MANUAL

BIS510P-*
Vision Sensor for Sheet Verification

$C\epsilon$

PEPPERL+FUCHS

SENSING YOUR NEEDS

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"

PEPPERL+FUCHS

PEPPERL+FUCHS

2015-03

PEPPERL+FUCHS

1   Introduction

Informative Symbols

Note!

This symbol brings important information to your attention.

Action

This symbol indicates a paragraph with instructions.

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

PEPPERL+FUCHS

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**

PEPPERL+FUCHS

3   Safety

3.1   Symbols relevant to safety

STOP

Danger!
This symbol indicates an imminent danger.

Non-observance will result in personal injury or death.

⚠

Warning!
This symbol indicates a possible fault or danger.

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

△

Caution!
This symbol indicates a possible fault.

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

3.2   Intended Use

The sheet verification sensor is used for monitoring sheet sequences using imaging technology or barcode evaluation.

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.

PEPPERL+FUCHS

4   Product Description

4.1   Use and Application Areas

The sheet verification sensor 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. This occurs 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. The device can be operated locally, 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. Time-critical signals such as triggers and results are transmitted via the digital inputs and outputs. The network interface enables commands and settings to be sent to the sensor or images to be loaded.

2015-03

PEPPERL+FUCHS

4.2 Indicators and Operating Controls

The illumination unit has 7 LED indicators that provide information on the various device statuses.

1	LED DIAG2	Gelb
2	LED DIAG1	Gelb
3	LED POWER	Grün
4	LED TEACH	Gelb
5	LED BAD	Gelb
6	LED GOOD	Gelb
7	LED TRIGGER	Gelb
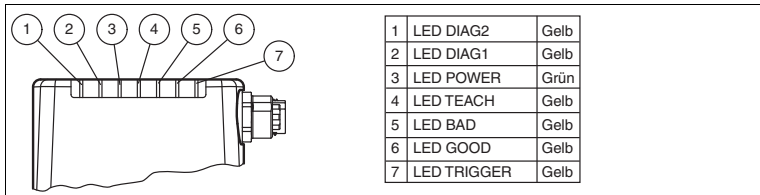
Figure 4.1   Displays and controls

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.*

Two buttons are located on the back of the sensor.
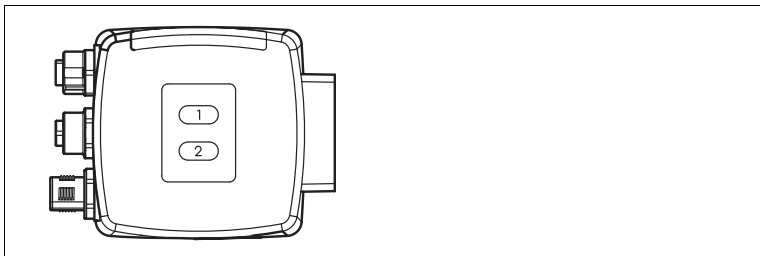
1   Scene illumination
Scene illumination lights up the sensor's field of vision, allowing easy positioning of the sensor during mounting.

2   Teach-in
During the Teach-in process, a reference image is captured and used for an image comparison or a code comparison.

PEPPERL+FUCHS

4.3 Interfaces and connections

The device includes the following connections:
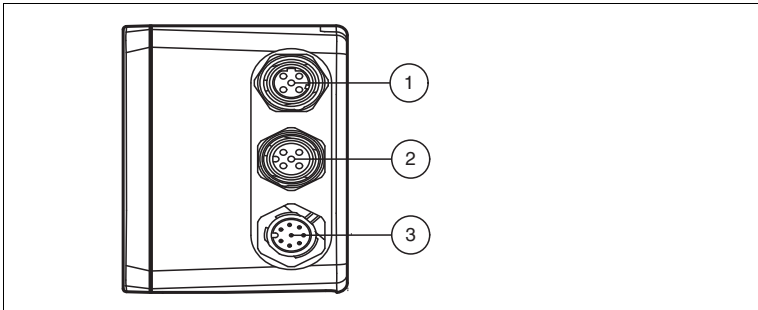
Figure 4.2   Device connections

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:
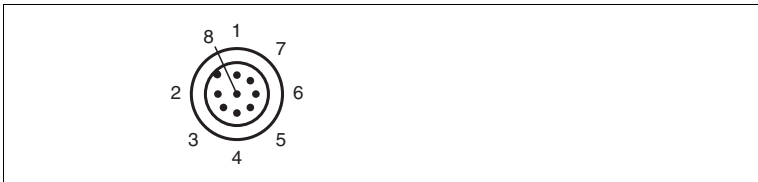
Figure 4.3   Connection layout for operating voltage and inputs and outputs

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

PEPPERL+FUCHS

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:
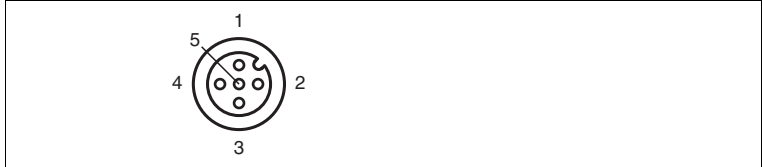
Figure 4.4   IO input connection layout

1   24 V power supply

2   IN1

3   Ground (GND)

4   IN2

5   Not used
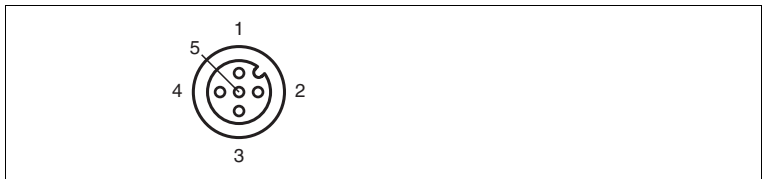
Figure 4.5   Encoder input connection layout

1   24 V power supply

2   B

3   Ground (GND)

4   A

5   Not used

PEPPERL+FUCHS

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:
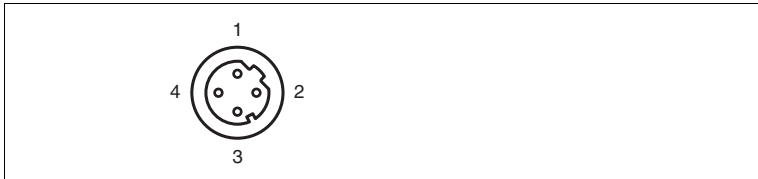
Figure 4.6   Network connection layout

1   TX+ Ethernet

2   RX+ Ethernet

3   TX- Ethernet

4   RX- Ethernet

4.4   Scope of Delivery

- BIS510P-*

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.

PEPPERL+FUCHS

4.5.2 Network Cable

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

Designation	Description
V45-G	RJ45 network plug, field attachable
V1S-G	M12 plug, 4-pin, field attachable
V1SD-G-2M-PUR-ABG-V45X-G	Double-ended cordset, RJ45 network plug with M12 plug, crossed, 4-pin
V1SD-G-2M-PUR-ABG-V45-G	Double-ended cordset, RJ45 network plug 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

PEPPERL+FUCHS

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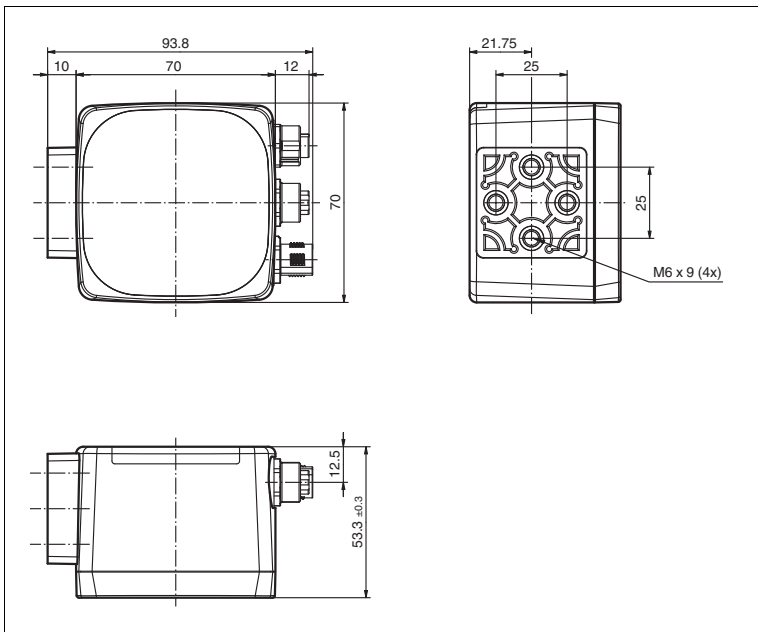
The vision sensor is mounted at a specific operating distance above the surface of the sheet.

PEPPERL+FUCHS

The following illustration shows an example of correct mounting:
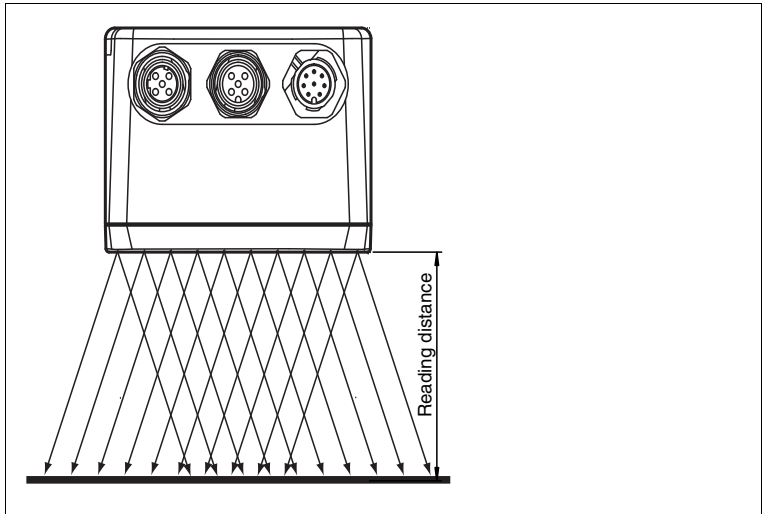
Figure 5.1   Installation example

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.

PEPPERL+FUCHS

▶ 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 sensor has been changed, you can reset the IP address to the factory default settings.

1. Connect the Vision Sensor to the power supply.

   ↳ The Vision Sensor powers up.

2. Wait until the LEDs flash.

3. Hold down buttons 1 and 2 on the back of the sensor at the same time for approx. two seconds.

   ↳ The LED flash sequence changes.
   The Vision Sensor powers up again and now uses the factory 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.

PEPPERL+FUCHS

6 Commissioning

6.1 Setting Up a Network Connection

Configuring the IP Address (Windows 7)

1. Select **Start > Control Panel**.
2. Select **Network and Sharing Center**.
3. Select **Change adapter settings** in the Network and Sharing Center view.
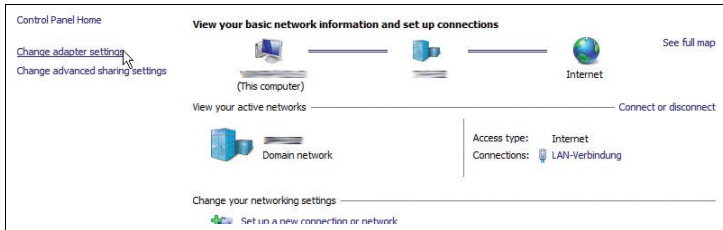
4. Double-click on the required connection.

   ↳ The properties window for the selected connection appears.

PEPPERL+FUCHS

17
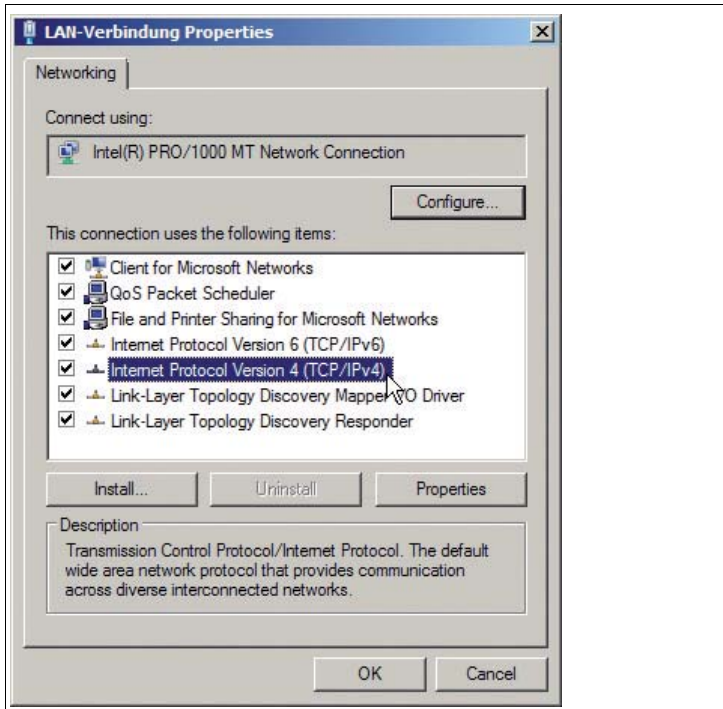
5. Double-click on **Internet Protocol Version 4 (TCP/IPv4)**.

↳ The properties window for the TCP/IP protocol appears.
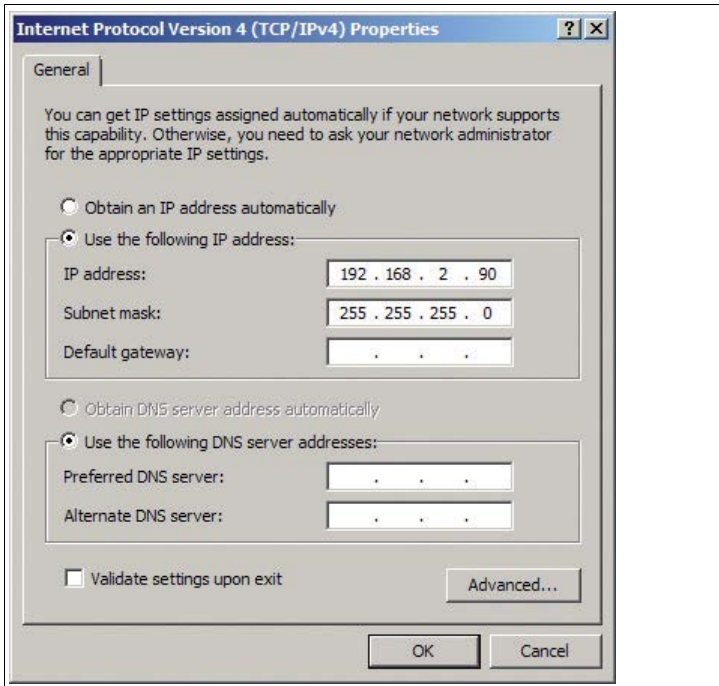
6. Select **Use the following IP address**.

7. Enter an IP address where the first three segments are identical to those of the IP address for the sensor, e.g., `192.168.002.090`. The last segment must be different to the IP address of the sensor.

8. Enter 255.255.255.0 as the subnet mask.

9. Click **OK** and **Close**.

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

2015-03

18

PEPPERL+FUCHS

6.2 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.
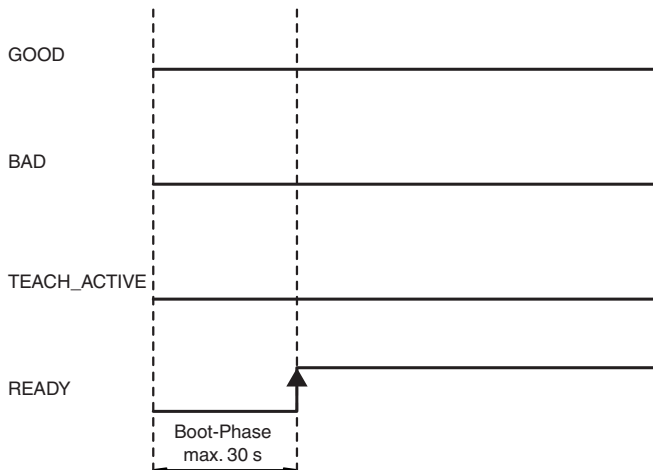
6.3 Functionality

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

6.3.1 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.

PEPPERL+FUCHS

6.3.2   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:
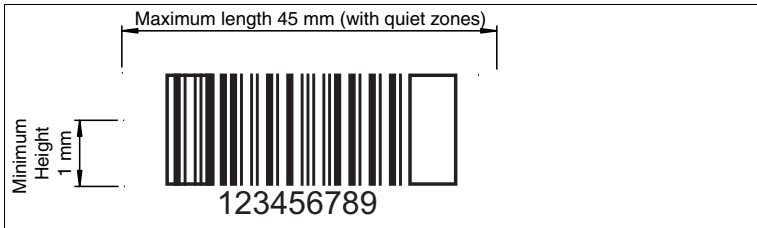
- Minimum bar width: 0.3 mm

The following barcode types can be read:

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

PEPPERL+FUCHS

6.4 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.

![Pepperl+Fuchs logo] **PEPPERL+FUCHS**

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.

6.5 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 BIS510P is usually passive, i.e., it never sends data automatically and responds only to requests. When active mode is activated, the sensor sends the fault pattern automatically whenever faults are detected.

Fault Pattern Memory

The BIS510P 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.

PEPPERL+FUCHS

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.
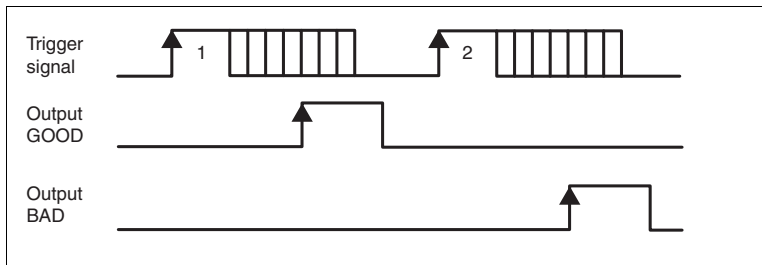
Figure 7.1   Signals in "Inspection" operating state

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.

PEPPERL+FUCHS

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.
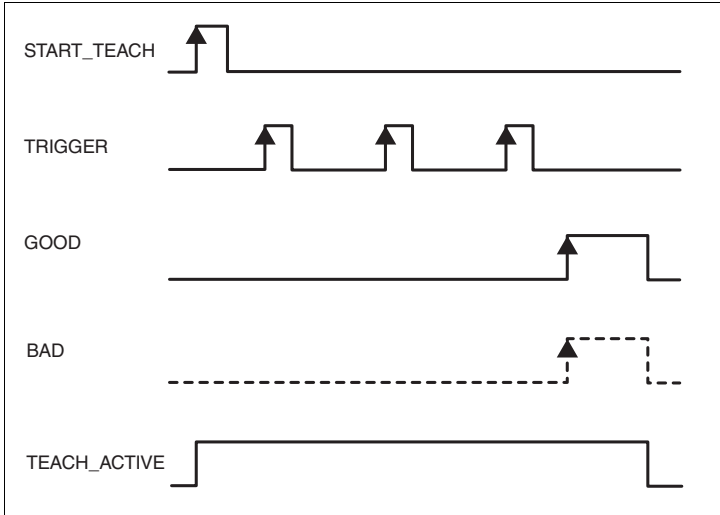
Figure 7.2   Signals in "Teach-In" operating state

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.

PEPPERL+FUCHS

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.
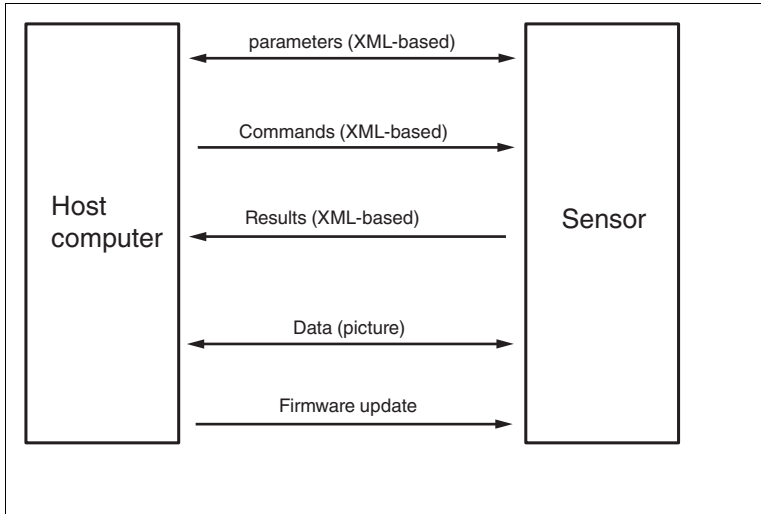
Figure 7.3   Network interface XML-Stings

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)

PEPPERL+FUCHS

7.4 Software Interface

A **.NET 2.0-based** software interface is provided for easy integration with PC software. This software interface is provided in the form of DLLs and handles the communication with the sensor. Integrate the DLLs into the programming environment (⟶ see Figure 7.4 on page 26) and run the programming lines indicated below.

All the examples relate to the Visual Studio 2010 programming environment and to the C# programming language.
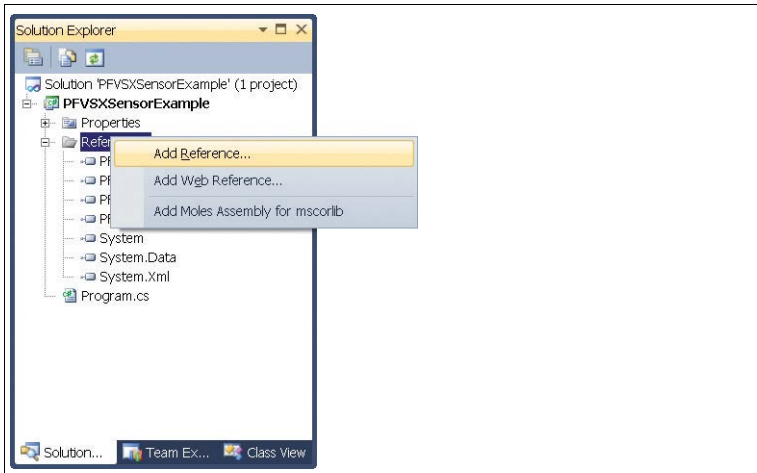
Figure 7.4   Integration interface DLL

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);
    sensor.SetSpecificSingleParameter(1, "Command",
"TriggerStart", "1");
    System.Threading.Thread.Sleep(1000);
    sensor.Disconnect();
  }
}
```

PEPPERL+FUCHS

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 in accordance with 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 Parameter Data

Received parameter data is stored in the sensor in a list. 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 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.

2015-03

PEPPERL+FUCHS

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

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

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

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

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

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

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

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

2015-03

PEPPERL+FUCHS

```
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 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 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

PEPPERL+FUCHS

```
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 results data can be retrieved with `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.

PEPPERL+FUCHS

```
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.

PEPPERL+FUCHS

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
SearchParams	Sensitivity	0 ... 5	Pattern comparison sensitivity Level 0 -> weak Level 5 -> strong

PEPPERL+FUCHS

Config ID	Parameter ID	Value	Description
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
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)

PEPPERL+FUCHS

Config ID	Parameter ID	Value	Description
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

PEPPERL+FUCHS

Config ID	Parameter ID	Value	Description
RotationEncoder	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.

2015-03

PEPPERL+FUCHS

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

PEPPERL+FUCHS

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
ElementShapeText	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.
ElementShapeRectangle	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

PEPPERL+FUCHS

8 Vision Configurator Software

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

The Vision Configurator software makes it easy to operate the sensor with its user-friendly interface. Standard functions include making connections to the sensor, specifying the operating parameters, saving data sets, and displaying data and error diagnostics.

Note!

The following user roles are already predefined with different permissions in the Vision Configurator.

User rights and password

User rights	Description	Password
Default	View all information Create users at same or lower level	A password is not required
User	View all information Sensor configuration Create users at same or lower level	User
Admin **PFAdmin**	View all information Sensor configuration Create and delete users	Request the admin password from Pepperl+Fuchs

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 machine control system using the TCP/IP protocol. To ensure proper communication, you must record all the changes made to the network configuration.

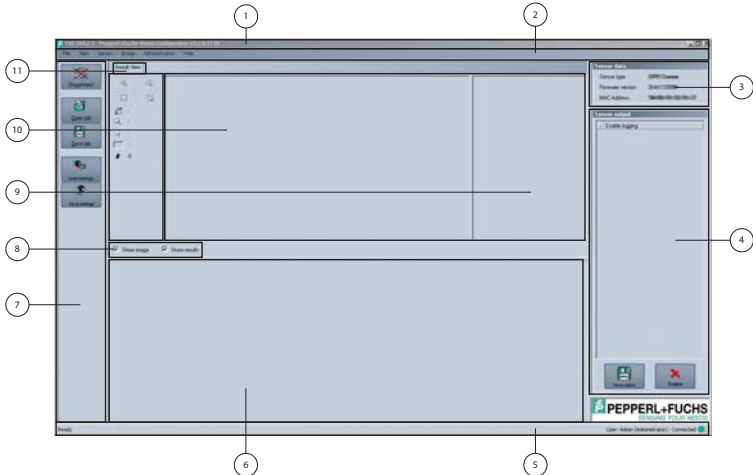An up-to-date description of the Vision Configurator software can be found at http://www.pepperl-fuchs.com.

PEPPERL+FUCHS

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

PEPPERL+FUCHS

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 functionality depends on the type of sensor that is connected and the permissions of the user logged in.
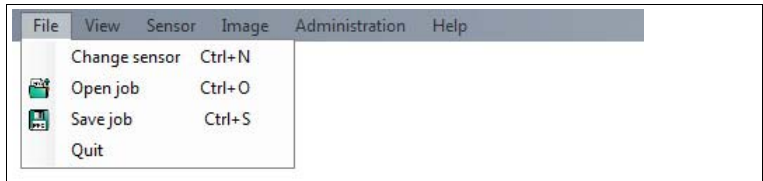

Figure 8.1   Menu Bar

8.2.1 *File* Menu

Figure 8.2   **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

PEPPERL+FUCHS

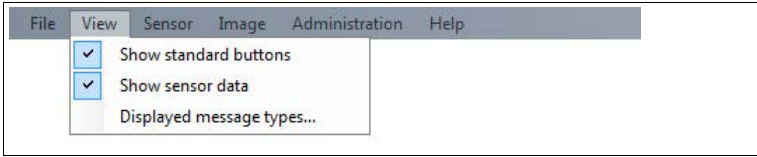## 8.2.2 *View* Menu

Figure 8.3 **View** menu

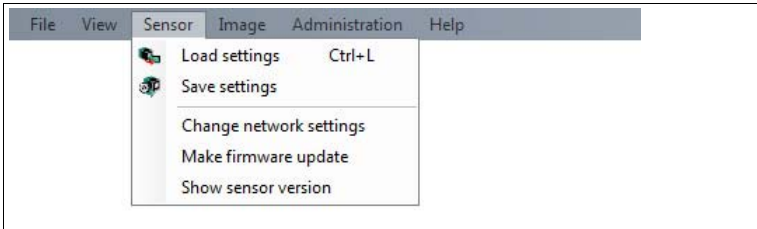| Show standard buttons	Toggles the display of the buttons in the bar on the left on and off.
Show sensor data	Hides the output of the sensor data in the top right of the screen.
Displayed message types…	This item enables you to specify which types of message are to be output by the sensor. Messages of the selected types output are displayed in the right-hand column "Sensor output." 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

Figure 8.4 **Sensor** menu

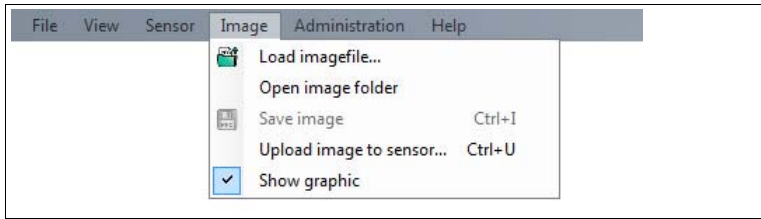| Load settings...	Loads the saved settings from the sensor
Save settings...	Saves the settings to the sensor
Change network settings...	If the sensor is connected to the PC via Ethernet, this command enables you to change some of the connection settings on the sensor. Once you have changed them, the 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 only be used by experienced users.
Show sensor version...	Displays the sensor version number.

Table 8.3 **Sensor** menu

2015-03

PEPPERL+FUCHS

8.2.4   *Image* Menu

Figure 8.5   **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

PEPPERL+FUCHS

8.2.5   *Administration* Menu

Figure 8.6   **Administration** menu

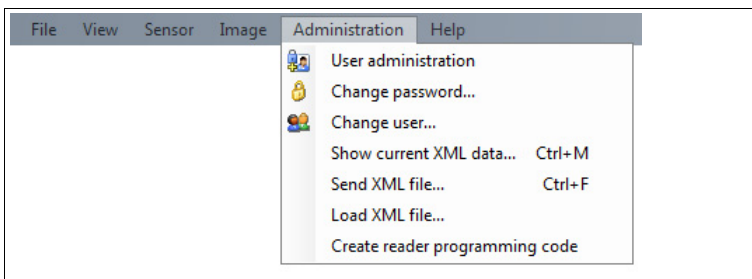|	
User administration	Opens a window in which all the currently created users at the same authorization level or lower are displayed. New users with the same authorization level or lower can also be created and deleted here. In addition, a user password can be reset to the default password for the relevant 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.
Create reader programming code	Enables the user to create control codes that can assign the sensor a device name and an IP address.

Table 8.5   **Administration** Menu
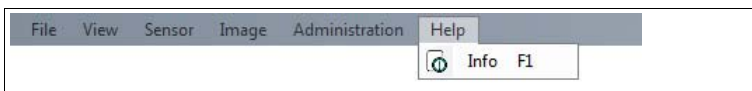
8.2.6   *Help* Menu

Figure 8.7   **Help** menu

Info	Displays information about Vision Configurator.

Table 8.6   **Help** menu

PEPPERL+FUCHS

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

Network

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

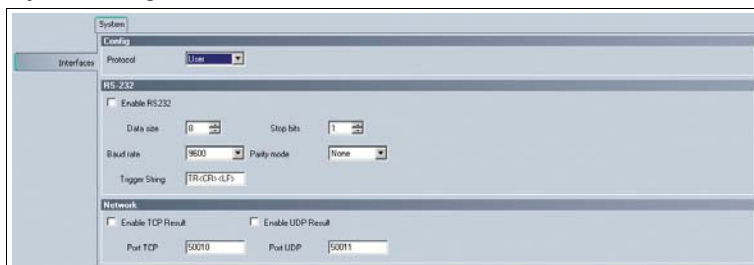**System configuration window - Interfaces menu item**

Figure 8.8   **System tab - Interfaces** configuration window

Config

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

RS-232

Enable RS-232	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

PEPPERL+FUCHS

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

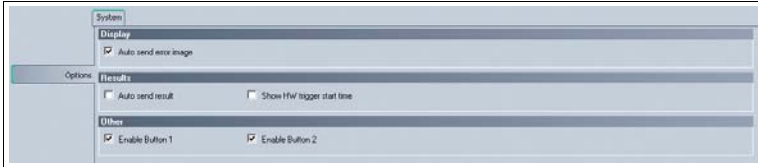| Enable Network Result	Transfer of results (code content) over the network interface (TCP/IP)
Port	Enter the corresponding port

System configuration window - Options menu item

Figure 8.9   **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

PEPPERL+FUCHS

8.3.2 Camera tab

Camera configuration window - Common menu item
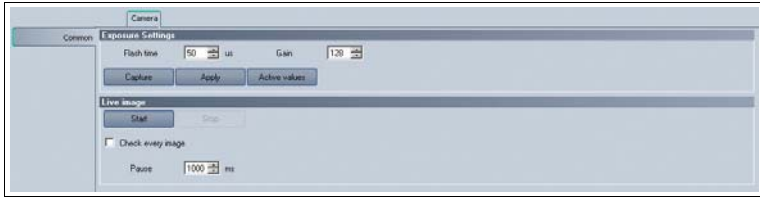
Figure 8.10   **Camera tab - Common** configuration window

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

Figure 8.11   **Teach tab - Common** configuration window

Teach Modes

Capture mode	See chapter 6.3.1
Inspection	See chapter 6.3.2
Direct teach	See chapter 6.3.2

![PEPPERL+FUCHS]

Teach Tab Configuration Window Capture Options Menu Item

Figure 8.12   **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

Figure 8.13   **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

Figure 8.14   **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

PEPPERL+FUCHS

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
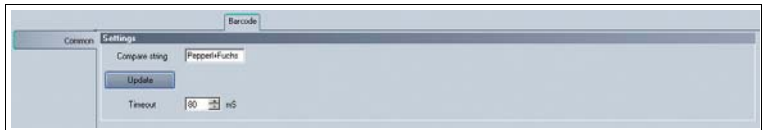
Barcode configuration window - Common menu item

Figure 8.15 **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

PEPPERL+FUCHS

Barcode configuration window - Codes menu item

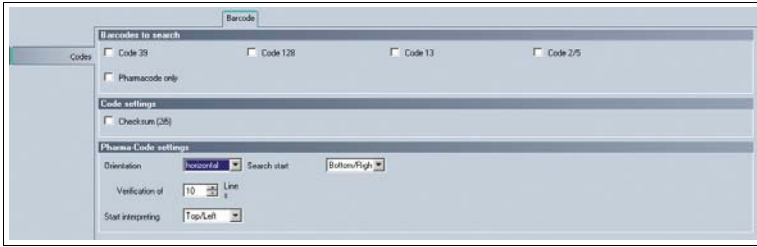
Figure 8.16   **Barcode tab - Codes** configuration window

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
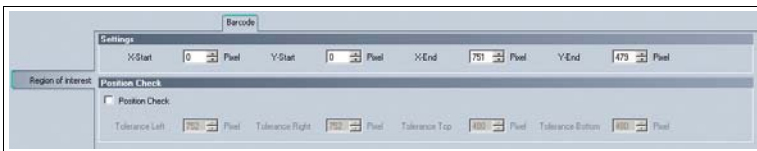

Figure 8.17   **Barcode tab - Region of interest** configuration window

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

2015-03

PEPPERL+FUCHS

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.

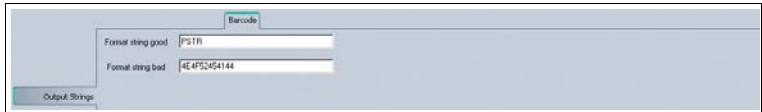**Barcode configuration window - Output Strings menu item**

Figure 8.18   **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

PEPPERL+FUCHS

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

Figure 8.19   **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
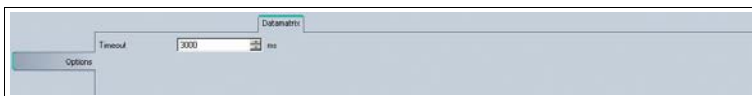

Figure 8.20   **Data Matrix tab - Options** configuration window

Options

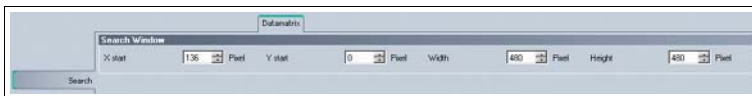| Timeout	Maximum read time; if this time is exceeded, a "No Read" is output

Data Matrix configuration window - Search menu item

Figure 8.21   **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.

PEPPERL+FUCHS

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.

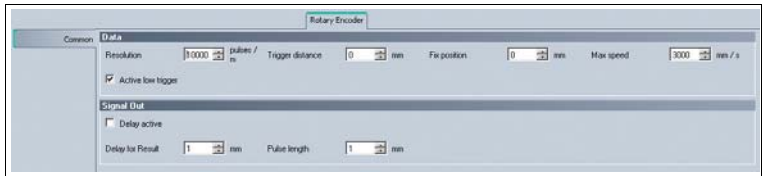**Rotary Encoder configuration window - Common menu item**

Figure 8.22　　**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
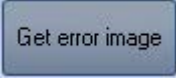

Figure 8.23　　**Rotary Encoder tab - Test** configuration window

Test

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

PEPPERL+FUCHS

8.4 Toolbar

The toolbar contains various function icons.

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

PEPPERL+FUCHS

Get error image	Displays the last fault pattern
Get next error image	Displays previous fault patterns

8.5 Sensor Data

This area shows the type of sensor connected, the firmware version of the connected sensor, and the MAC address.

Figure 8.24   Sensor data

8.6 Sensor Output

This area shows the communication between the control system and the connected sensor. To select which messages are to be displayed, select **View > Displayed message types**.
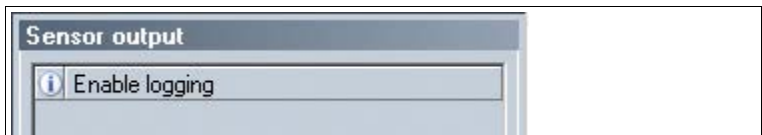

Figure 8.25   Sensor output

Two buttons are located in the lower area.

Save output	Saves the content of the window to a text file.
Delete	Deletes the contents of the window.

PEPPERL+FUCHS

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 8.26   Result View screen
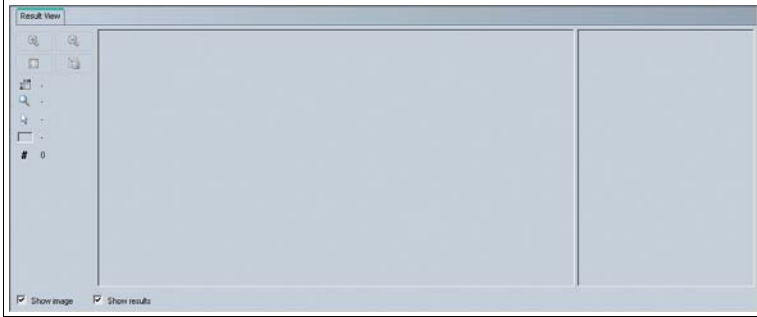
When you press the right mouse button or the context menu button, the following context menu appears:

Figure 8.27   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

PEPPERL+FUCHS
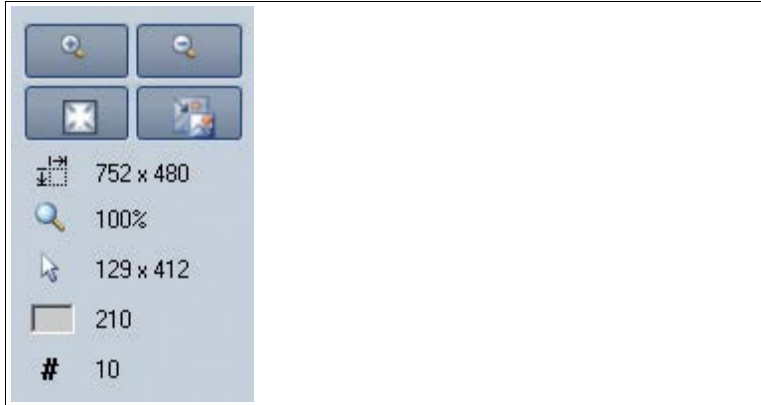
Figure 8.28   Image View toolbar screen

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

PEPPERL+FUCHS

57

9 Maintenance and Repair

9.1 Maintenance

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.

PEPPERL+FUCHS

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.

PEPPERL+FUCHS

11 Appendix

11.1 Technical Data

General data

Light type	Integrated LED flash (white)
Symbologies	2/5 interleaved, Code13, Code39, Code128, Pharmacode
Object size	25 mm x 25 mm
Read distance	51 mm
Depth of focus	± 5 mm
Read field	65 mm x 40 mm
Evaluation frequency	10 Hz
Object speed	Triggered max. 4 m/s

Nominal ratings

Imager	
Type	CMOS, global shutter
Number of pixels	752 x 480 pixels
Grayscales	256
Image capture	No delay, program-controlled or triggered externally

Indicators/operating controls

Operating status indicator	LED green: Ready for operation
Operating controls	2x buttons
LED display	Trigger, Good, Bad, Teach, Diag1, Diag2, PWR

Electrical data

Operating voltage	24 VDC ± 15 %, PELV
No-load 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

PEPPERL+FUCHS

Input

Input voltage	External supply 24 V ± 15 % PELV
Number/type	1 trigger input 1 teach input 2 inputs (encoderA, encoderB)
Input current	Approx. 2 mA at 24 VDC
Switching threshold	Low: < 10 V, high: > 15 V
Cable length	Max. 30 m

Output

Number/type	Good, Bad, Teach Active, Ready, OUT1
Switching mode	PNP, short-circuit and overload protection
Switching voltage	External supply 24 V ± 15 % PELV
Switching current	Max. 100 mA per output
Cable length	Max. 30 m

Ambient conditions

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

Mechanical data

Degree of protection	IP67
Connection	M12x1 plug, 8-pin, standard (power supply+IO), M12x1 socket, 5-pin, standard (IO), M12x1 socket, 4-pin, D-coded (LAN)
Material	
Housing	PC/ABS
Optical face	Plastic lens
Mounting	4 x M6 thread
Weight	Approx. 160 g

Conformity with standards and directives

Conformity with standards	
Immunity	EN 61326-1
Emitted interference	EN 61000-6-4
Degree of protection	EN 60529
Laser class	IEC 60825-1:2007

PEPPERL+FUCHS

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