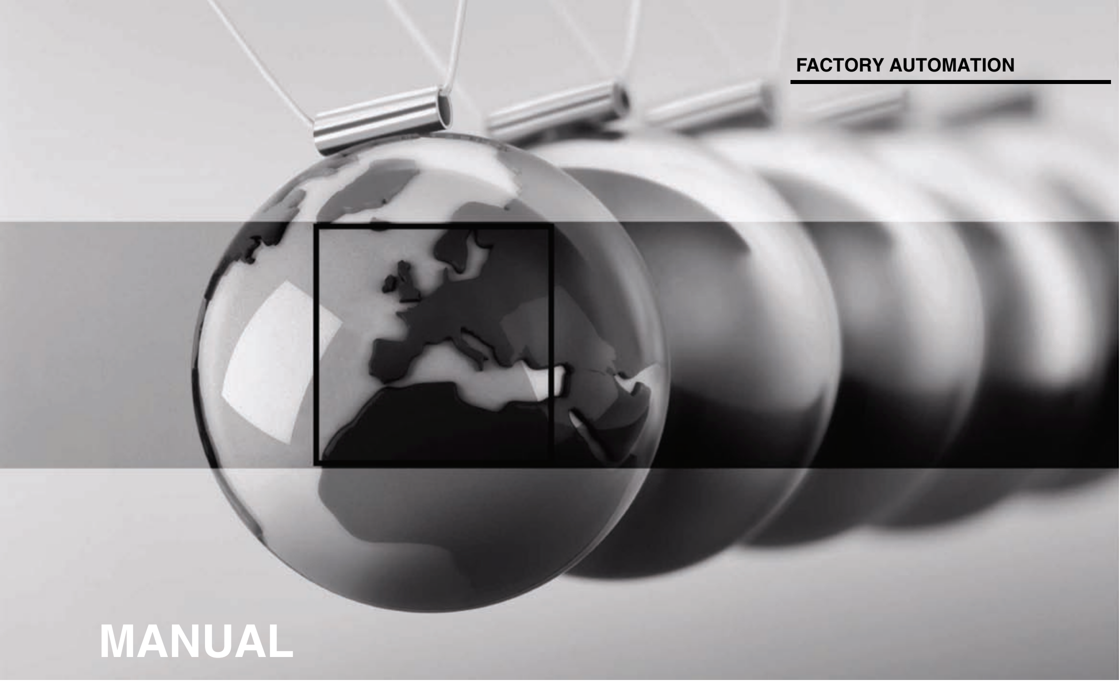
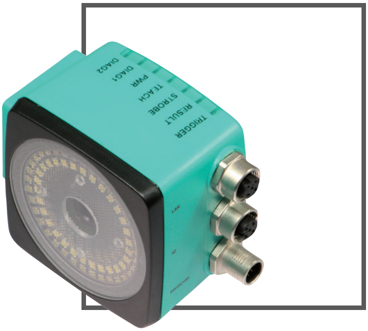


FACTORY AUTOMATION

MANUAL

BIS510-*

Vision sensors for sheet verification



CE

PEPPERL+FUCHS

SENSING YOUR NEEDS

1 Introduction

Congratulations

You have chosen a device manufactured by Pepperl+Fuchs. Pepperl+Fuchs develops, produces and distributes electronic sensors and interface modules for the market of automation technology on a worldwide scale.

Symbols used

The following symbols are used in this manual:

ℹ ***Note!***

This symbol draws your attention to important information.

➤ Handling instructions

You will find handling instructions beside this symbol

Contact

If you have any questions about the device, its functions, or accessories, please contact us at:

Pepperl+Fuchs GmbH
Lilienthalstraße 200
68307 Mannheim
Telephone: +49 621 776-4411
Fax: +49 621 776-274411
E-Mail: fa-info@pepperl-fuchs.com

2 Declaration of conformity

This product was developed and manufactured under observance of the applicable European standards and guidelines.

Note!

A Declaration of Conformity can be requested from the manufacturer.

The product manufacturer, Pepperl+Fuchs GmbH, D-68307 Mannheim, has a certified quality assurance system that conforms to ISO 9001.

ISO**9001**

3 Safety

3.1 Symbols relevant to safety

STOP

Danger!

This symbol indicates a warning about an immediate possible danger.

In case of ignoring the consequences may range from personal injury to death.

⚠

Warning!

This symbol indicates a warning about a possible fault or danger.

In case of ignoring the consequences may cause personal injury or heaviest property damage.

△

Caution!

This symbol indicates a warning about a possible fault.

In case of ignoring the devices and any connected facilities or systems may be interrupted or fail completely.

3.2 Intended use

The BIS system is a sheet verification sensor designed exclusively for the purpose of identifying sheet sequences. The sensor contains a camera, an illumination unit, an evaluation computer that generates digital input and output signals, and a network interface.

3.3 General safety instructions

Always operate the device as described in these instructions to ensure that the device and connected systems function correctly. The protection of operating personnel and plant is only guaranteed if the device is operated in accordance with its intended use.

The operating company bears responsibility for observing locally applicable safety regulations.

Installation and commissioning of all devices must be performed by a trained professional only.

User modification and or repair are dangerous and will void the warranty and exclude the manufacturer from any liability. If serious faults occur, stop using the device. Secure the device against inadvertent operation. In the event of repairs, return the device to your local Pepperl+Fuchs representative or sales office.

4 Product Description

4.1 Sheet verification sensor—use and application areas

The sheet verification sensor from Pepperl+Fuchs is a quick and simple solution for monitoring correct sheet sequences in gathering, folding, and binding machines. Sequences can be monitored either through image comparison or by reading barcodes printed on the sheets. The identification is performed in conjunction with each machine cycle at a paper throughput speed of up to 4 m/s and a maximum of 10 sheets/s. The sensor receives a trigger signal and returns digital signals that indicate whether the current sheet corresponds with the pattern that was taught in. An encoder signal can be used to delay the trigger signal. The device can be operated locally, as a stand-alone unit, or in a network where several sensors are connected together. The sensor contains a camera, an illumination unit, an evaluation computer that generates digital input and output signals, and a network interface. The sensor is connected to a machine control system via these interfaces. Time-critical signals such as triggers and results are transmitted via the digital inputs/outputs. The network interface can be used for sending commands to the sensor to change the operating mode and configure different parameters, and for loading images.

The sheet verification sensor is a single compact unit: the camera, laser unit, and a DSP board for digitalizing and processing captured image information are enclosed in a single housing.

4.2 Displays and controls

The illumination unit contains 7 LED indicators that provide information on the status of the device.

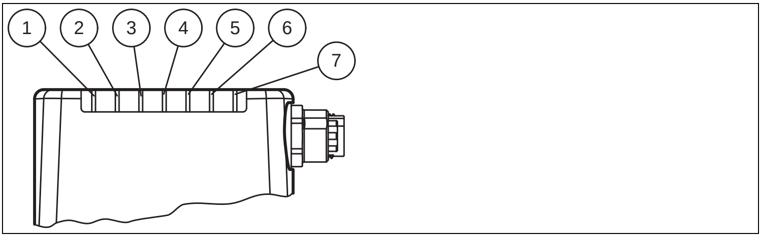


Figure 4.1 Displays and controls

1 **DIAG 2**
Yellow LED. Generates different flashing sequences to signal diagnostic messages.

2 **DIAG 1**
Yellow LED. Generates different flashing sequences to signal diagnostic messages.

3 **Power (PWR)**
Lights up green when the sensor is ready for operation.

4 **Teach in**
Lights up yellow when the teach-in process starts.

5 **Reading process triggered (BAD)**
Lights up yellow if the reading was unsuccessful.

6 **Reading process triggered (GOOD)**
Lights up yellow if the reading was successful.

7 **Trigger sensor (TRG)**
Lights up yellow when a trigger sensor is connected.*

4.3 Interfaces and connections

The device includes the following connections:

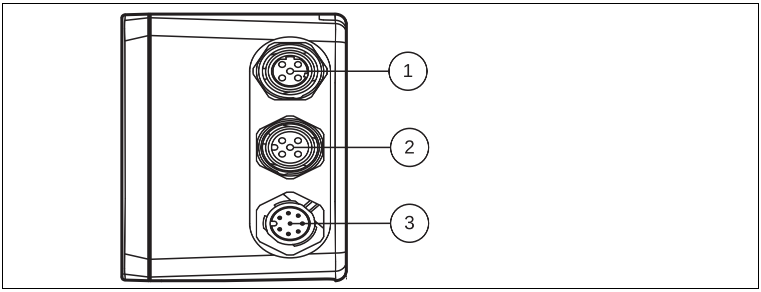


Figure 4.2 Device connections

1 Network (4-pin M12 socket)

2 Input IO (5-pin M12 socket)

3 Power supply, inputs and outputs (8-pin M12 connector)

Power supply

There is an 8-pin M12 plug on the side of the housing to connect the power supply and the inputs and outputs. The following diagram shows the pin assignment:

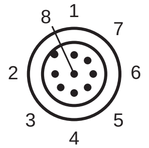


Figure 4.3 Connection layout for operating voltage and inputs and outputs

1	Trigger IN
2	+UB
3	Good OUT
4	Bad OUT
5	Start_Teach IN (applying a high level starts a Teach-in process)
6	Ready OUT
7	GND
8	Teach_Active OUT

IO input and encoder input

There is a 5-pin M12 socket on the side of the sensor housing. An encoder or trigger sensor may be connected here depending on the version. The following diagram shows the pin assignment:

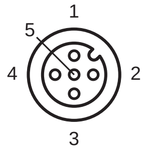


Figure 4.4 IO input connection layout

1	24 V power supply
2	IN1
3	Ground (GND)
4	IN2
5	Not used

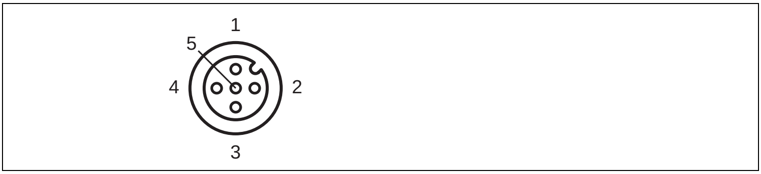


Figure 4.5 Encoder input connection layout

1	24 V power supply
2	B
3	Ground (GND)
4	A
5	Not used

Network

There is a 4-pin M12 socket on the back of the housing to connect to the network. The following diagram shows the pin assignment:

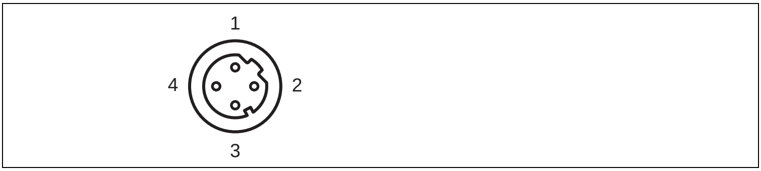


Figure 4.6 Network connection layout

1	TX+ Ethernet
2	RX+ Ethernet
3	TX- Ethernet
4	RX- Ethernet

4.4 Delivery package

- BIS510-*

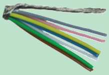
4.5 Accessories

Various accessories are available.

4.5.1 Power supply

Use the following connection cable to connect the power supply, inputs and outputs to the sensor.

M12 connection cables

	Material	Length	Cable end, field attachable 
8-pin M12 socket, straight 	PUR	2 m	V19-G-2M-PUR-ABG
		5 m	V19-G-5M-PUR-ABG
		10 m	V19-G-10M-PUR-ABG

Field-attachable M12 connectors

Model number	Description	mm^2	Cable dia.
V19-G-ABG-PG9	8-pin M12 socket, straight	max. 0.75	5 to 8 mm

Other lengths on request.

4.5.2 Network cable

The sensor is connected to the network using an M12 connector.

Designation	Description
V45-G	RJ45 network connector, field attachable
V1S-G	4-pin M12 connector, field attachable
V1SD-G-2M-PUR-ABG-V45X-G	Connection cable, RJ45 network connector with M12 plug, cross-over, 4-pin
V1SD-G-2M-PUR-ABG-V45-G	Connection cable, RJ45 network connector with M12 plug, 4-pin

4.5.3 Inputs and outputs

The inputs and outputs of the sensor are connected via an M12 connector.

Designation	Description
V15S-G-5M-PUR-ABG	Male cordset, M12, 5-pin, PUR cable, shielded cap nut

5 Installation

5.1 Preparation

Unpacking the unit

1. Check that all package contents are present and undamaged.

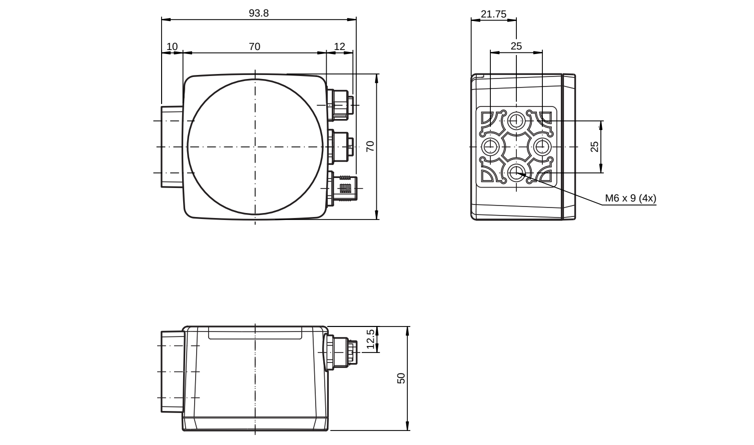
   ↳ If anything is damaged, inform the shipper and contact the supplier.
2. Check that all items are present and correct based on your order and the shipping documents.

   ↳ If you have any questions, please contact Pepperl+Fuchs.
3. Keep the original packing material in case you need to store or ship the unit at a later time.

5.2 Mounting the device

The device has four symmetrically positioned M6 threads on the base of the housing to allow easy installation of the sensor in your plant.

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



The vision sensor is mounted at a specific operating distance above the surface of the sheet.

The following illustration shows an example of correct mounting:

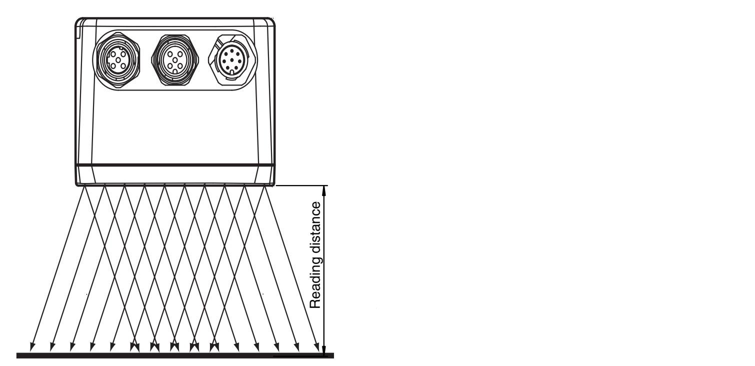


Figure 5.1 Installation example

5.3 Connecting the device

Connecting the supply voltage

To supply voltage to the sensor, proceed as follows:

1. Plug the 8-pin M12 socket into the connector provided on the side of the housing.
2. Screw the cap nut onto the connector as far as it will go.

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

Note!

Documenting the network configuration

The sensor communicates with the connected evaluation system using the TCP/IP protocol. To ensure proper communication, you must record all the changes made to the network configuration.

Note!

Network cabling

Use a crossover network cable to connect the sensor directly to a PC. If you are operating the sensor within a network, use a twisted-pair network cable to connect the sensor to the network.

Establishing a network connection

To establish a network connection, proceed as follows.

1. If you are using a network cable with an RJ45 network plug at one end and a 4-pin M12 socket at the other, insert the 4-pin M12 socket in the connector on the side of the sensor.
2. The sensor is delivered with a fixed IP address (192.168.2.3). To facilitate communication within the network, you must configure your network. The configuration data can be found in the network configuration overview.

Resetting the IP address

If the IP address of the Vision Sensor has been changed, you can set the default IP address.

1. Connect the Vision Sensor to the power supply.

   ↳ The Vision Sensor powers up.
2. Wait until the LEDs flash.
3. Simultaneously press operating buttons 1 & 2 for approx. 2 seconds.

   ↳ The LED flash sequence changes.
   The Vision Sensor powers up again with the default IP address.

Connecting a trigger sensor

To connect a trigger sensor, proceed as follows.

1. Plug the 5-pin M12 plug into the socket provided on the housing.
2. Screw the threaded nipple onto the connector as far as it will go

   ↳ to secure the trigger cable against inadvertent removal.

5.4 Storage and transport

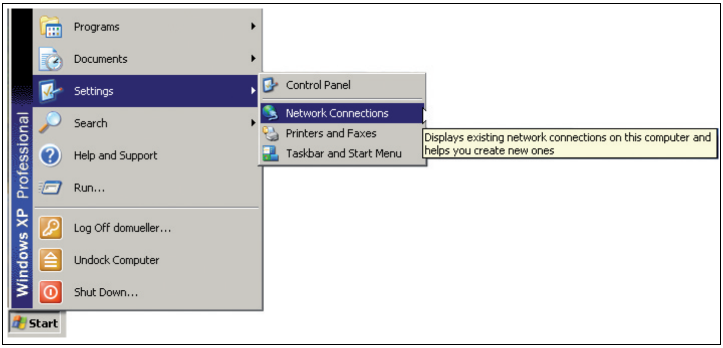
For storage and transport purposes, package the unit using shockproof packaging material and protect it against moisture. The best method of protection is to package the unit using the original packaging. Furthermore, ensure that the ambient conditions are within allowable range.

6 Commissioning

6.1 Assigning an IP address to a network connection using Windows XP

To assign an IP address to a network connection using Windows XP, proceed as follows.

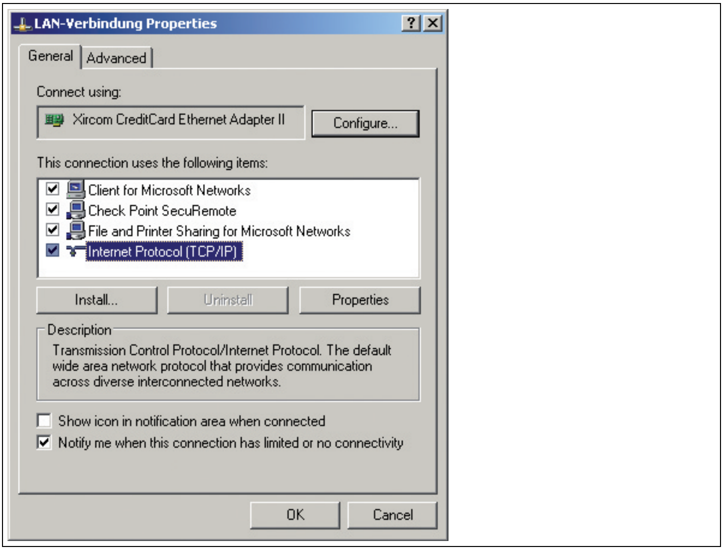
1. First select "**Network Connections**".



Displays existing network connections on this computer and helps you create new ones

2. Then open the required connection by double clicking on it.

↳ The Properties dialog box for the relevant connection will open.



Connect using:
Xircom CreditCard Ethernet Adapter II

This connection uses the following items:

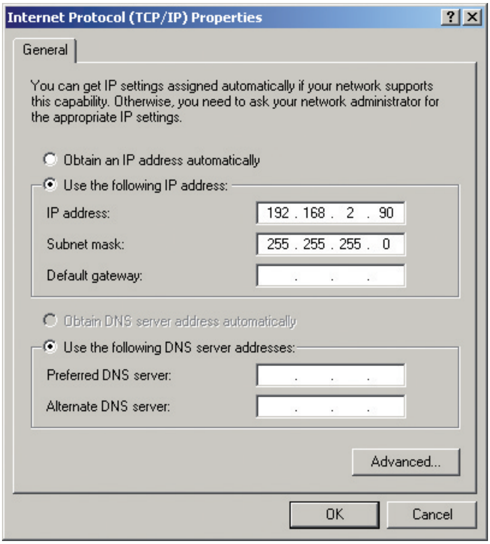
☑ Client for Microsoft Networks
☑ Check Point SecuRemote
☑ File and Printer Sharing for Microsoft Networks
☑ Internet Protocol (TCP/IP)

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

□ Show icon in notification area when connected
☑ Notify me when this connection has limited or no connectivity

3. Select the "**Internet Protocol (TCP/IP)**" element from the Properties dialog box by double clicking on it.

↳ The TCP/IP properties dialog box will open.



Internet Protocol (TCP/IP) Properties

General

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

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 2 . 90

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

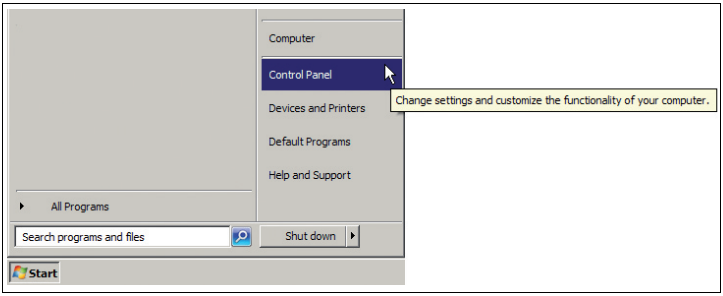
4. In the TCP/IP properties dialog box, activate "**Use the following IP address**".
5. Enter an IP address which only differs from the sensor IP address in the very last segment.
6. Enter 255.255.255.0 as the subnet mask.
7. Then confirm your entries on the TCP/IP properties page and the LAN connection properties page using "**OK**" and "**Close**".

↳ This completes the network configuration and the sensor can be used.

6.2 Assigning an IP address to a network connection under Windows 7

To assign an IP address to a network connection under Windows 7:

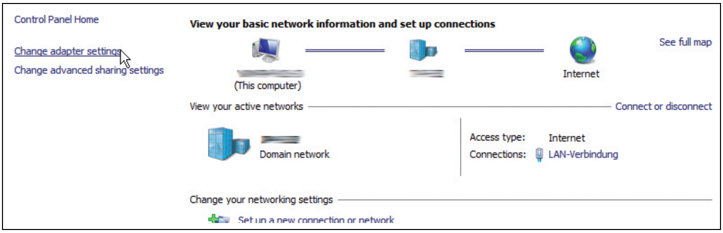
1. First, select "**Control Panel**" from the **Start** menu:



Change settings and customize the functionality of your computer.

2. Select the **Network and Sharing Center** option:

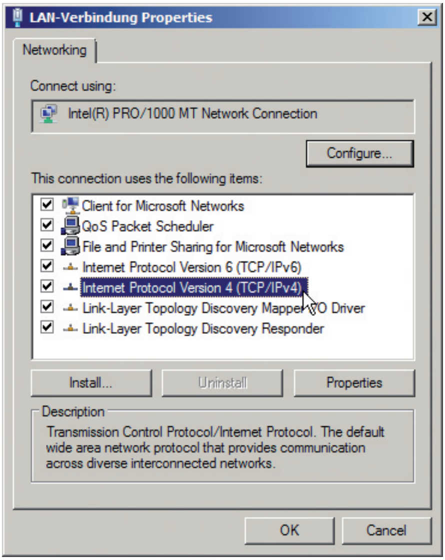
3. In the Network and Sharing Center window, select **Change adapter settings**


Control Panel Home
Change adapter settings
Change advanced sharing settings
View your basic network information and set up connections
See full map
(This computer)
Internet
View your active networks
Connect or disconnect
Domain network
Change your networking settings
Set up a new connection or network

Access type: Internet
Connections: LAN-Verbindung

4. Then open the required connection by double-clicking on it.

↳ The Properties dialog box for the relevant connection will open.



LAN-Verbindung Properties

Networking

Connect using:

Intel(R) PRO/1000 MT Network Connection

This connection uses the following items:

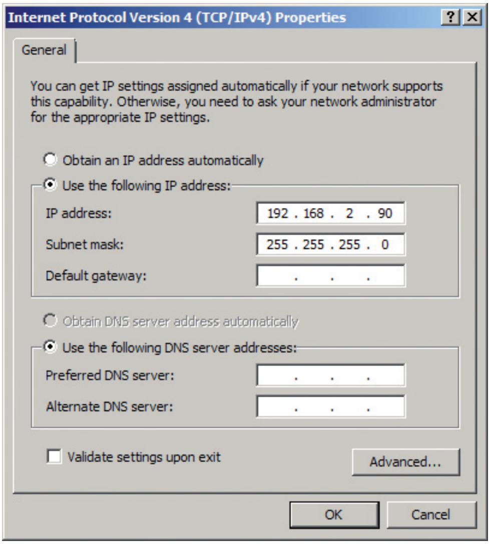
- ☑ Client for Microsoft Networks
- ☑ QoS Packet Scheduler
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ Internet Protocol Version 6 (TCP/IPv6)
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☑ Link-Layer Topology Discovery Mapper I/O Driver
- ☑ Link-Layer Topology Discovery Responder

Description

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

5. Select the "**Internet Protocol Version 4 (TCP/IPv4)**" element from the Properties dialog box by double-clicking on it.

↳ The TCP/IP properties dialog box will open.



Internet Protocol Version 4 (TCP/IPv4) Properties

General

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

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 2 . 90

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit

6. Activate "**Use the following IP address**" in the TCP/IP properties dialog box.

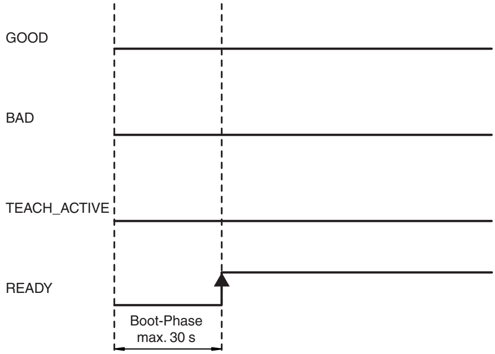
7. Enter an IP address that differs from the sensor IP address in the very last segment only.
8. Enter 255.255.255.0 as the subnet mask.
9. Then click "**OK**" and "**Close**" on the TCP/IP properties page and the LAN connection properties page to confirm your entries.

↳ This completes the network configuration and the sensor can be used.

6.3 Sensor start

Start sequence

The GOOD, BAD, TEACH_ACTIVE and READY outputs are set to low when the sensor starts up. The READY output adopts a high status as soon as the sensor is ready for operation (max. 30 seconds). The READY output reverts to low when a hardware fault occurs.



6.4 Functionality

The sensor captures at least one image from each sheet scheduled for inspection. Depending on the operating mode, this image is either compared with a previously taught-in image (“image comparison”) or a code in the barcode is evaluated and compared with a taught-in or predetermined code (code comparison).

The following operating modes are available:

- Image comparison
- Code comparison

6.4.1 Image comparison

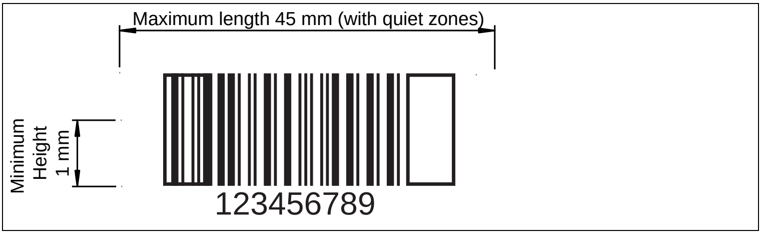
With image comparison, a sheet's image is taught into the system and all subsequent sheets are compared with this image. If the sheets are sufficiently similar, the system issues an "OK" evaluation, otherwise the evaluation is "NOK" (not ok). During image comparison, a shift in the sheet's position of up to 10 mm with respect to the taught-in sheet is permitted in a horizontal and vertical direction.

6.4.2 Code comparison

A code must be printed on the lower or upper edge of the sheet being checked. The sensor can read this barcode. The entire code must be positioned within the image field of the sensor. The barcode modules must contrast sufficiently with the background to guarantee reliable decoding. Pepperl+Fuchs defines a grayscale value difference (e.g., 20 gray scales) for this purpose. Each critical case must be tested individually because end users will not understand the meaning of this information. The code can be aligned horizontally or vertically within the image field. The sensor offers two methods for reading codes: **code comparison** or **code reading**. Code reading is available as an option only.

- **Code comparison:** The barcode is compared with a specified reference barcode. The sensor issues the result "Good" if the barcode that was read matches the stored barcode. The sensor issues the result "Bad" if the two barcodes do not match or the barcode could not be read. The barcode used for the comparison can either be specified as a value or defined by scanning in a barcode on a sheet.

  The barcode you intend to read must have the following dimensions:



- Minimum bar width: 0.3 mm

The following barcode types can be read:

- 2/5 interleaved
- Code39
- Code128
- Pharmacode
- Data Matrix (optional)

6.5 Teach-in process

The Teach-in process is activated by setting the **START_TEACH** input or using an XML command. The sensor indicates that the Teach-in process is running by setting the **TEACH_ACTIVE** signal. One or more images are captured prior to teaching in. Completion of the Teach-in process is indicated by resetting the **TEACH_ACTIVE** output.

The following processes are active during the Teach-in:

- Determination of the sheet length
- Determination of the best exposure time (flash duration)
- Selection of a suitable image capture position on the sheet
- Calculation of the code content or the pattern from the best image capture position

The following Teach-in modes can be selected:

- **Single image (external trigger):** The next image is taught using the standard flash time.
- **Single fix position (rotary encoder trigger + external trigger):** Image capture at a specified fixed position. The settings for this are entered on the Rotary Encoder tab. A rotary encoder delays the flash position relative to the edge of the sheet. To detect the edge of the sheet, an external trigger is required. Only after the triggering the encoder signals are evaluated.
- **Triple fix position (rotary encoder trigger):** Similar to single fix position, except that one image per sheet is captured on three sheets. The exposure time is varied and the best image is used as the reference image.
- **Automatic position (rotary encoder trigger):** A maximum of five images are captured, spread over the length of three sheets with different exposures. The image with the best modulation and content is identified from all images captured and taught in as a pattern for comparison.
- **Triple image (external trigger):** Three images are captured at the trigger position to determine the optimum exposure time and the one with this time is taught in as the reference image.
- **Direct Teach (no trigger):** Setting the Teach signal triggers one or three simultaneous image captures depending on the operating mode. No external trigger is needed.

A maximum of 15 images can be captured during the Teach-in process, from which the image with the largest number of clear edges is selected as the best image. If none of the images captured for the Teach-in contains a sufficient number of edges, the sensor operates in grayscale value mode. Setpoint values and tolerances for the following statistical characteristics are then defined in the pattern image:

- Minimum grayscale value
- Maximum grayscale value
- Average grayscale value
- Contrast (difference between maximum and minimum grayscale values)
- Variance

During the inspection, a check is performed to determine whether the values of the characteristics in the current image fall within the tolerances.

The taught-in data is transferred to the sensor's flash memory after the Teach-in process and is stored permanently in the sensor. This means inspections can continue without requiring another Teach-in operation if there is an interruption to the power supply.

Teaching in a pattern sheet

1. Start the Teach-in process by sending a rising pulse edge at the **TEACH** input or send the corresponding XML command to the sensor. The sensor indicates that the Teach-in process is running by setting the **TEACH_ACTIVE** signal.
2. Depending on the Teach-in operating mode, one or more image captures are triggered at the trigger input by one or more rising edges.
3. Completion of the Teach-in process is indicated by resetting the **TEACH_ACTIVE** output.

↳ The Teach-in process is completed.

6.6 Additional functions

Statistics

Several values are logged in the sensor and are reset to zero during each Teach-in process. The values can be read out using a result message. The following information is logged:

- Number of sheets
- Number of sheets with "bad" image comparison
- Number of sheets with unreadable barcode
- Number of sheets with incorrect barcode

The counters read zero when the operating voltage is switched off.

Active mode

On the data interface, the BIS510 is usually passive, i.e., it never sends data automatically but rather responds only to requests. When active mode is activated, the sensor sends the fault pattern automatically whenever faults are detected.

Fault pattern memory

The BIS510 operates with a circular buffer for fault patterns. The circular buffer can store up to 5 patterns. If a new fault occurs, the new fault pattern overwrites the oldest pattern.

7 Operation

7.1 Inspection operating state

Pulses at the TRIGGER input trigger an image capture and a subsequent inspection is performed. A high signal at the GOOD output or the BAD output indicates the result of the inspection.

To start an inspection, proceed as follows:

Image inspection

1. As soon as a rising edge is present at the trigger signal, an image is captured and a subsequent inspection is performed.
2. After the image is evaluated, the GOOD output or the BAD output issues a high signal.

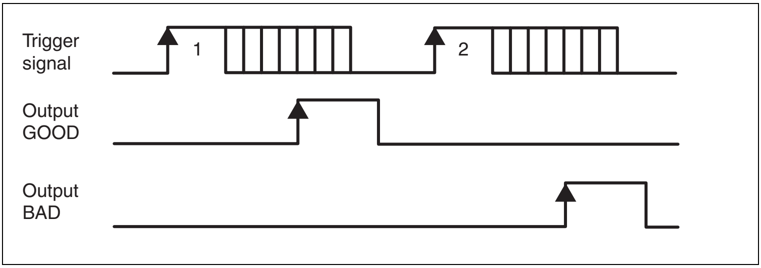


Figure 7.1 Signals in "Inspection" operating state

The sensor can store a maximum of 10 results (good/bad) temporarily in a shift register, i.e. a maximum of 10 products may be located between the sensor and ejection point.

7.2 Teach-in operating state

To teach in a pattern, proceed as follows:

Teaching in patterns

1. Send a rising edge to the **START_TEACH** Input. The sensor is set to teach-in mode and the **TEACH-ACTIVE** output is set to indicate this.

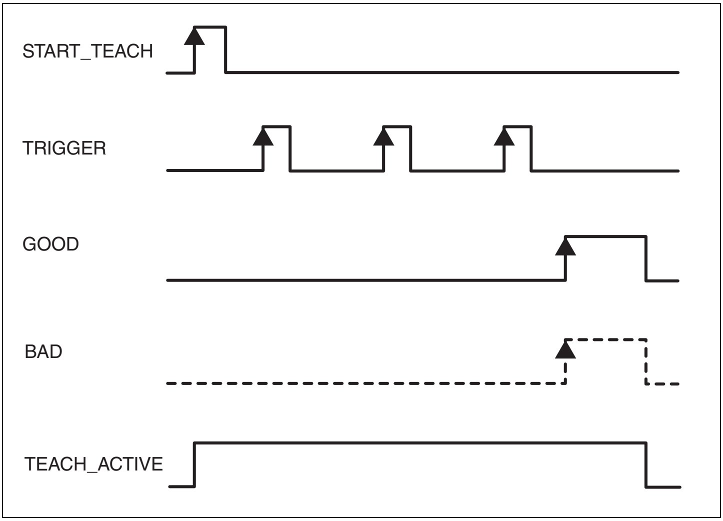


Figure 7.2 Signals in "Teach-In" operating state

2. Depending on the teach-in mode, one or more image captures are triggered at the trigger input by rising edges.
3. The sensor teaches in the best of the captured images and uses it as a pattern.
4. The **TEACH_ACTIVE** output is set during the teach-in process.

↳ The teach-in process is completed when the **GOOD** or **BAD** output is set to high.
"GOOD" means that image detection was successful.
"BAD" means that image detection was unsuccessful and the sensor is operating in gray value comparison mode.

Resetting the **GOOD** or **BAD** output also resets the **TEACH_ACTIVE** output.

7.3 Network interface

The network interface is used to transfer data (parameters, images, software updates) and commands in both directions (host to sensor and sensor to host).

XML strings are used to transfer parameters, commands and results.

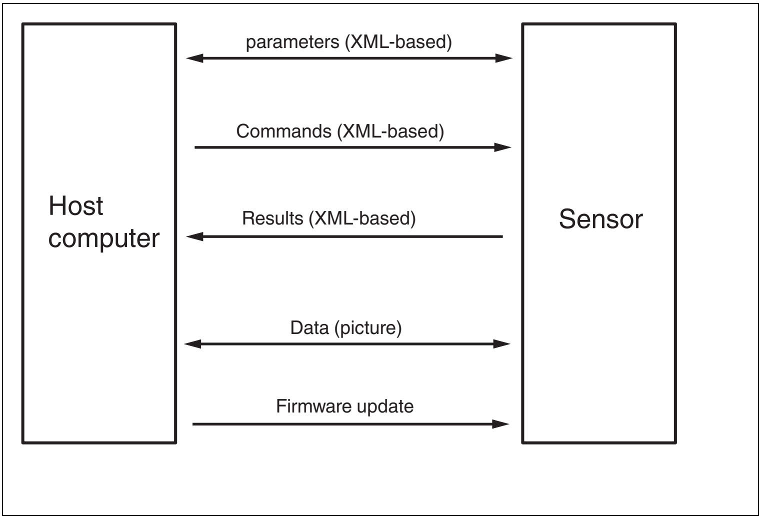


Figure 7.3 Network interface XML-Stings

Communication is established through TCP/IP via port 50005. The default IP address is 192.168.2.3.
Each data package consists of a pre-header and the actual data:

Pre-header (length: 12 bytes)	Data (length varies)

7.4 Software interface

A **.NET 2.0 based** software interface is provided for easy integration with PC software. This is provided in the form of .NET DLLs and handles the communication to the sensor. To do this, integrate the interface DLLs into the programming environment (⟶ see Figure 7.4 on page 27) and run the programming lines indicated below.

All the examples relate to the Visual Studio 2008 programming environment and to the C# (C Sharp) programming language.

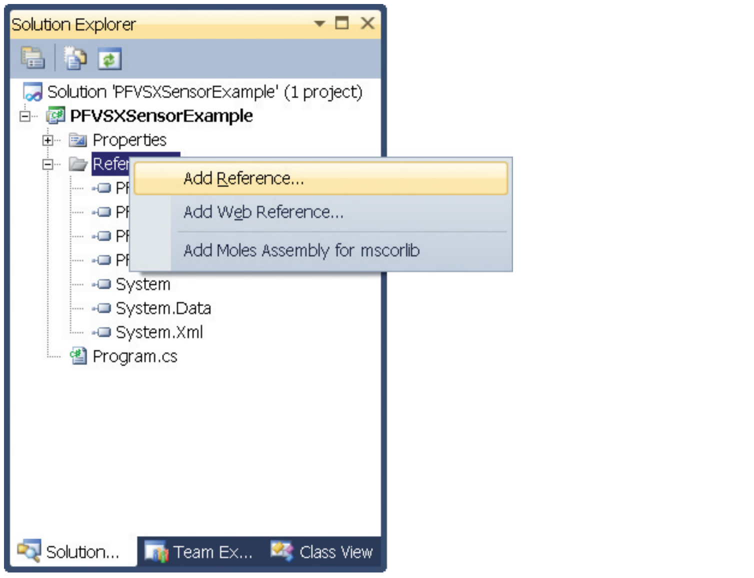


Figure 7.4 Integration interface DLL

Below is an example 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);
      System.Threading.Thread.Sleep(2000); //Wait for Connection
      sensor.SetSingleParameter("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 work with the VSX protocol. To do this, the library establishes a connection to the sensor and handles the communication according to the protocol. The user is provided with functions for setting parameters on the sensor, retrieving parameter values from the sensor, and saving and loading whole parameter sets both locally and on the sensor. The user can also receive sensor images.

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

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

Creating an object

Create an object to access the library functions.

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

Retrieving received parameter data

Received parameter data is saved in a list in the sensor. You can retrieve individual parameter data sets from this list with the following function:

```
string GetSingleParameter(string configId, string
parameterId)
```

Synchronizing sensor data

To synchronize the list with the current sensor data, you can call up the following function:

```
void GetAllParametersFromSensor()
```

Displaying changed data

If data has been received from the sensor and a change has taken place in the internal list, this is shown 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 from the sensor that has been saved with `SaveSensorSettings`.
After the parameter set has been received from the sensor, a `ParameterDataReceived` event is triggered with `Modifier = LOAD_Data`.
The data can then be called up using the `GetSingleParameter` method.

```
void GetNetworkSettings()
```

Requests the sensor's current connection settings.
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 the sensor sends log messages or not. 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 sensor's current parameter set. After the parameter set has been received from the sensor, a `ParameterDataReceived` event is triggered with `Modifier = None`. This indicates that the internal list has been updated with the parameter set. Subsequently, you can retrieve individual parameters via `GetSingleParameter`.

```
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 GetSingleParameter(string configId, string parameterId)
```

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. To update the list, `GetAllParametersFromSensor` must be called up.
If the specified parameter does not appear in the list, an `InternalErrorEvent` is triggered with `ErrorType = PARAMETER_NOT_FOUND`.
`configId`: Configuration ID of a parameter
`parameterId`: Parameter ID of a parameter
Response: The current value of the specified parameter

```
void SetSingleParameter(string configId, string parameterId,
string value)
```

`configId`: Configuration ID of a parameter
`parameterId`: Parameter ID of a parameter
`value`: New value for the parameter
Sets a single parameter, defined by the configuration ID and parameter ID, to the specified value. If the sensor changes only this one value, a `SingleDataReceived` event is triggered. The set value can be taken from this and checked to make sure that the value on the sensor was set appropriately. In certain cases, due to the set value, it is possible for other dependent parameters on the sensor to be changed. In this case, a `ParameterDataReceived` event is triggered with `Modifier = NONE` and all the parameters should be checked for new values with `GetSingleParameter`.
If the specified parameter is not present in the list, an `InternalErrorEvent` is triggered with `ErrorType = PARAMETER_NOT_FOUND`.

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

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

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

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

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

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

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

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

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

frameCounter: Number of the image associated with the graphics
Response: List of overlay graphics for the specified image
Returns a list of graphics for an image overlay. Receipt of these graphics is indicated by the ShapeDataReceived(ushort frameCounter, List<IElement> shapeList) event. 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 the 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: 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 is triggered with Modifier = LOAD_DEFAULT_DATA.

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

Saves the current parameter set to the specified file. The current parameter set is called up from the sensor and saved after receipt. After a successful save, a SaveDataOnHdd event is triggered. If an error occurs during a save, an InternalError event is triggered with ErrorType = SAVE_FILE_ERROR.
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 is triggered with Modifier = NONE. If the data set cannot be loaded correctly, an InternalError event is triggered with ErrorType = LOAD_FILE_ERROR.
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. The connection must be reestablished with the new IP via Connect.

```
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 has been triggered, the internal list of parameters has been updated and the individual parameter values should be retrieved with `GetSingleParameter`.

```
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 items of result data can be queried using `GetResultParameter`.

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

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

```
event AcceptReceived()
```

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

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

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

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

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

```
event SaveDataOnSensorReceived()
```

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

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

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

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

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

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

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

The following event is supported by laser triangulation sensors only:

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

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

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

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

```
event SaveDataOnHddReceived()
```

This is triggered when the current parameter set has been successfully saved into 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 not be loaded from the file or if the file contains invalid data
- In all cases in which `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 `InternalError` event.

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

See `ParameterDataReceived` event.

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

See `LogDataReceived` event.

7.4.1 Configuration overview

The parameters in the table below can be set using the `SetSingleParameter` and `GetSingleParameter` methods in the software interface. Each parameter is identified by a **config ID** and a **parameter ID**.

Configuration parameters

Config ID	Parameter ID	Value	Description
General	OptionSendErr	1/0	Activate/deactivate automatic sending of fault pattern
	OptionAutoResult	1/0	Activate/deactivate automatic sending of result
	OptionShowHWTrigger	1/0	Activate/deactivate log entry with time stamp for trigger signal
	PosMark	1/0	Activate/deactivate image field marking
	IPAddress		IP address configuration
	SubNetMask		Subnet mask configuration
	Gateway		Gateway configuration
	EnableBtn1	1/0	Activate/deactivate button 1 on the rear of the sensor
	EnableBtn2	1/0	Activate/deactivate button 2 on the rear of the sensor
Camera	FlashTime	0 ... 255	Exposure time and flash time setting in µs
	Gain	0 ... 255	Gain setting
	Snapshot	1/0	Activates camera shot with the current settings
	UseCamSettings	1/0	Uses the set values for the next Teach-in process
	GetCamTeachParams	1/0	Sensor value readings
	StartLive	1/0	Starts live image transmission
	StopLive	1/0	Stops live image transmission
	CheckImage	1/0	Activate/deactivate evaluation of every image captured in live image mode
	DisplayPause	1 ... 10000	Setting for the pause time between two image captures
Search Params	Sensitivity	0 ... 5	Pattern comparison sensitivity Level 0 -> weak Level 5 -> strong
Command	TriggerStart	1	Triggers an image capture
	ReloadData	1	Restores the taught data after a TeachStart
	GetLastImg	1	Requests the current image
	GetErrImg	1	Requests the last fault pattern
	GetNextErrImg	1	Displays the next fault pattern
	GetErrorImgListSize	1	Requests the number of saved fault patterns
	GetLogMsg	1	Display all log messages

Config ID	Parameter ID	Value	Description
VOS510	TeachMode	TEACH_MODE_1_1_1	Teach-in of an image on a sheet with a specified exposure time
		TEACH_MODE_IMAGE_AUTO	Teach-in of one image per sheet on a total of three sheets with varying exposure time. The best image is then adopted
		TEACH_MODE_TRIPLE_IMAGE	Teach-in of a total of five images per sheet on a total of three sheets with varying exposure time. The best image is then adopted
	InspectionMode	PATTERN_INSPECTION	Image comparison inspection mode
		BARCODE_INSPECTION	Barcode comparison inspection mode
	DirectTeach	0/1	Activate/deactivate direct teach mode Teach pulse or Teach command triggers an image capture for Teach-in
	CapturePositionOnPage	1 ... 5	Number of positions per sheet for automatic Teach-in
	TeachTimeout	100 ... 1000000	Timeout for Teach-in (image comparison and barcode)
Barcode	CompareCode		Entry for the reference code comparison string
	Update	0/1	Activate/deactivate setting of the last barcode read as the reference
	Timeout	1 ... 1000000	Timeout for barcode reading and barcode comparison
	StartAutomatic	0/1	Activates or deactivates the automatic search
	Orientation	hor	Barcode and Pharmacode search horizontal only
		ver	Barcode and Pharmacode search vertical only
	Code39	0/1	Activates or deactivates Code39
	Code128	0/1	Activates or deactivates Code128
	Code13	0/1	Activates or deactivates Code13
	Code25	0/1	Activates or deactivates Code25
	CheckSum	0/1	Activates or deactivates checksum evaluation for Code 2/5 interleaved
	PharmaCodeOnly	0/1	Activates or deactivates Pharmacode. When activated, the barcode is deactivated
	MinCodeLen	0 ... 20	Minimum number of digits in the Pharmacode
	MaxCodeLen	0 ... 20	Maximum number of digits in the Pharmacode
	MinOkLine	0 ... 100	Minimum number of lines with identical decoding result for Pharmacode
	PharmaDir	BOTTOM	Interpretation of Pharmacode: Bottom/right = from right or bottom to left or top
		TOP	Interpretation of Pharmacode: Top/left = from left or top to right or bottom
	SearchStart	BOTTOM	Search direction for Pharmacode: Bottom/right = from right or bottom to left or top
		TOP	Search direction for Pharmacode: Top/left = from left or top to right or bottom

Config ID	Parameter ID	Value	Description
Rotation Encoder	Resolution	1 ... 9999999 pulses/m	Resolution of rotary encoder Default value: 10000 pulses/m
	TrigDistance	0 ... 9999999 mm	Distance between the trigger sensor and the central read field on the sensor Default value: 0
	FixPos	0 ... 9999999 mm	Specifies a fixed position at which reading is to be performed
	MaxSpeed	10 ... 10000 mm/s	Maximum speed during reading Default value: 3000 mm/s
	TrigPolNeg	1	Activate/deactivate initiation of a trigger signal for a low level
	DelayActive	0	Activate/deactivate an output delay
	ResultDelay	1 ... 2000 mm	Entry for the output delay
	PulseLength	1 ... 1000 mm	Entry for the pulse length
	EncoderPos	0 inc	Current rotary encoder position in [inc]
	ReadEncoderPos	0/1	Activate/deactivate reading of rotary encoder position

Table 7.1 Configuration parameters

7.4.2 Result overview

Accessing the result data

The sensor transmits result data in the following situations:

- After transferring each image
- After completing a Teach-in process
- After a sensor status request
- After every inspection

The result data is contained in the `CONFIGURATION Id=“Result”` node. The node contains two different result parameters depending on the situation.

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

The shape data is plotted directly in the image. It consists of either geometrical shapes, such as colored rectangles (ElementShapeRectangle), or a type label (ElementShapeText). It contains position and size properties. The result data is exclusively text data, which is transmitted in barcode mode and after Teach-in.

Result data

The result data is transmitted with the `SensorResultDataReceived` event and can be retrieved in the event handler using the `GetResultList` and `GetResultParameter` method.

The following event and parameter IDs are permitted:

Result data

Event	Parameter ID	Value	Description
Trigger	BCString		Barcode read data
	BCType	Code128 Code39 Code13 2/5 Interleaved Pharmacode	Displays the read barcode symbology
	BCTime	0 ... 200 ms	Barcode decoding time in milliseconds
	BCLQuality	0 ... 100%/0 ... 100%	Centrality of the barcode position (horizontal/vertical) 100%/100% -> The barcode read is located exactly in the center.
Teach	Check	"SEARCH_GOOD" "SEARCH_NORMAL" "SEARCH_WEAK"	Simplified quality output split into three sections
	TeachIndex	1 ... 5	Reference image number
	TeachPosition		Reference image position
	TeachQuality	1 ... 16	Detailed teach quality output 1 -> poor quality, 16 -> best quality

Table 7.2 Result data

Shape data

The shape data is transmitted with the `ShapeDataReceived` and `ImageDataReceived` events. The `GetShapes` method is used to retrieve the data.

The following parameters are transmitted:

Shape data

Type	Parameter ID	Value	Description
Element ShapeText	ImageType	ActImage ErrImage TeachImage	Current image Fault pattern Reference image
	OperatingMode	Correlation Greylevel CodeCompare	Image comparison mode Grayscale value mode. Image comparison using grayscale values. Barcode comparison mode
	Result	PatternGood PatternBad BarcodeGood BarcodeNoRead	Image comparison good Image is not identical to reference image Barcode comparison is good Barcode could not be read
	BCCompare		Comparison barcode data
	BCString		Data for the barcode read
	BCType	Code128 Code39 Code13 2/5 Interleaved Pharmacode	Displays the read barcode symbology
	BCTime	0 ... 100 ms	Barcode decoding time in milliseconds
	BCLQuality	0 ... 100%/ 0 ... 100%	Centrality of the barcode position (horizontal/vertical) 100%/100% -> The barcode read is located exactly in the center.
Element Shape Rectangle	Check		Specifies the position of the partial image in the read image
	Ref		Specifies the anchor point in the read partial image used for comparison with the reference image

Table 7.3 Shape data

8 Vision Configurator Software

The sensor is commissioned and operated with the Vision Configurator software.

The Vision Configurator software provides you with a user-friendly interface for easy operation of the sensor. Standard tasks include establishing connections to the sensor, specifying the operating parameters, saving data sets, as well as the transfer and display of data and error diagnostics.

Establishing a Network Connection

To establish a network connection with the sensor, proceed as follows:

1. Supply the sensor with power.
2. Start the Vision Configurator software.
3. Select the connected sensor.
4. Check that the correct IP address has been entered.
5. Enter your user name and password.

   ↳ A connection to the sensor is established.

Note!

Documenting the Network Configuration

The sensor communicates with the connected evaluation system using the TCP/IP protocol. To ensure proper communication, you must record all changes made to the network configuration.

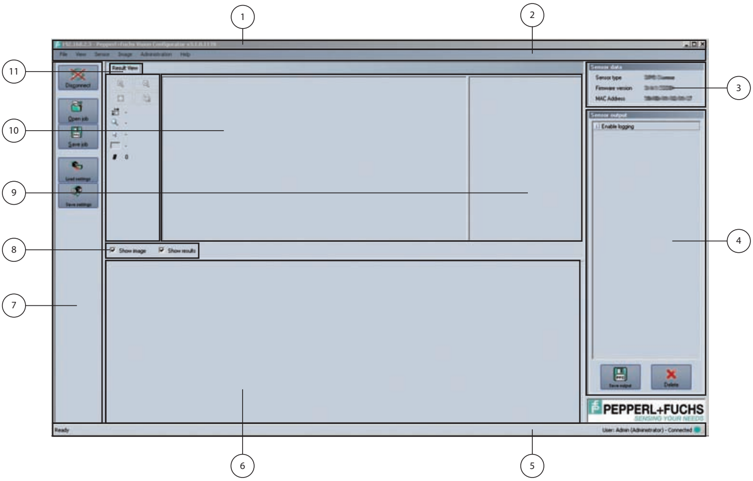
An up-to-date description of the Vision Configurator software can be found at http://www.pepperl-fuchs.com.

8.1 Application window structure

The application screen opens after you log in.

Note!

The individual functions depend on the type of sensor connected and the current authorization level and are, therefore, not always all visible.



The software is designed to be similar to most Windows applications.

1	Title bar	■ Shows the IP address, the software name, and the version number ■ Contains the **Minimize/Maximize/Close** buttons
2	Menu bar	■ Displays all the menus in the program ■ Provides an overview and helps with navigation
3	**Sensor data** screen	■ Displays data for the connected sensor
4	**Sensor output** screen	**Sensor output** screen
5	Status bar	■ Displays status information about the application
6	Configuration window	■ Contains the sensor-specific parameters that you can set
7	Toolbar	■ Contains icon buttons as an extension to the menu
8	Checkbox	■ Show image: activate or deactivate the image display ■ Show results: activate and deactivate the result display

9	Results area	- Displays result data from the sensor - A varying number of tabs can be displayed depending on which sensor is connected. - This field can be activate or deactivate with the point **Show results**
10	Image display	- Displays the images captured or stored in the error memory - This field can be activate or deactivate with the point **Show image**
11	Tab	Displays information about the current image and the pixel under the mouse pointer. The following items are displayed: - Image size - Zoom level - Mouse position in image coordinates - Current grayscale value - Image number

8.2 Menu bar

The menu bar contains a list of menu items. The individual functions depend on the type of sensor connected and the current authorization level and are, therefore, not always all visible.

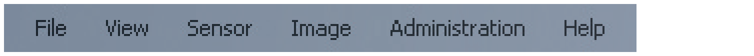
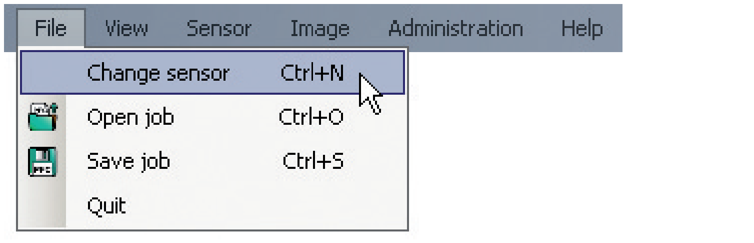


Figure 8.1 Menu bar

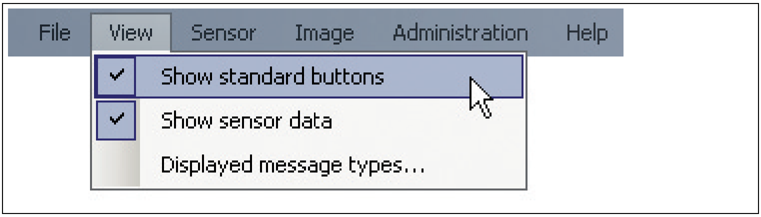
8.2.1 *File* menu



Change sensor	Disconnects the sensor 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 8.1 **File** menu

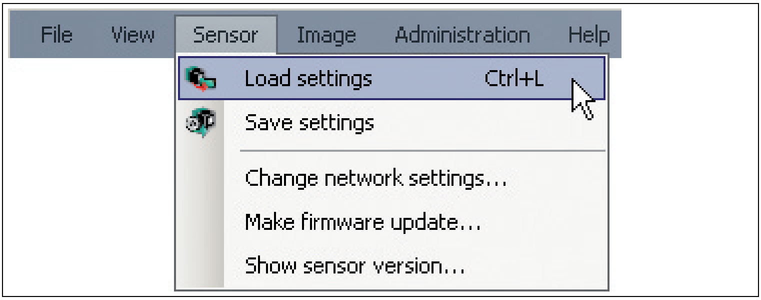
8.2.2 *View* menu



Show standard buttons	Toggles the display of buttons in the left bar on and off.
Show sensor data	Hides the display of sensor data in the top right of the screen.
Displayed message types…	This item enables you to specify which types of message are to be output by the sensor. Messages output for the selected types are displayed in the "Sensor output" column on the right. Info: Information is displayed Warning: Warnings are displayed Error: Errors are output Critical: Serious errors are output Assert: Internal errors are displayed

Table 8.2 **View** menu

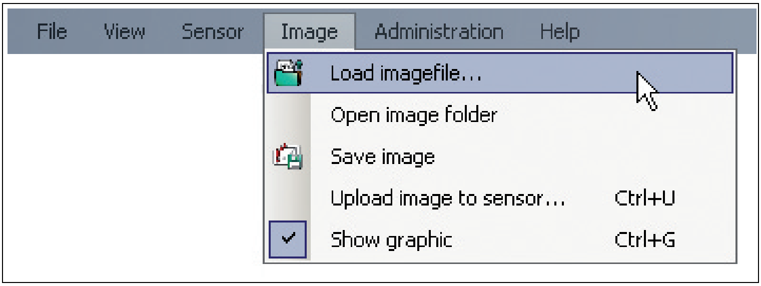
8.2.3 *Sensor* menu



Load settings...	Loads the saved settings from the sensor
Save settings...	Saves the settings to the sensor
Change network settings...	If the sensor is connected to the PC via Ethernet, this command enables you to change some of the connection settings on the sensor. Once you have changed them, the connection between Vision Configurator and the sensor is automatically disconnected. If required, change the network address to the newly allocated IP and reconnect Vision Configurator to the sensor. Press the **Connect** button to open the Login screen again.
Make firmware update...	Performs a firmware update. This command should be used by experienced users only.
Show sensor version...	Displays the sensor version number.

Table 8.3 **Sensor** menu

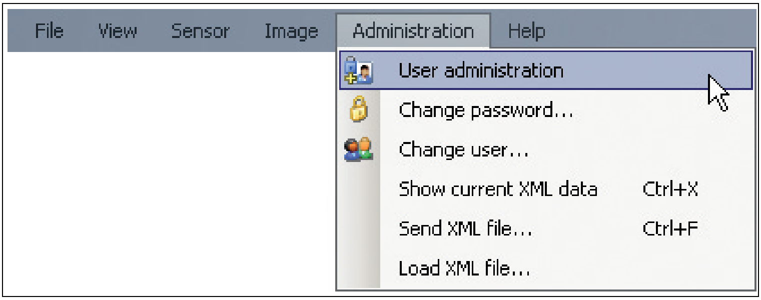
8.2.4 *Image* menu



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

Table 8.4 **Image** menu

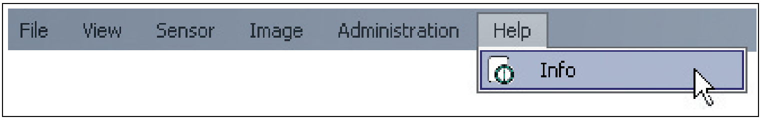
8.2.5 *Administration* menu



User administration	Opens a window in which all the currently created users at the same authorization level or lower are displayed. New users with 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.
Show current XML data	Loads the current XML data from the sensor and displays it in a separate window.
Send XML file...	Saves the XML data on a PC.
Load XML file...	Loads XML data from a PC.

Table 8.5 **Administration** menu

8.2.6 *Help* menu



Info	Displays information about Vision Configurator.

Table 8.6 **Help** menu

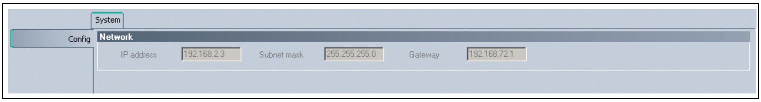
8.3 Configuration window

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.

8.3.1 System tab

All "Interface" menu options are not visible with all sensor versions.

System configuration window - Config menu item

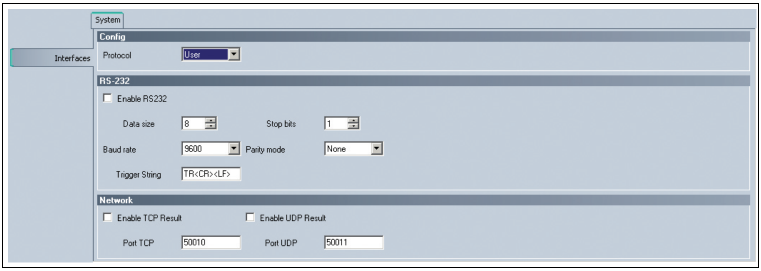


IP address 192.168.2.3 Subnet mask 255.255.255.0 Gateway 192.168.72.1

Network

IP address	Displays the sensor's IP address
Subnet mask	Displays the sensor's subnet mask
Gateway	Displays the sensor gateway

System configuration window - Interfaces menu item



Protocol User

□ Enable RS232

Data size 8 Stop bits 1

Baud rate 9600 Parity mode None

Trigger String TR<CR><LF>

□ Enable TCP Result □ Enable UDP Result

Port TCP 50010 Port UDP 50011

Figure 8.2 **System tab - Interfaces** configuration window

Config

Protocol	Data transfer User: Data is transmitted in binary form XML: Data is transmitted as XML

2012-10

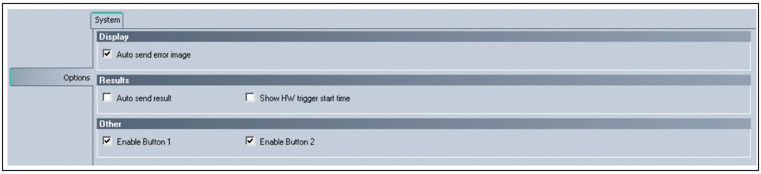
RS-232

Enable RS232	Activate or deactivate RS 232 transmission
Data size	Number of data bits. Adjustment range 5 ... 8
Stop bits	Number of stop bits. Adjustment range 1 ... 2
Baud rate	Select the baud rate. Adjustment options: 9600; 57600; 38400; 19200; 115200
Parity mode	Setting for detection of transmission errors. Adjustment options: Even; None; Odd
Trigger string	Specifies a character string. If this character string is sent using the RS 232 interface, a trigger is initiated.

Network

Enable Network Result	Transfer of results (code content) over the network interface (TCP/IP)
Port	Enter the corresponding port

System configuration window - Options menu item



☑ Auto send error image

☐ Auto send result ☐ Show HW trigger start time

☑ Enable Button 1 ☑ Enable Button 2

Figure 8.3 **System tab - Options** configuration window

Display

Auto send error image	Logging of errors/fault patterns

Results

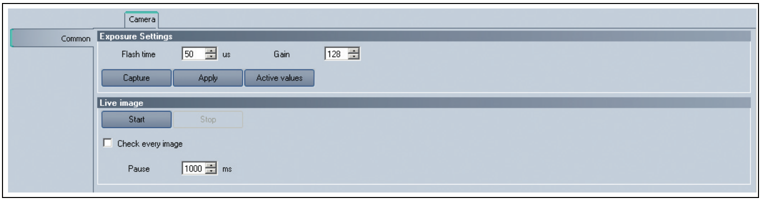
Auto send result	Transmission of result data via Ethernet or RS 232
Show HW trigger start time	Output of trigger start time in the output data

Other

Enable Button 1	Activates or deactivates button 1
Enable Button 2	Activates or deactivates button 2

8.3.2 Camera tab

Camera configuration window - Common menu item



Flash time 50 us Gain 128

□ Check every image

Pause 1000 ms

Figure 8.4 **Camera tab - Common** configuration window

Exposure settings

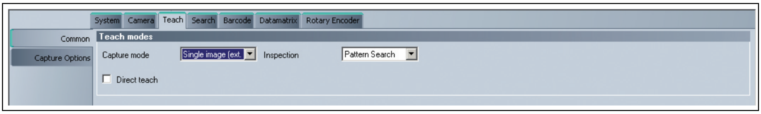
Flash time	Exposure time setting in µs
Gain	Gain setting
Capture button	Captures an image with the set flash time
Apply button	The currently set values are used for the next Teach-in process.
Active values button	Resets to the values from the last Teach-in process

Live image

Start	Starts the live image
Stop	Stops the live image
Check every image	Live image with result output
Pause	Specify the pause between two captured images in live image mode in ms

8.3.3 Teach tab

Teach configuration window - Common menu item



Capture mode Single image (ext. Inspection Pattern Search

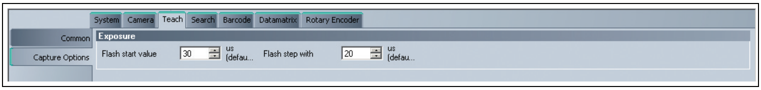
□ Direct teach

Figure 8.5 **Teach tab - Common** configuration window

Teach modes

Capture mode	See chapter 6.4.1
Inspection	See chapter 6.4.2
Direct teach	See chapter 6.4.2

Teach Tab configuration window - Capture Options menu item



Exposure

Flash start value 30 us (defau...

Flash step with 20 us (defau...

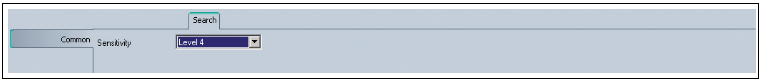
Figure 8.6 **Teach tab - Capture Options** configuration window

Exposure

Exposure time control settings	
Flash start value	Minimum exposure time in µs
Flash step width	In the Teach-in process, the flash time is adjusted automatically via incremental increases. This step width can be set here.

8.3.4 Search tab

Search configuration window - Common menu item



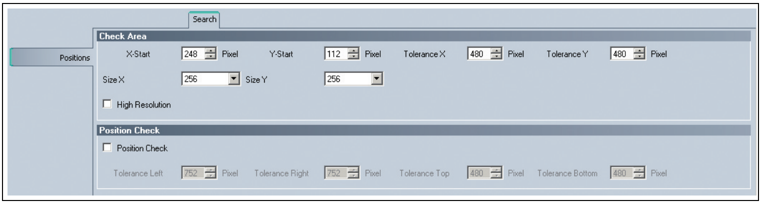
Sensitivity Level 4

Figure 8.7 **Search tab - Common** configuration window

Common

Sensitivity	Sensitivity adjustment High values lead to a high hit rate but also to an increase in incorrect rejections. For Calendar, even minimal differences are detected.

Search configuration window - Positions menu item



Check Area

X-Start 248 Pixel Y-Start 112 Pixel Tolerance X 480 Pixel Tolerance Y 480 Pixel

Size X 256 Size Y 256

□ High Resolution

Position Check

□ Position Check

Tolerance Left 752 Pixel Tolerance Right 752 Pixel Tolerance Top 480 Pixel Tolerance Bottom 480 Pixel

Figure 8.8 **Search tab - Positions** configuration window

Check area

X-Start	Specification of the read range. Input of the X start position
Y-Start	Specification of the read range. Input of the Y start position
Tolerance X	Input of the tolerance range in X direction
Tolerance Y	Input of the tolerance range in Y direction
Size X	Specification of the read range. Size in X direction
Size Y	Specification of the read range. Size in Y direction
High Resolution	Activates the search using high resolution

You can use the check position settings (Check Area) in image comparison mode to move the position of the area of the pattern image to be compared.

Note!

After every change, the pattern image must be taught again.

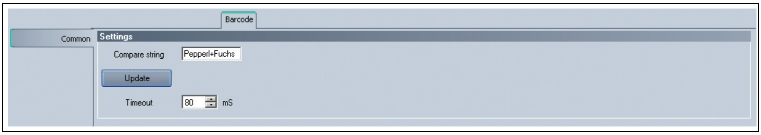
Position check

Position Check	Activates or deactivates the tolerance range check
Tolerance Left	Specification of the read range. Input of the X start position
Tolerance Right	Input of tolerance range to right
Tolerance Top	Input of tolerance range at top
Tolerance Bottom	Input of tolerance range at bottom

In addition to checking a taught pattern or barcode, you can also perform a position check. This checks whether the position of the pattern or barcode in the current image lies within the specified tolerance range around the taught position. If the position check is enabled, the tolerance range is shown as a blue frame in the current image and in the pattern. All positions are checked and specified in pixels.

8.3.5 Barcode tab

Barcode configuration window - Common menu item



Compare string Pepperl+Fuchs

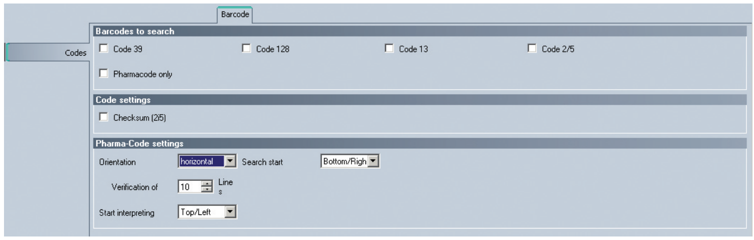
Timeout 80 mS

Figure 8.9 **Barcode tab - Common** configuration window

Settings

Compare string	The character string is compared with the barcode read. If the code is identical, "Good" is output
Update	Requests the reference barcode
Timeout	Maximum reading time; if this time is exceeded, a "No Read" is output

Barcode configuration window - Codes menu item



Barcodes to search

☐ Code 39 ☐ Code 128 ☐ Code 13 ☐ Code 2/5

☐ Pharmacode only

Code settings

☐ Checksum (2i5)

Pharma-Code settings

Orientation: horizontal
Search start: Bottom/Righ
Verification of: 10 Lines
Start interpreting: Top/Left

Figure 8.10 **Barcode tab - Codes** configuration window

Barcodes to search

Code 39	Select whether Code 39 is to be read
Code 128	Select whether Code 128 is to be read
Code 13	Select whether Code 13 is to be read
Code 2/5	Select whether Code 2/5 is to be read
Pharmacode only	Select whether only the Pharmacode is to be read

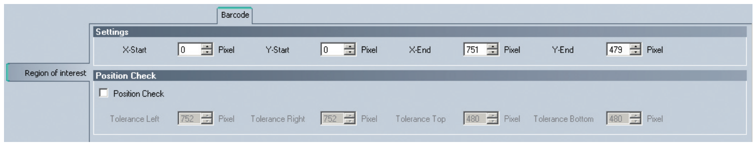
Code settings

Checksum (2i5)	Activates the checksum when reading Code 2/5 interleaved

Pharmacode settings

Orientation	Select either the ladder barcode (vertical) or picket fence barcode (horizontal)
Search start	Which corner of the image is closest to the barcode to be read
Verification of	The anticipated number of Pharmacode bars
Start interpreting	Barcode read direction

Barcode configuration window - Region of interest menu item



Settings

X-Start: 0 Pixel
Y-Start: 0 Pixel
X-End: 751 Pixel
Y-End: 479 Pixel

Position Check

☐ Position Check

Tolerance Left: 752 Pixel
Tolerance Right: 752 Pixel
Tolerance Top: 480 Pixel
Tolerance Bottom: 480 Pixel

Figure 8.11 **Barcode tab - Region of interest** configuration window

Settings

X-Start	Specification of the read range. Input of the X start position
Y-Start	Specification of the read range. Input of the Y start position
X-End	Specification of the read range. Input of the X end position
Y-End	Specification of the read range. Input of the Y end position

You can use the read range settings (Settings) to move the position of the read range, in which the barcode is located.

Note!

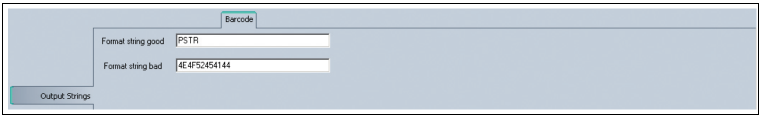
After every change, the barcode must be taught in again.

Position check

Position Check	Activates or deactivates the tolerance range check
Tolerance Left	Specification of the read range. Input of the X start position
Tolerance Right	Input of tolerance range to right
Tolerance Top	Input of tolerance range at top
Tolerance Bottom	Input of tolerance range at bottom

In addition to checking a taught pattern or barcode, you can also perform a position check (Position Check). This checks whether the position of the pattern or barcode in the current image lies within the specified tolerance range around the taught position. If the position check is enabled, the tolerance range is shown as a blue frame in the current image and in the pattern. All positions are checked and specified in pixels.

Barcode configuration window - Output Strings menu item



Format string good: PSTR

Format string bad: 4E4F52454144

Figure 8.12 **Barcode tab - Output Strings** configuration window

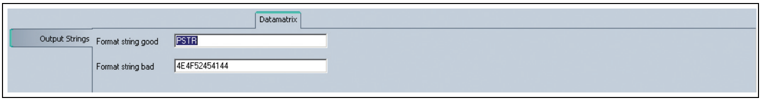
Output strings

Format string good	Output string for good read PSTR: Outputs the value read PSTR(0,5): Outputs the value read from position 0, 5 characters long PSTR(3,3): Outputs the value read from position 3, 3 characters long
Format string bad	Output string for bad read

8.3.6 Data Matrix tab

The Data Matrix tab is optional and is not available in all versions.

Data Matrix configuration window - Output Strings menu item



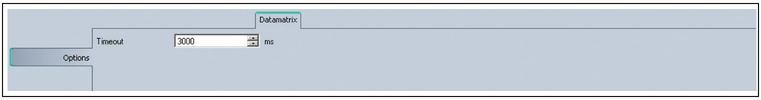
Format string good: PSTR
Format string bad: 4E4F52454144

Figure 8.13 **Data Matrix tab - Output Strings** configuration window

Output strings

Format string good	Output string for good reading PSTR: Outputs the value read PSTR(0.5): Outputs the value read from position 0, 5 characters long PSTR(3.3): Outputs the value read from position 3, 3 characters long
Format string bad	Output string for bad reading

Data Matrix configuration window - Options menu item



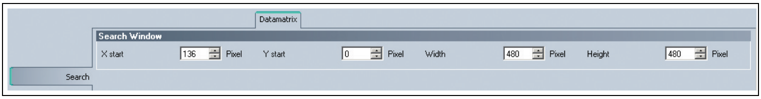
Timeout 3000 ms

Figure 8.14 **Data Matrix tab - Options** configuration window

Options

Timeout	Maximum read time; if this time is exceeded, a "No Read" is output

Data Matrix configuration window - Search menu item



Search Window
X start 136 Pixel Y start 0 Pixel Width 480 Pixel Height 480 Pixel

Figure 8.15 **Data Matrix tab - Search** configuration window

Search window

X start	X start position of read range
Y start	Y start position of read range
Width	Width of read range
Height	Height of read range

You can use the read range settings (Search window) to move the position of the area to be read, in which the data matrix code should be located.

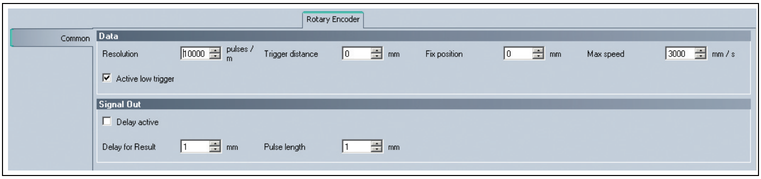
Note!

After every change, the data matrix code must be taught in again.

8.3.7 Rotary Encoder tab

On this tab, you can change the settings for the rotary encoder input.

Rotary Encoder configuration window - Common menu item



Data

Resolution 10000 pulses / m

Trigger distance 0 mm

Fix position 0 mm

Max speed 3000 mm / s

☑ Active low trigger

Signal Out

☐ Delay active

Delay for Result 1 mm

Pulse length 1 mm

Figure 8.16 **Rotary Encoder tab - Common** configuration window

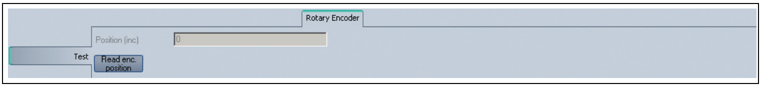
Data

Resolution	Rotary encoder resolution in pulses per meter
Trigger distance	Distance between the trigger sensor/edge of the sheet and the center of the sensor read field.
Fix position	Manual entry of the image capture position starting from the edge of the sheet
Max speed	Maximum speed of the conveyor
Active low trigger	Activate or deactivate triggers for a falling edge

Signal out

Delay active	Activates the adjustment option for the delay time
Delay for result	Delay time in mm between the trigger and the result output
Pulse length	Length of the output pulse for the result output in mm

Rotary Encoder configuration window - Test menu item



Position (inc) 0

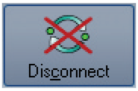
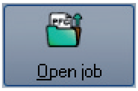
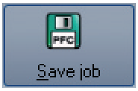
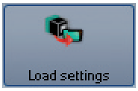
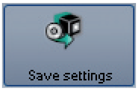
Figure 8.17 **Rotary Encoder tab - Test** configuration window

Test

Position (inc)	Displays the current rotary encoder position
Read enc. position	Reads the rotary encoder position

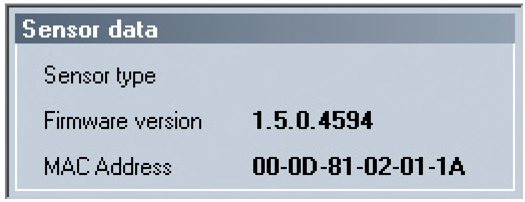
8.4 Toolbar

The toolbar contains various function icons.

	The connection between the PC and the sensor is disconnected
	Opens a setting saved on the hard disk
	Saves the settings made on the hard disk
	Discards all changes and loads the saved parameters from the sensor
	Permanently saves all data on the sensor
	Triggers the sensor. Depending on the sensor operating mode, it transmits a single measured value or continuous measured values
	Starts Teach-in mode to teach a reference image and then waits for trigger signals
	Displays the last image captured. For multi-trigger image captures, you can scroll through all images by pressing repeatedly.
	Displays the taught reference image
	Displays all images captured in the Teach-in process
	Displays the last fault pattern
	Displays previous fault patterns

8.5 Sensor Data Screen

The connected sensor type, the firmware version of the connected sensor, and the MAC address are displayed in the Sensor Data field.



Sensor data

Sensor type

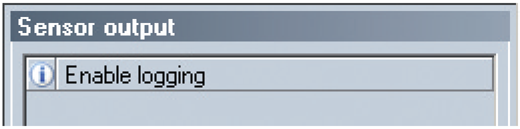
Firmware version 1.5.0.4594

MAC Address 00-0D-81-02-01-1A

Figure 8.18 Sensor data screen

8.6 Sensor output screen

Communication between the host PC and the connected sensor is displayed in the Sensor output field. You can select the messages to be displayed under the **View/Displayed message types...** menu item.

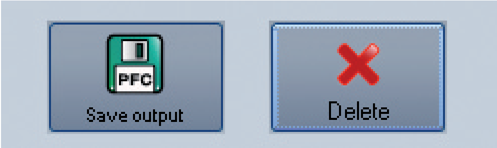


Sensor output

Enable logging

Figure 8.19 Sensor output screen

There are two buttons in the lower section of the Sensor output screen.



Save output	Saves the entire content of the window to a txt file
Delete	Deletes the entire content of the window

8.7 Image display

The image display can be used to display various images, from reference images and fault patterns to the current image. The **Show image** and **Show results** check boxes can be used to activate and deactivate the image display and result display.

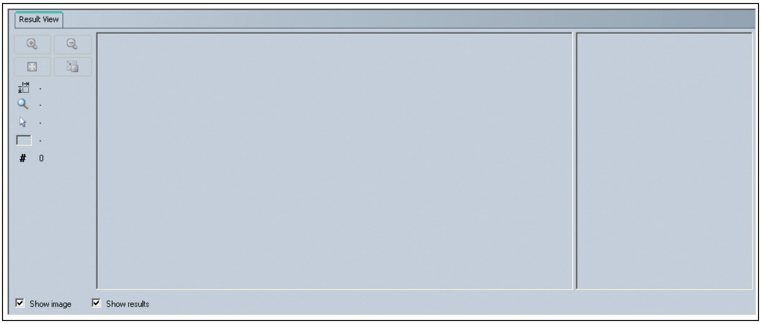


Figure 8.20 Result View screen

When you press the right mouse button or the context menu button, the following context menu appears:

Figure 8.21 Image View context menu screen

Context menu

Load image file...	Loads a sensor image. You can select the sensor image.
Open image folder	Opens the storage location
Save image	Saves the displayed sensor image

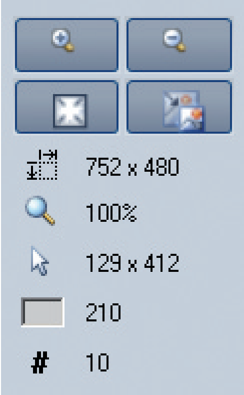


Figure 8.22 Image View toolbar screen

Magnifier +	Zooms in on sensor image
Magnifier -	Zooms out on sensor image
Fit to window	Fits the sensor image to the window
Original size	Set original image size
Size details	Specify the size of the sensor image
Zoom factor	Displays the zoom factor. Zoom factor 1 is original size
Position details	Specify the position of the mouse pointer
Grayscale value details	Grayscale value details for the pixel indicated by the mouse pointer
Image number	Specify the image number

Table 8.7 **Image View toolbar** menu

9 Maintenance and repair

9.1 Maintenance

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

Observe the following instructions when cleaning:

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

9.2 Repair

The devices must not be repaired, changed or manipulated. If there is a defect, the product must always be replaced with an original device.

10 Troubleshooting

10.1 What to do in the event of an error

Before requesting a service call, please check that the following actions have been taken:

- Test the equipment according to the following checklists,
- Telephone assistance from the Service Center in order to isolate the problem.

Checklist

Fault	Cause	Remedy
"PWR" LED not lit up	The power supply is switched off.	Check whether there is a reason why it is switched off (installation or maintenance work etc.). Switch the power supply on if appropriate.
"PWR" LED not lit up	Wiring fault in the splitter or control cabinet.	Check the wiring carefully and repair any wiring faults.
No connection to the device	Network cable not connected.	Connect the network cable.
No connection to the device	Wrong network cable used.	Direct connection between PC and device: Use a crossover network cable. Connection via an existing network: Use a twisted-pair network cable.
READY output does not switch to high	Hardware error	Check the rotary encoder.

- If none of the above remedies correct the problem, please contact the Service Center. Please have ready the fault patterns and version numbers of the sensor. The version number can be found at the bottom left of the operator interface.

11 Appendix

11.1 Technical data

General specifications

Light type	Integrated LED lightning (white)
Symbologies	2/5 interleaved, Code13, Code39, Code128, Pharmacode
Object size	25 mm x 25 mm
Read distance	55 mm
Depth of focus	± 5 mm
Reading field	65 mm x 40 mm
Evaluation frequency	10 Hz
Target velocity	triggered max. 4 m/s

Nominal ratings

Camera	
Type	CMOS , Global shutter
Number of pixels	752 x 480 pixels
Gray scale	256
Image recording	real-time , Program-controlled or triggered externally

Indicators/operating means

Operating display	LED green: Ready for operation
Controls	2 x Button
LED indicator	Trigger, Good, Bad, Teach, Diag1, Diag2, PWR

Electrical specifications

Operating voltage	24 V DC ± 15% , PELV
No-load supply current	max. 250 mA
Power consumption	6 W

Interface 1

Interface type	Ethernet
Protocol	TCP/IP
Transfer rate	100 MBit/s
Cable length	max. 30 m

Input

Input voltage	to be applied externally 24 V ± 15% PELV
Number/Type	1 trigger input 1 teach input 2 input (IN1, IN2)
Input current	approx. 2 mA at 24 V DC
Switching threshold	low: < 10 V, high: > 15 V
Cable length	max. 30 m

Output

Number/Type	Teach active, Good, Bad, Ready
Switching type	PNP , short-circuit/overload protected
Switching voltage	to be applied externally 24 V ± 15 % PELV
Switching current	max. 100 mA each output
Cable length	max. 30 m

Ambient conditions

Ambient temperature	0 ... 45 °C (32 ... 113 °F)
Storage temperature	-20 ... 60 °C (-4 ... 140 °F)
Relative humidity	80 % , noncondensing
Shock resistance	< 50 g
Vibration resistance	< 3 G , 11 ... 200 Hz

Mechanical specifications

Protection degree	IP65
Connection	8-pin, M12x1 connector, standard (supply+IO) , M12 x 1 female connector, 5-pin, standard (IO) , M12x1 socket, 4-pin, D-coded (LAN)
Material	
Housing	PC/ABS
Optical face	Plastic pane
Installation	4 x M6 threading
Mass	approx. 160 g

Compliance with standards and directives

Standard conformity	
Noise immunity	EN 61326-1
Emitted interference	EN 61000-6-4
Protection degree	EN 60529
Laser class	IEC 60825-1:2007

FACTORY AUTOMATION – SENSING YOUR NEEDS

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

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

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

www.pepperl-fuchs.com

PEPPERL+FUCHS
SENSING YOUR NEEDS

Subject to modifications

TDOCT2119D__ENG
10/2012