that has been 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.

2023-10

```
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(uint frameCounter, string confi-
gId, 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(uint 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)`.

```
Bitmap GetImage(uint 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(uint frameCounter, Bitmap image)`.

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

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

```
IList<ElementShapeBase> GetShapes(uint 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)`. 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()
```

2023-10

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` is triggered.

```
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(uint 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 viewed 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 }
```

See event `InternalError`..

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

See event `ParameterDataReceived`.

```
enum LogMessageTypes { DEBUG, INFO, RESULT_OK, RESULT_NOT_OK, WARN-
ING, 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
	ResDecode	String	Decoding result
	CounterGood		Number of successful read operations
	CounterBad		Number of failed read operations
	CounterBadSup-pressed		Number of suppressed read operations
	CounterWeak		Number of successful read operations

Table 7.1 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

7.2 EtherNet TCP/IP Communication with Easy Mode

The EtherNet 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: 50010**. 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 software.

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.

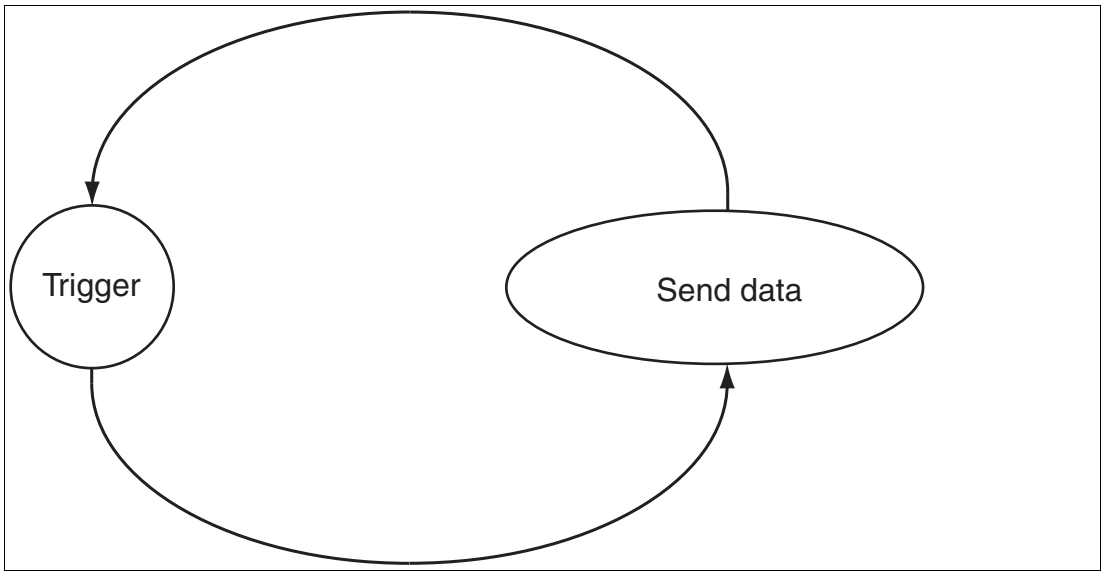


Figure 7.2 Easy Mode flow chart

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.	
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 Byte 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 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 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

Calculation of 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 =	Byte0	XOR	Byte1
X =	X	XOR	Byte2
X =	X	XOR	Byte3
...			
X =	X	XOR	Byte29
X =	X	XOR	Byte30

Byte 31 = checksum value X

Example telegram structure:

Response to the trigger when reading successfully:

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)

8 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

Error	Possible cause	Remedy
Communication with Vision Configurator is not possible	The IP address is configured incorrectly.	The factory default IP address is 192.168.2.5
	Older version of the Vision Configurator	The latest version of the Vision Configurator software can be found online at https://www.pepperl-fuchs.com.
Image is not displayed in Vision Configurator	The transfer of image captures to Vision Configurator is not activated.	Enable the **Image transfer active** option under the **Camera** tab, under the **Image** menu item (see chapter 5.6.4).
Code sheets cannot be read	The decoder is set to the wrong code sheet.	Select the correct code sheet type under the **OIT** tab under the **Code plate** menu item (see chapter 5.6.2).
	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 under the **OIT** tab, under the **Global** menu item. See chapter 5.6.2
	The image is too dark.	Change the settings for exposure and gain under the **Camera** tab, under the **Acquisition** menu item (see chapter 5.6.4).
	The lighting unit is set for the reading of a different code sheet.	Adjust the lighting unit according to the code sheet (see chapter 3.2.2).

Contacting the Service Center

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

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

2023-10

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

/ DOCT-6401B