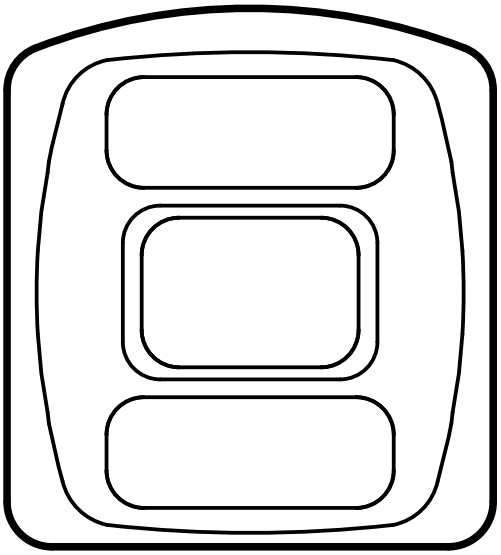
Operating instructions
3D sensor

O3D300
O3D302
O3D310
O3D312

UK



11533585 / 00 04 / 2023

Contents

UK

1. Preliminary note

This document is intended for specialists. These specialists are people who are qualified by their appropriate training and their experience to see risks and to avoid possible hazards that may be caused during operation or maintenance of the device. The document contains information about the correct handling of the device.

Read this document before use to familiarise yourself with operating conditions, installation and operation. Keep this document during the entire duration of use of the device.

1.1 Symbols used

► Instructions

> Reaction, result

[…] Designation of keys, buttons or indications

→ Cross-reference

Important note
Non-compliance may result in malfunction or interference.

Information
Supplementary note

1.2 Warnings used

NOTE
Warning of damage to property.

1.3 Open source information

This product can contain Free Software or Open Source Software from various software developers which is subject to the following licenses: General Public License version 1, version 2 and version 3 (General Public License version 3 in conjunction with the GNU Compiler Collection Runtime Library Exception version 3.1), Lesser General Public License version 2.1, Lesser General Public License version 3, Berkeley Software Distribution (BSD-2-Clause, BSD-3-Clause, BSD-4-Clause), MIT-License (MIT), Python Software Foundation License 2.0, Pearl Artistic License and Artistic License 2.0, Microsoft Public License, Apache Software License Version 1.0, 1.1 und 2.0, ISC License, libpng License, zlib License, the Academic Free License version 2.1, Mozilla Public License 2.0.

For the components subject to the General Public License in their respective versions the following applies:

This program is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation. If version 1 applies to the software: either version 1 of the License or (at your option) any later version; if version 2 (or 2.1) applies to the software: either version 2 (or 2.1) of the License or (at your option) any later version; if version 3 applies to the software: either version 3 of the License or (at your option) any later version. The following disclaimer of the software developers applies to the software components that are subject to the General Public License or the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU General Public License and the GNU Lesser General Public License for more details.

The responsibility of ifm electronic gmbh for ifm products, in the case of product-specific software, remains unaffected by the above disclaimer. Please note that the firmware for the ifm products is in some cases provided free of charge. The price of the ifm products has then to be paid for the respective device itself (hardware) and not for the firmware. For the latest information on the license agreement for your product please visit www.ifm.com

For binaries that are licensed under any version of the GNU General Public License (GPL) or the GNU LGPL or Mozilla Public License 2.0 you may obtain the complete corresponding source code of the GPL software from us by sending a written request to: opensource@ifm.com or to ifm electronic gmbh, Friedrichstraße 1, 45128 Essen, Germany.

We charge €30 for each request. Please write “source for product Y” in the memo line of your payment. Your request should include (i) the name of the covered binary, (ii) the name and the version number of the ifm product, (iii) your name and (iv) your return address.

The following applies to components covered by the General Public License in their respective versions. This offer is valid to anyone in receipt of this information. This offer is valid for at least three years (from the date you received the GPL/LGPL covered code).

2. Safety instructions

2.1 General

These instructions are an integral part of the device. They contain texts and figures concerning the correct handling of the device and must be read before installation or use.

Observe the operating instructions. Non-observance of the instructions, operation which is not in accordance with use as prescribed below, wrong installation or incorrect handling can seriously affect the safety of operators and machinery.

2.2 Target group

These instructions are intended for authorised persons according to the EMC and low-voltage directives. The device must be installed, connected and put into operation by a qualified electrician.

2.3 Electrical connection

Disconnect the device externally before handling it.

The connection pins may only be supplied with the signals indicated in the technical data and on the device label and only the approved accessories of ifm may be connected.

2.4 Tampering with the device

In case of malfunctions or uncertainties please contact the manufacturer. Any tampering with the device can seriously affect the safety of operators and machinery. This is not permitted and leads to the exclusion of any liability and warranty claims.

3. Functions and features

The O3D3xx 3D sensor is a photoelectric sensor measuring the distance between the sensor and the nearest surface point by point using the time-of-flight principle. The O3D3xx 3D sensor illuminates the scene with an infrared light source and calculates the distance by means of the light reflected from the surface.

From the image data, process values are generated via internal image processing and compared to threshold values. The comparative and process values are linked to the digital outputs. This allows to solve the following applications:

- Completeness monitoring
- Level measurement
- Distance monitoring
- Dimensioning of rectangular objects
- Sorting of rectangular objects

The measured data and process values can be provided via Ethernet and evaluated by the user. Parameter setting of the O3D3xx 3D sensor is also done via Ethernet.

The O3D3xx 3D sensor may only be used under the operating conditions specified in the data sheet.

The device safety is rated for use under the following environmental conditions:

- Indoor use
- Altitudes up to 2000 m
- Relative air humidity up to max. 90%, non condensing
- Pollution degree 3

Because of the requirements for electromagnetic interference emissions, the device is intended for use in industrial environments. The device is not designed for use in domestic areas.

The device may only be used under the operating conditions specified in the data sheet.

4. Items supplied

- O3D3xx 3D sensor
- Brief instructions

The data sheet and other documentation (software manual, etc.) are available on our website: www.ifm.com

5. Accessories

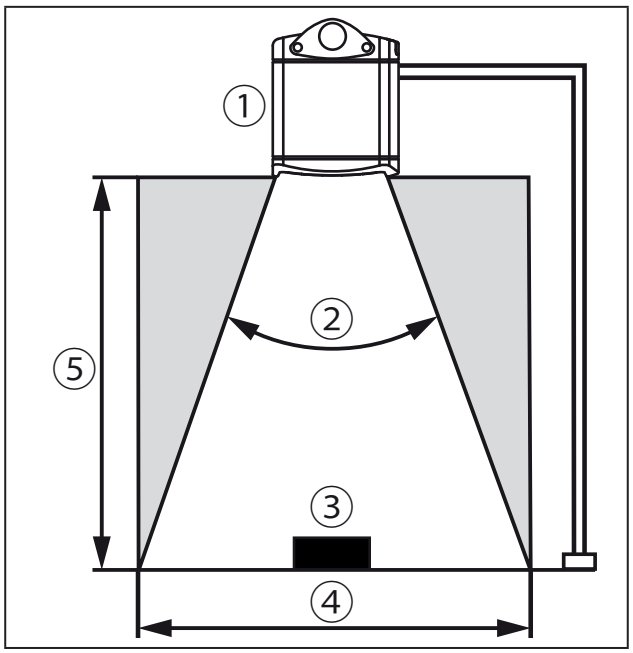
The following accessories are needed for the operation of the device:

Article number	Description
E11950	Power supply cable for camera/sensor
E11898	M12 industrial Ethernet connection cable

The ifm Vision Assistant software is available free of charge on our website: www.ifm.com

6. Installation

The chapter describes what has to be observed before installation and how to install the sensor.



① Device
② Angle of aperture
③ Object
④ Field of view
⑤ Distance between device and object

6.1 Select installation location

Observe the following instructions for the selection of the installation location:

► The object ③ must be completely in the field of view ④.

> The size of the field of view depends on the sensor type and is indicated in the data sheet. The size of the field of view also depends on the distance of the sensor to the object ⑤: With increasing distance the field of view becomes larger.

► Take tolerances into account when positioning the object.

► When determining the distance between sensor and object ⑤ take the measuring range of the sensor into account.

> The measuring range is indicated in the data sheet of the sensor.

► Select a distance as small as possible between sensor and object ⑤.

> If the distance is as small as possible, the object is detected with the maximum resolution.

► Avoid any strong ambient light and sunlight at the installation location.

> An extraneous light level of over 8 klx (with solar spectrum) causes measurement errors. In fact, only the infrared component between 800 and 900 nm is of concern.

► Avoid installation in heavily polluted environments.

> In heavily polluted environments the sensor lens will get dirty despite downwards orientation ①.

► Avoid transparent panes between the sensor ① and the object ③.

> Transparent panes reflect part of the light even if a very clean glass pane is used.

If the instructions are not observed, measurement errors may occur.

6.2 Additional sensor installation guidance

The surface temperature of the sensor depends on the operating mode, the parameter selection and the thermal exposure of the sensor to the environment.

Make sure that the sensor complies with the following requirement:

The surface temperature for easily accessible surfaces may be max. 25 °C higher than the ambient temperature (to IEC 61010-2-201).

The following diagrams contain typical warning limits as a reference for the installer.

The diagrams are valid for the following operating modes:

- Low [1 exposure]
- Moderate [2 exposures]
- High [3 exposures]

In the event of moderate and high exposures the typical warning limits must be determined via the sum of the exposure times. The exposure times are indicated in the software ifm Vision Assistant.

Follow one of the instructions if the warning limits are exceeded:

► Reduce surface temperature (→ 6.2.3).

► Mount the sensor in a location or housing that provides protection from the heat source but maintains air circulation around the sensor.

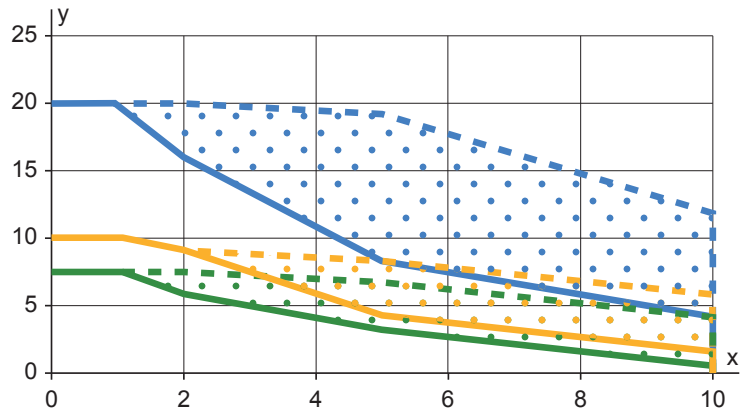
> An increase in the surface temperature of the sensor should be prevented.

The parameter "Max. background distance" is set in the ifm Vision Assistant. In the diagrams the warning limits of the parameter are shown with dashed and continuous lines.

If the sensor is in one of the dotted areas, the surface temperature must be reduced (→ 6.2.3). If the warning limit is exceeded despite a heat-dissipating installation, it is possible to additionally mount the contact protection.

If you stay below the typical warning limits in case of normal installation, no measures need to be taken.

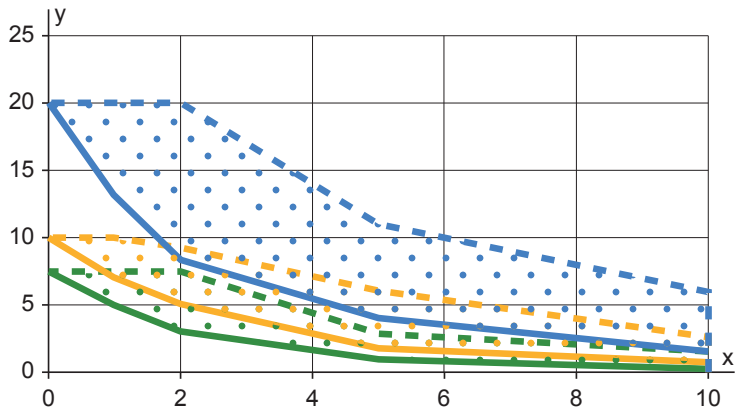
6.2.1 Typical warning limits for O3D300 / O3D302



Parameter "Max. background distance"	
Installation on heat-conductive metal parts with heat conductor (→ 6.2.3)	
Warning limit	Parameter
	< 5 m
	< 30 m
	> 30 m
Normal installation	
Warning limit	Parameter
	< 5 m
	< 30 m
	> 30 m

x = exposure time [ms]
y = frame rate [fps]

6.2.2 Typical warning limits for O3D310 / O3D312



Parameter "Max. background distance"	
Installation on heat-conductive metal parts with heat conductor (→ 6.2.3)	
Warning limit	Parameter
	< 5 m
	< 30 m
	> 30 m
Normal installation	
Warning limit	Parameter
	< 5 m
	< 30 m
	> 30 m

x = exposure time [ms]
y = frame rate [fps]

6.2.3 Reduce surface temperature

With the following measures the surface temperature can be reduced:

► Mount the sensor on heat-conductive metal parts.

> A large-surface contact of the sensor with metal parts increases heat dissipation (e.g. aluminium).

► Use a heat conductor when mounting the sensor on metal parts.

> The heat-conductive effect is increased by means of the heat conductor. The heat conductor is available as accessories (→ 6.4).

► Reduce obstructions around the device. Reduce the density of objects mounted near the device.

> Obstructions around the sensor and a high installation density may have a negative impact on convection (air movement).

► Mount one or two heat sinks on the sensor.

> The heat sinks increase the surface of the sensor, reducing the surface temperature. The heat sinks are available as accessories (→ 6.4).

► Reduce exposure time, frame rate or max. background distance.

> The operating mode used and the parameters can increase the surface temperature.

6.3 Install sensor

Observe the following instructions when installing the sensor:

► Mount the sensor using 2x M5 screws or mounting set.

> The bore dimensions for the M5 screws are indicated in the data sheet.

> The mounting set is available as accessories (→ 6.4).

► Use strain reliefs for all cables connected to the device.

► Do not apply force to the seals of the device (see next figure).

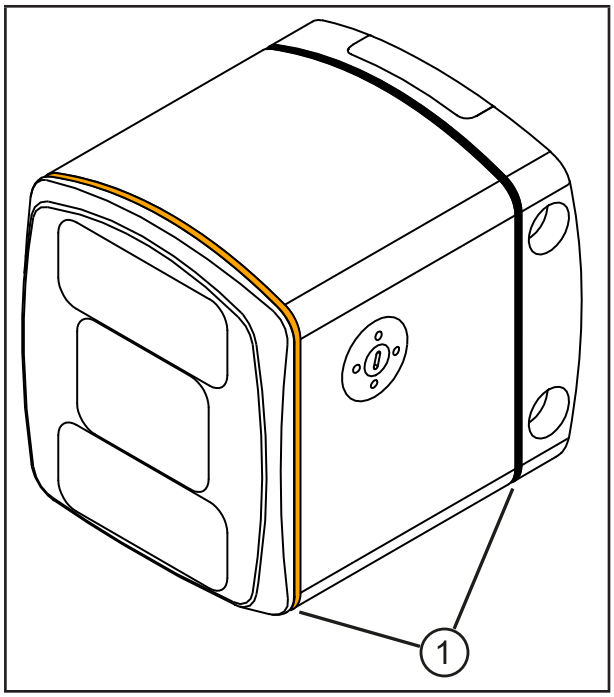
> If the seals are subjected to external mechanical force, the IP protection class specified in the data sheet cannot be guaranteed. Prevent force from acting on the seals with the following measures: provide for a recess in the fixture (for example a groove) or use spacers.

Observe the following instructions when installing an O3D300 and O3D310:

► Mount the sensor so that the focal setter can be accessed with a screw driver.

> The position of the focus adjustment screw is indicated in the scale drawing (→ 12).

If the device is permanently used in wet areas, the nut of the M12 Industrial Ethernet cable (e.g. E11898) may corrode. Use a cable with a high-grade stainless steel nut for permanent use in wet areas.



①: Seals of the device

6.4 Mounting accessories

Depending on the location and type of installation, you can use the following mounting accessories:

Article number	Description
E3D301	Smart Camera mounting set
E3D302	Smart Camera cooling element
E3D303	Smart Camera heat conductor
E3D304	2x Smart Camera cooling element

You can find more information about the accessories at: www.ifm.com

7. Electrical connection

Observe the following instructions before electrical installation.

NOTE

The device must be connected by a qualified electrician. Observe the electrical data in the data sheet.

Device of protection class III (PC III).

The electrical supply must only be made via PELV circuits.

Electric supply must correspond to UL61010-1, chapter 9.4 - Limited Energy:

The overcurrent protection device must switch off a current of 6.6 A in 120 s. For the correct rating of the overcurrent protection device take the technical data of the sensor and wiring into account.

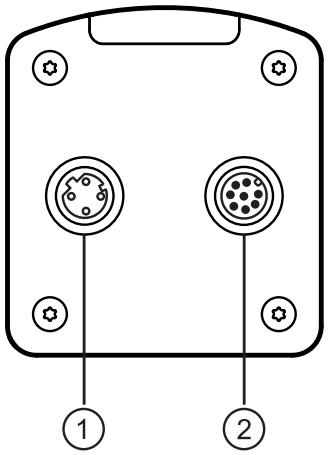
The separation of external circuits must comply with UL61010-2-201, fig. 102.

For cable lengths > 30 m use an additional protection against surge voltages to IEC 6100-4-5.

Disconnect power before connecting the device.

For the scope of validity cULus:
Minimum temperature rating of the cable to be connected to the field wiring terminals: 70 °C.

7.1 Wiring



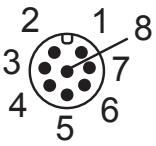
① Ethernet

M12 socket, D-coded, 4 poles

1	TD +
2	RD +
3	TD -
4	RD -
S	Shield

② Power supply

M12 connector, A-coded, 8 poles



1	U+
2	Trigger input
3	GND
4	Switching output 1 - (digital or analogue)
5	Switching output 3 - ready
6	Switching output 2 - (digital)
7	Switching input 1
8	Switching input 2

Cover unused Ethernet connection with the protective cap (E73004).
Tightening torque 0.6...0.8 Nm.

The behaviour of the switching inputs and outputs can be set with the software ifm Vision Assistant.
The setting PNP or NPN always applies to all switching inputs and outputs.

When installing actuators and sensors make sure that the setting is correct
(e.g. photoelectric sensors for triggering).

The switching outputs can also be operated as pulse outputs which reset their switching signal after an adjustable time.

The analogue output provides current / voltage against GND.

7.1.1 Pin 1 / 3 (24 V / GND)

The permissible voltage range is indicated in the data sheet of the sensor.

7.1.2 Pin 2 (trigger input)

The image capture of the sensor can be triggered with a switching signal via the trigger input.

The following trigger edges can be used:

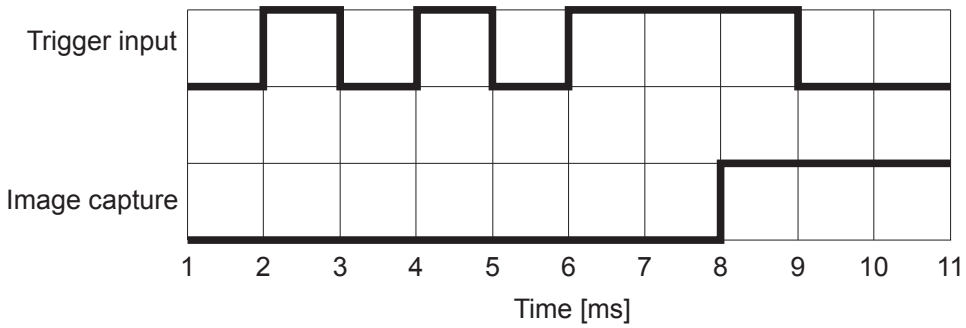
- Falling edge triggers image capture
- Rising edge triggers image capture
- Falling and rising edges trigger image capture

Further possibilities to trigger the sensor:

- Process interface command (→ 13.2)
- Continuous image capture with fixed frame rate

The trigger input is internally debounced. Depending on the electrical installation debouncing of the trigger wire is not necessary.

Internal debouncing prevents several short pulses from triggering. The pulse must be at least 2 ms long to be recognised as a trigger.



7.1.3 Pin 4 / 5 / 6 (switching outputs)

The switching outputs 1 to 3 provide the different sensor statuses. Besides the sensor status the switching outputs can also provide the reference values necessary to solve the application.

The electrical specifications of the switching outputs 1 to 3 are indicated in the data sheet.

Switching output 3 provides the sensor status "Ready for trigger" as default setting.

"Switching output switched" means that the respective sensor status has occurred.

Depending on the setting the sensor status can have one of the following values:

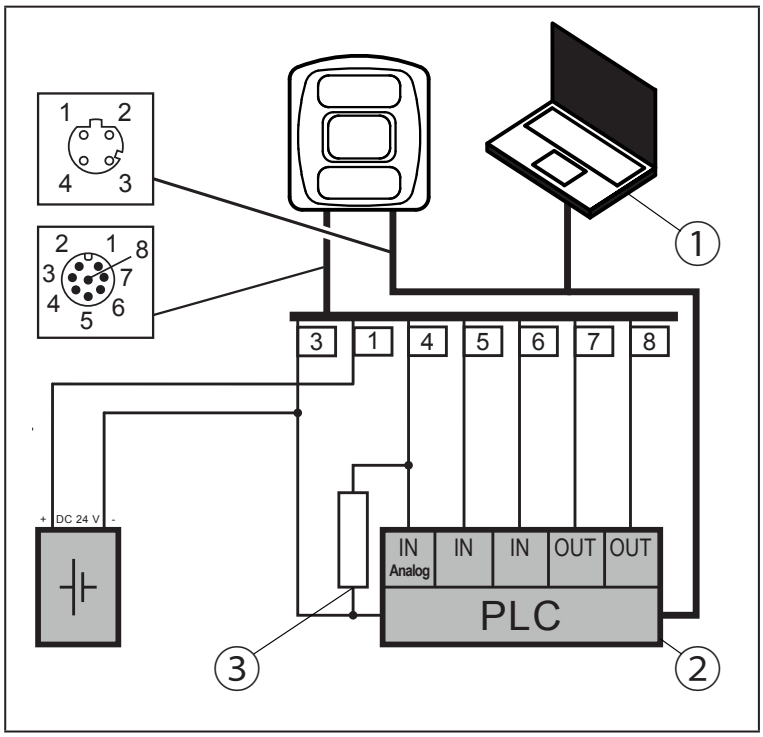
- "Ready for trigger"
  The sensor signals that a new image can be captured. Only with this sensor status trigger operations are processed. For the continuous image capture the status "Ready for trigger" is not output.
- "Image capture finished"
  The sensor signals that the image capture is finished. The sensor status can be used for cascading sensors.
- "Evaluation finished"
  The sensor signals that image processing is finished. At that moment the switching outputs are already updated. The image data is transmitted via Ethernet.
- "Error"
  The sensor signals an internal error. Detailed information about errors can be requested via Ethernet.

7.1.4 Pin 4 (analogue output)

The switching output 1 / analogue output can be used as switching output or analogue current output (4-20 mA) / analogue voltage output (0-10 V).

The analogue current output offers more transmission reliability than the analogue voltage output. The analogue current output is independent of the cable length and ensures better signal quality towards the industrial controller.

In the industrial controller the analogue current is converted into analogue voltage via a load resistor against GND. The load resistor is selected according to the indications in the data sheet. High-resistance load resistors are to be preferred over low-resistance load resistors due to the lower heat development in the device.



① Notebook (parameter setting)

② Industrial controller (evaluate / trigger)

③ Load resistor

Using the ifm Vision Assistant software it is possible to assign one process value each to the start value (4 mA / 0 V) and the end value (20 mA / 10 V) of the analogue output.

7.1.5 Pin 7 / 8 (switching inputs)

The switching inputs provide the following functions:

- Select active application (→ 7.3)

The different parameter settings of the functions are indicated in the software manual.

The electrical data of the switching outputs 1 and 2 is indicated in the data sheet of the sensor.

7.2 Wiring examples

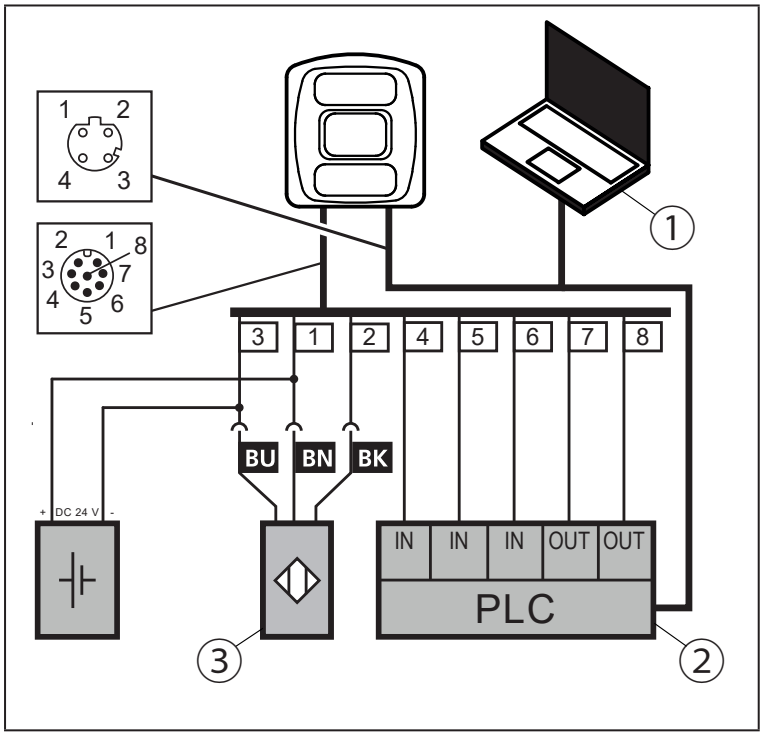
Wiring examples of the sensor are given below.

7.2.1 Trigger image capture with proximity sensor

The sensor can be triggered externally:

- via Ethernet
- via a proximity sensor connected to the trigger input

The following illustration shows the wiring with a proximity sensor.



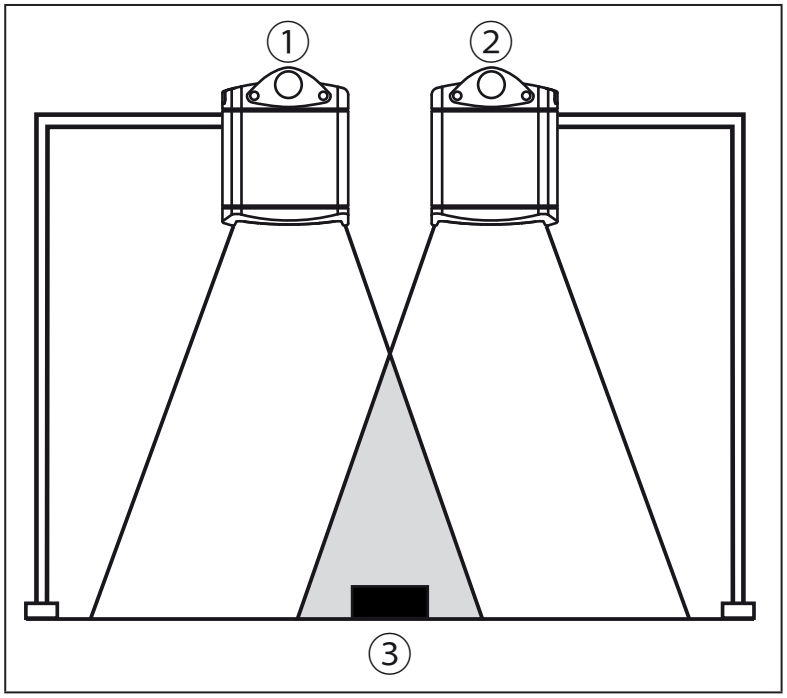
① Notebook (parameter setting)

② Proximity sensor

③ Industrial controller (evaluate / trigger)

7.2.2 Install several sensors next to each other

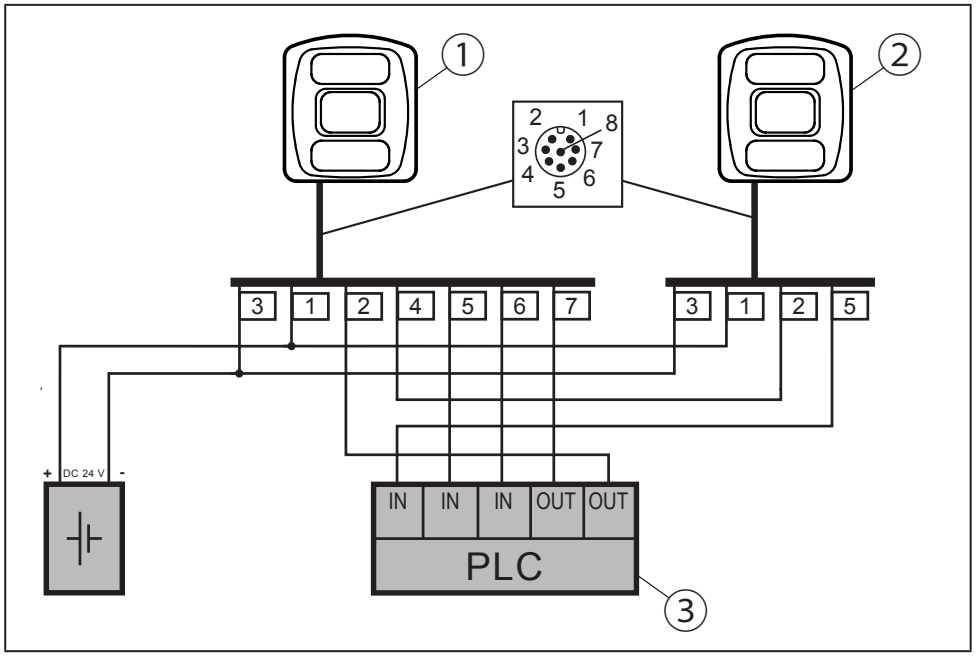
Sensors installed next to each other can cause measurement errors due to simultaneous exposure.



① Device
② Device
③ Object

The measurement errors can be avoided in two ways:

- Cascade sensors via HW trigger
  During cascading a controller triggers the image capture of sensor ① (see figure below). After completion of the image capture, sensor ① automatically triggers sensor ②. At the same time, pin 4 of sensor ① provides the sensor status "Image capture finished". Sensor ② signals the end of the sequence to the industrial controller ③.



① Device
② Device
③ Industrial controller (evaluate / trigger)

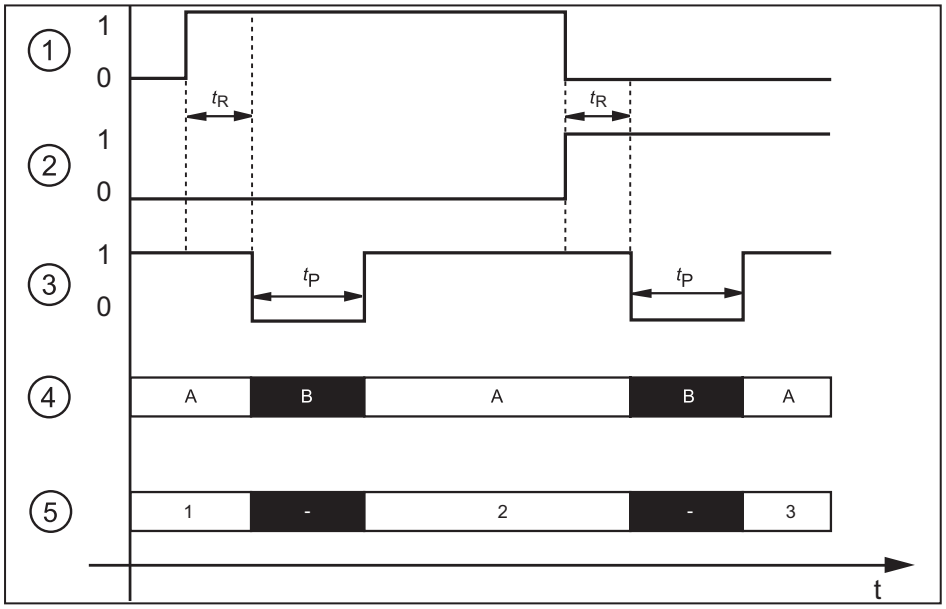
- Use different frequency channels
  With the software ifm Vision Assistant each sensor can be assigned its own frequency channel. The different frequency channels reduce the occurrence of measurement errors.

The ifm Vision Assistant software is available free of charge on our website: www.ifm.com

7.3 Static selection of the application

Up to 32 different inspection tasks can be stored in the sensor. With the corresponding configuration the first four applications can be selected via the two switching inputs.

Input 2	Input 1	Application no.
0	0	1
0	1	2
1	0	3
1	1	4



Example: Selection application 1 → application 2 → application 3

①	Switching input 1 = 0 → 1 → 0
②	Switching input 2 = 0 → 0 → 1
③	READY output
④	Trigger input
	A: trigger enabled
	B: trigger disabled
⑤	ID number of the active application

For the selection of the applications the monitoring time t_R and the trigger disable time t_P have to be taken into consideration.

Monitoring time t_R: After a change in edges the external selection of the application does not start before the state of both switching inputs remains stable for 20 ms.

Trigger disable time t_P: The trigger input is disabled during the selection of the application. The disable time depends on:

- the number of applications on the device
- the number of models in the application to be activated

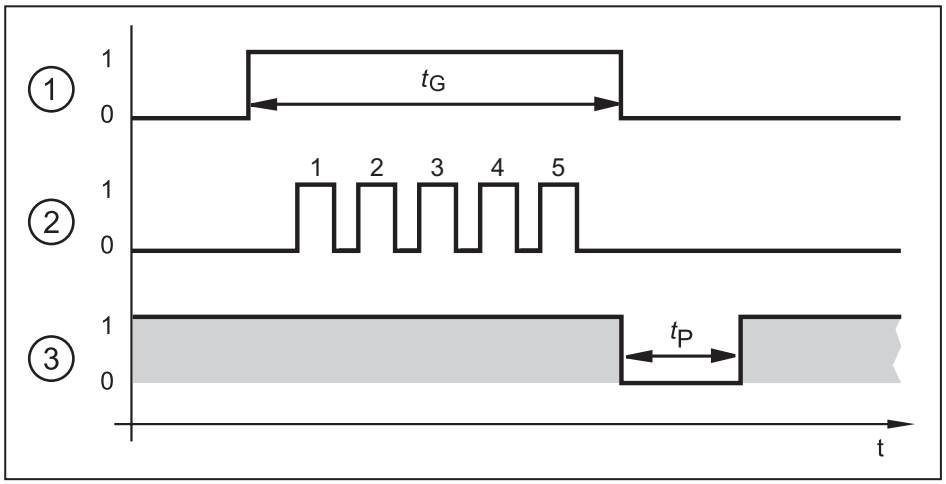
The figure above shows the PNP output logic (factory setting). The behaviour of the NPN output logic is the opposite of that of the PNP output logic:

- PNP output logic: In case of a high signal (1), voltage is applied.
- NPN output logic: In case of a low signal (0), voltage is applied.

For more detailed information about the configuration of the selection of the application we refer you to the software manual of the device. www.ifm.com

7.4 Pulse-controlled selection of the application

As an alternative to the static selection the selection of the application can also be pulse-controlled.



①	Gate signal, switching input 1 = 0 → 1 → 0 (t_G = signal active)
②	Pulse signal, switching input 2 or trigger input = 0 → 5 pulses → 0
③	READY output

While there is an active signal on switching input 1 (gate signal), the device counts incoming pulses and activates the respective application.

Number of pulses = ID number of the application

Either switching input 2 or the trigger input of the device can be used as pulse input.

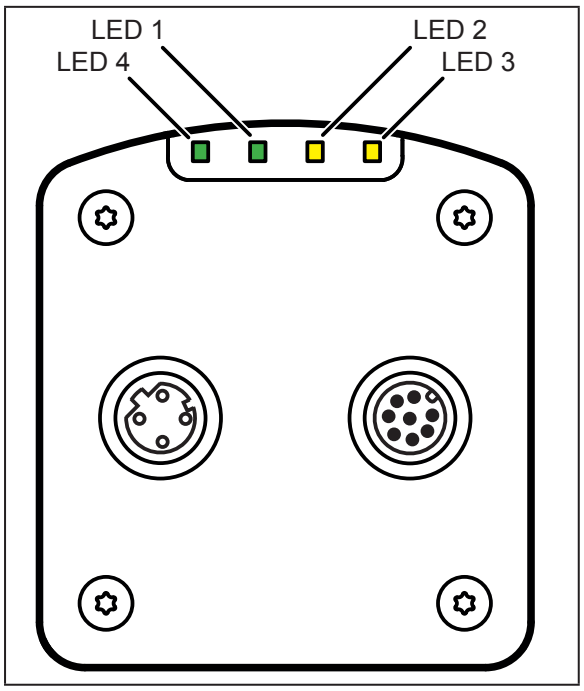
The figure above shows the PNP output logic (factory setting). The behaviour of the NPN output logic is the opposite of that of the PNP output logic:

- PNP output logic: In case of a high signal (1), voltage is applied.
- NPN output logic: In case of a low signal (0), voltage is applied.

For more detailed information about the configuration of the selection of the application we refer you to the software manual of the device. www.ifm.com

8. Indicators

Via the LED indicators 1 - 4 the sensor signals the current operating state.



LED 4 (Ethernet)	LED 1 (Power)	LED 2 (Out 1)	LED 3 (Out 2)	Description
	On			Sensor is ready for operation, supply voltage applied
	Flashes at 0.5 Hz			No parameters set or parameter setting was not loaded into the sensor On Off
	Flashes 2x at 0.5 Hz			Sensor is in the parameter setting mode On Off
		On		Switching output 1 switched
		Flashes at 8 Hz		Switching output 1 shorted
			On	Switching output 2 switched
			Flashes at 8 Hz	Switching output 2 shorted
On				Ethernet connected
Flashes				Ethernet transmitting data
Off				Ethernet not connected
		Flashes at 8 Hz	Flashes at 8 Hz	Sensor signals internal error
		Flashes at 2 Hz	Flashes at 2 Hz	Sensor signals correctable error. The error information can be read via Ethernet
	Running light ⇒			Device booting
	Running light ⇐			Sensor carrying out firmware update

9. Set-up

After power on the device is put into operation. After 15 seconds the sensor is in the evaluation mode where saved applications are executed. The indicators signal the current operating state (→ 8).

Up to 32 applications can be saved on the sensor. An application can be activated in different ways:

- ifm Vision Assistant software
- Process interface command
- Switching input 1 and 2
- Switching input 1 and trigger input

9.1 Set parameters of the device

The sensor is set using the ifm Vision Assistant software (→ see software manual).

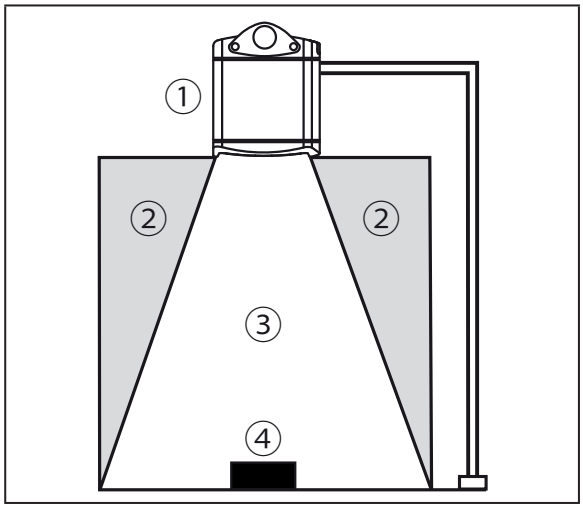
The software ifm Vision Assistant and detailed information about the measuring principle of the device are described in the software manual.

The ifm Vision Assistant software is available free of charge on our website:
www.ifm.com

The software manual is available on our website:
www.ifm.com

9.2 Detect object

The conditions which lead to a high detection rate of objects are described below.



① Device
② Zone of influence
③ Field of view
④ Object

Optimum detection of an object ④ is given if the following conditions are met:

- Object is positioned in the field of view ③
- Object is the nearest visible object to the sensor ①
- Zone of influence ② is clear from objects (obstructions etc.)
- Lens window of the sensor is free from soiling.

If the conditions are not met, measurement errors may occur.

9.3 Transmit process values

9.3.1 Transmit process values of the completeness monitoring via EtherNet/IP

The device can transmit the process values to a PLC via the EtherNet/IP fieldbus. The process values are displayed in the ifm Vision Assistant as output string as below:

Only one fieldbus can be active at a time. The fieldbus is adjustable (→ software manual).

In the output string the process values are separated by a semicolon. The output string is transferred to a PLC in the displayed sequence.

Observe the following remarks for the transmission of the output string to a PLC:

- Bytes 0 to 7 are part of the output string. They are not displayed in the ifm Vision Assistant (see screenshot above).
- Semicolons ";" in the output string are not transferred.
- Float values are converted into 16-bit integers before the transmission.
- All numerical values are converted into 16-bit integers before the transmission.

The output string is composed of the following:

star;0;00;0;+0.000;01;7;-0.068;02;6;+0.013;03;0;+0.001;stop

Byte no.	Data	Coding	Process value	Unit	Description	Comments
0 1	2#0000_0000 2#0010_0000	Binary Binary	1.5		Duplicated command word	• Bit 1.5 shows a successful trigger command
2 3	2#0000_0000 2#0000_0000	Decimal Decimal			Synchronous / asynchronous message identification	
4 5	30 0	Decimal Decimal	30		Message counter	• The device has received 30 messages • Increments by 1 with each action (trigger, message sent etc.).
6 7	0 0	Decimal Decimal			Reserved	
8 9 10 11	s t a r	ASCII ASCII ASCII ASCII	star		Start string	
12 13	0 0	Decimal Decimal	0		Status of all ROIs (0 = bad, 1 = good)	Shows the status of the completeness monitoring
14 15	0 0	Decimal Decimal	0		ROI ID	With activated position adjustment bytes 14 and 15 are used by it. 0 = position is not adjusted 1 = position is adjusted All following data is shifted by 2 bytes; i.e. the first ROI ID starts with bytes 16 and 17.

Byte no.	Data	Coding	Process value	Unit	Description	Comments
16 17	0 0	Decimal Decimal	0		ROI status	ROI status: 0 = good 1 = reference level not taught 2 = teaching failed 3 = reference level invalid 4 = no valid pixels 5 = reference level does not contain any valid pixels 6 = overfill 7 = underfill
18 19	0 0	Decimal Decimal	0	mm	ROI value	
20 21	1 0	Decimal Decimal	1		ROI ID	
22 23	7 0	Decimal Decimal	7		ROI status	
24 25	-67 -1	Decimal Decimal	-67	mm	ROI value	
26 27	2 0	Decimal Decimal	2		ROI ID	
28 29	6 0	Decimal Decimal	6		ROI status	
30 31	14 0	Decimal Decimal	14	mm	ROI value	
32 33	3 0	Decimal Decimal	3		ROI ID	
34 35	0 0	Decimal Decimal	0		ROI status	
36 37	0 0	Decimal Decimal	0	mm	ROI value	
38 39 40 41	s t o p	ASCII ASCII ASCII ASCII	stop		Stop string	

Faulty execution of a command leads to the following status:

- Error bit = 1
- Duplicated command word is displayed
- Asynchronous message bit = 0
- Asynchronous message identification = 0
- Message counter increments by 1

9.3.2 Transmit process values of the completeness monitoring via PROFINET

The device can transmit the process values to a PLC via the PROFINET fieldbus. The process values are displayed in the ifm Vision Assistant as output string as below:

Only one fieldbus can be active at a time. The fieldbus is adjustable (→ software manual).

UK

In the output string the process values are separated by a semicolon. The output string is transferred to a PLC in the displayed sequence.

Observe the following remarks for the transmission of the output string to a PLC:

- Bytes 0 to 7 are part of the output string. They are not displayed in the ifm Vision Assistant (see screenshot above).
- Semicolons ";" in the output string are not transferred.
- Float values are converted into 16-bit integers before the transmission.
- All numerical values are converted into binary 16-bit integers before the transmission.

The output string is composed of the following:

star;0;00;0;+0.000;01;7;-0.068;02;6;+0.013;03;0;+0.001;stop

Byte no.	Data	Coding	Process value	Unit	Description	Comments
0 1	2#0010_0000 2#0000_0000	Binary Binary	0.5		Duplicated command word	• Bit 0.5 shows a successful trigger command
2 3	2#0000_0000 2#0000_0000	Decimal Decimal			Synchronous / asynchronous message identification	
4 5	0 30	Decimal Decimal	30		Message counter	• The device has received 30 messages. • Increments by 1 with each action (trigger, message sent etc.).
6 7	0 0	Decimal Decimal			Reserved	
8 9 10 11	s t a r	ASCII ASCII ASCII ASCII	star		Start string	
12 13	0 0	Decimal Decimal	0		Status of all ROIs (0 = bad, 1 = good)	Shows the status of the completeness monitoring
14 15	0 0	Decimal Decimal	0		ROI ID	With activated position adjustment bytes 14 and 15 are used by it. 0 = position is not adjusted 1 = position is adjusted All following data is shifted by 2 bytes; i.e. the first ROI ID starts with bytes 16 and 17.

Byte no.	Data	Coding	Process value	Unit	Description	Comments
16 17	0 0	Decimal Decimal	0		ROI status	ROI status: 0 = good 1 = reference level not taught 2 = teaching failed 3 = reference level invalid 4 = no valid pixels 5 = reference level does not contain any valid pixels 6 = overfill 7 = underfill
18 19	0 0	Decimal Decimal	0	mm	ROI value	
20 21	0 1	Decimal Decimal	1		ROI ID	
22 23	0 7	Decimal Decimal	7		ROI status	
24 25	-1 -67	Decimal Decimal	-67	mm	ROI value	
26 27	0 2	Decimal Decimal	2		ROI ID	
28 29	0 6	Decimal Decimal	6		ROI status	
30 31	0 14	Decimal Decimal	14	mm	ROI value	
32 33	0 3	Decimal Decimal	3		ROI ID	
34 35	0 0	Decimal Decimal	0		ROI status	
36 37	0 0	Decimal Decimal	0	mm	ROI value	
38 39 40 41	s t o p	ASCII ASCII ASCII ASCII	stop		Stop string	

Faulty execution of a command leads to the following status:

- Error bit = 1
- Duplicated command word is displayed
- Asynchronous message bit = 0
- Asynchronous message identification = 0
- Message counter increments by 1

9.3.3 Transmit process values of the completeness monitoring via TCP/IP

The device can transmit the process values to a PLC via the TCP/IP protocol. The process values are displayed in the ifm Vision Assistant as output string as below:

```
star;0;00;0;+0.000;01;7;-0.068;02;6;+0.013;03;0;
+0.001;stop
```

In the output string the process values are separated by a semicolon. The output string is transferred to a PLC in the displayed sequence.

UK

Observe the following remarks for the transmission of the output string to a PLC:

- Semicolons ";" in the output string are not transferred.
- All numerical values are converted into binary 16-bit integers before the transmission.

The output string is composed of the following (data type: ASCII):

star;0;00;0;+0.000;01;7;-0.068;02;6;+0.013;03;0;+0.001;stop

Process value	Unit	Description
star		Start string
0		Status of all ROIs (0 = bad, 1 = good)
00		ROI ID
0		ROI status
+0.000	m	ROI value
01		ROI ID
7		ROI status
-0.068	m	ROI value
02		ROI ID
6		ROI status
+0.013	m	ROI value
03		ROI ID
0		ROI status
+0.001	m	ROI value
stop		Stop string

ROI status:
0 = good
1 = reference level not taught
2 = teaching failed
3 = reference level invalid
4 = no valid pixels
5 = reference level does not contain any valid pixels
6 = overfill
7 = underfill

9.3.4 Transmit process values of the dimensioning of the object via EtherNet/IP

The device can transmit the process values to a PLC via the EtherNet/IP fieldbus. The process values are displayed in the ifm Vision Assistant as output string as below:

```
star;1;0.200;0.150;0.307;+0.002;-0.044;
+0.100;170;099;100;098;stop
```

Only one fieldbus can be active at a time. The fieldbus is adjustable (→ software manual).

In the output string the process values are separated by a semicolon. The output string is transferred to a PLC in the displayed sequence.

Observe the following remarks for the transmission of the output string to a PLC:

- The output string is adjustable. The process values to be transferred can be set in the ifm Vision Assistant.
- Bytes 0 to 7 are part of the output string. They are not displayed in the ifm Vision Assistant (see screenshot above).
- Semicolons ";" in the output string are not transferred.
- Float values are converted into 16-bit integers before the transmission.
- All numerical values are converted into binary 16-bit integers before the transmission.

The output string is composed of the following:

star;1;0.104;0.088;0.109;+0.021;-0.011;+0.389;158;097;094;097;stop

Byte no.	Data	Coding	Process value	Unit	Description	Comments
0 1	2#0000_0000 2#0010_0000	Binary Binary	1.5		Duplicated command word	• Bit 1.5 shows a successful trigger command
2 3	2#0000_0000 2#0000_0000	Binary Binary			Synchronous / asynchronous message identification	
4 5	2#0000_0011 2#0000_0000	Binary Binary	3		Message counter	• The device has received 3 messages. • Increments by 1 with each action (trigger, message sent etc.).
6 7	2#0000_0000 2#0000_0000	Binary Binary			Reserved	
8 9 10 11	s t a r	ASCII ASCII ASCII ASCII	star		Start string	
12 13	2#0000_0001 2#0000_0000	Binary Binary	1		Result bit	0 = no box found 1 = box found
14 15	104 0	Decimal Decimal	104	mm	Width	
16 17	88 0	Decimal Decimal	88	mm	Height	
18 19	108 0	Decimal Decimal	109	mm	Length	
20 21	21 0	Decimal Decimal	21		x coordinate	

Byte no.	Data	Coding	Process value	Unit	Description	Comments
22 23	-11 -1	Decimal Decimal	-11		y coordinate	
24 25	-124 1	Decimal Decimal	389		z coordinate	
26 27	-98 0	Decimal Decimal	158		Degree of rotation	
28 29	97 0	Decimal Decimal	97		Quality width	
30 31	93 0	Decimal Decimal	94		Quality height	
32 33	97 0	Decimal Decimal	97		Quality length	
34 35 36 37	s t o p	ASCII ASCII ASCII ASCII	stop		Stop string	

Faulty execution of a command leads to the following status:

- Error bit = 1
- Duplicated command word is displayed
- Asynchronous message bit = 0
- Asynchronous message identification = 0
- Message counter increments by 1

9.3.5 Transmit process values of the dimensioning of the object via PROFINET

The device can transmit the process values to a PLC via the PROFINET fieldbus. The process values are displayed in the ifm Vision Assistant as output string as below:

```
star;1;0.200;0.150;0.307;+0.002;-0.044;
+0.100;170;099;100;098;stop
```

Only one fieldbus can be active at a time. The fieldbus is adjustable (→ software manual).

In the output string the process values are separated by a semicolon. The output string is transferred to a PLC in the displayed sequence.

Observe the following remarks for the transmission of the output string to a PLC:

- The output string is adjustable. The process values to be transferred can be set in the ifm Vision Assistant.
- Bytes 0 to 7 are part of the output string. They are not displayed in the ifm Vision Assistant (see screenshot above).
- Semicolons ";" in the output string are not transferred.
- Float values are converted into 16-bit integers before the transmission.
- All numerical values are converted into binary 16-bit integers before the transmission.

The output string is composed of the following:

star;1;0.104;0.088;0.109;+0.021;-0.011;+0.389;158;097;094;097;stop

Byte no.	Data	Coding	Process value	Unit	Description	Comments
0 1	2#0010_0000 2#0000_0000	Binary Binary	0.5		Duplicated command word	• Bit 0.5 shows a successful trigger command
2 3	2#0000_0000 2#0000_0000	Binary Binary			Synchronous / asynchronous message identification	
4 5	2#0000_0000 2#0000_0011	Binary Binary	3		Message counter	• The device has received 3 messages. • Increments by 1 with each action (trigger, message sent etc.).
6 7	2#0000_0000 2#0000_0000	Binary Binary			Reserved	
8 9 10 11	s t a r	ASCII ASCII ASCII ASCII	star		Start string	
12 13	2#0000_0000 2#0000_0001	Binary Binary	1		Result bit	0 = no box found 1 = box found
14 15	0 104	Decimal Decimal	104	mm	Width	
16 17	0 88	Decimal Decimal	88	mm	Height	
18 19	0 109	Decimal Decimal	109	mm	Length	
20 21	0 21	Decimal Decimal	21		x coordinate	

Byte no.	Data	Coding	Process value	Unit	Description	Comments
22 23	-1 -11	Decimal Decimal	-11		y coordinate	
24 25	1 -124	Decimal Decimal	389		z coordinate	
26 27	0 -98	Decimal Decimal	158		Degree of rotation	
28 29	0 97	Decimal Decimal	97		Quality width	
30 31	0 94	Decimal Decimal	94		Quality height	
32 33	0 97	Decimal Decimal	97		Quality length	
34 35 36 37	s t o p	ASCII ASCII ASCII ASCII	stop		Stop string	

Faulty execution of a command leads to the following status:

- Error bit = 1
- Duplicated command word is displayed
- Asynchronous message bit = 0
- Asynchronous message identification = 0
- Message counter increments by 1

9.3.6 Transmit process values of the dimensioning of the object via TCP/IP

The device can transmit the process values to a PLC via the TCP/IP protocol. The process values to be sent can be selected in the ifm Vision Assistant. The process values are displayed in the ifm Vision Assistant as output string as below:

```
star;1;0.200;0.150;0.307;+0.002;-0.044;
+0.100;170;099;100;098;stop
```

In the output string the process values are separated by a semicolon. The output string is transferred to a PLC in the displayed sequence.

Observe the following remarks for the transmission of the output string to a PLC:

- Semicolons ";" in the output string are not transferred.
- All numerical values are converted into binary 16-bit integers before the transmission.

The output string is composed of the following (data type: ASCII):

star;1;0.104;0.088;0.109;+0.021;-0.011;+0.389;158;097;094;097;stop

Process value	Unit	Description
star		Start string
1		Object found
0.104	m	Width
0.088	m	Height
0.109	m	Length
+0.021		x coordinate
-0.011		y coordinate
+0.389		z coordinate
158		Degree of rotation
097		Quality width
094		Quality height
097		Quality length
stop		Stop string

9.3.7 Transmit process values of the level measurement via EtherNet/IP

The device can transmit the process values to a PLC via the EtherNet/IP fieldbus. The process values are displayed in the ifm Vision Assistant as output string as below:

Only one fieldbus can be active at a time. The fieldbus is adjustable (→ software manual).

UK

The output string is transferred to a PLC in the displayed sequence.

Observe the following remarks for the transmission of the output string to a PLC:

- Bytes 0 to 7 are part of the output string. They are not displayed in the ifm Vision Assistant (see screenshot above).
- Semicolons ";" in the output string are not transferred.
- Float values are converted into 16-bit integers before the transmission.
- All numerical values are converted into binary 16-bit integers before the transmission.

The output string is composed of the following:

0070

Byte no.	Data	Coding	Process value	Unit	Description	Comments
0 1	2#0000_0000 2#0010_0000	Binary Binary	1.5		Duplicated command word	Bit 1.5 shows a successful trigger command
2 3	2#0000_0000 2#0000_0000	Decimal Decimal			Synchronous / asynchronous message identification	
4 5	30 0	Decimal Decimal	30		Message counter	• The device has received 30 messages. • Increments by 1 with each action (trigger, message sent etc.).
6 7	0 0	Decimal Decimal			Reserved	
8 9	0 0	Decimal Decimal	0		Status of all ROIs (0 = bad, 1 = good)	Shows the status of the level measurement
10 11	0 0	Decimal Decimal	0		ROI ID	ROI status: 0 = good 6 = overfill 7 = underfill
12 13	7 0	Decimal Decimal	7		ROI status	
14 15	0 0	Decimal Decimal	0	mm	ROI value	

Faulty execution of a command leads to the following status:

- Error bit = 1
- Duplicated command word is displayed
- Asynchronous message bit = 0
- Asynchronous message identification = 0
- Message counter increments by 1

9.3.8 Transmit process values of the level measurement via PROFINET

The device can transmit the process values to a PLC via the PROFINET fieldbus. The process values are displayed in the ifm Vision Assistant as output string as below:

Only one fieldbus can be active at a time. The fieldbus is adjustable (→ software manual).

The output string is transferred to a PLC in the displayed sequence.

Observe the following remarks for the transmission of the output string to a PLC:

- Bytes 0 to 7 are part of the output string. They are not displayed in the ifm Vision Assistant (see screenshot above).
- Semicolons ";" in the output string are not transferred.
- Float values are converted into 16-bit integers before the transmission.
- All numerical values are converted into binary 16-bit integers before the transmission.

The output string is composed of the following:

0070

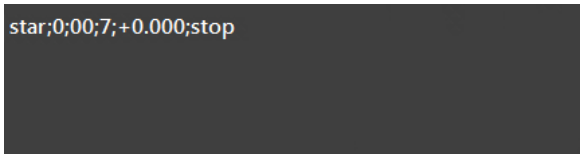
Byte no.	Data	Coding	Process value	Unit	Description	Comments
0 1	2#0010_0000 2#0000_0000	Binary Binary	0.5		Duplicated command word	Bit 0.5 shows a successful trigger command
2 3	2#0000_0000 2#0000_0000	Decimal Decimal			Synchronous / asynchronous message identification	
4 5	0 30	Decimal Decimal	30		Message counter	• The device has received 30 messages. • Increments by 1 with each action (trigger, message sent etc.).
6 7	0 0	Decimal Decimal			Reserved	
8 9	0 0	Decimal Decimal	0		Status of all ROIs (0 = bad, 1 = good)	Shows the status of the level measurement
10 11	0 0	Decimal Decimal	0		ROI ID	ROI status: 0 = good 6 = overfill 7 = underfill
12 13	0 7	Decimal Decimal	7		ROI status	
14 15	0 0	Decimal Decimal	0	mm	ROI value	

Faulty execution of a command leads to the following status:

- Error bit = 1
- Duplicated command word is displayed
- Asynchronous message bit = 0
- Asynchronous message identification = 0
- Message counter increments by 1

9.3.9 Transmit process values of the level measurement via TCP/IP

The device can transmit the process values to a PLC via the TCP/IP protocol. The process values are displayed in the ifm Vision Assistant as output string as below:



In the output string the process values are separated by a semicolon. The output string is transferred to a PLC in the displayed sequence.

UK

Observe the following remarks for the transmission of the output string to a PLC:

- Semicolons ";" in the output string are not transferred.
- All numerical values are converted into binary 16-bit integers before the transmission.

The output string is composed of the following (data type: ASCII):

star;0;00;7;+0.000;stop

Process value	Unit	Description
star		Start string
0		Status of all ROIs (0 = bad, 1 = good)
00		ROI ID
7		ROI status
+0.000	m	ROI value
stop		Stop string

ROI status:
0 = good
6 = overfill
7 = underfill

9.3.10 Transmit process values of robot pick & place via EtherNet/IP

The device can transmit the process values to a PLC via the EtherNet/IP fieldbus.

Only one fieldbus can be active at a time. The fieldbus is adjustable (→ software manual).

In the output string the process values are separated by a semicolon. The output string is transmitted to a PLC in the displayed sequence.

Observe the following notes to transmit the output string to a PLC:

- Bytes 0 to 7 are part of the output string. They are not displayed in the ifm Vision Assistant.
- Bytes 14 to 35 are repeated for each object set under "Number of objects" (maximum 10 repetitions).
- Semicolons ";" in the output string are not transmitted.
- Float values are converted into 16-bit integers before the transmission.
- All numerical values are converted into 16-bit integers before the transmission.

The output string is as follows:

0;01;08;1;0.338;0.142;0.452;+0.075;-0.071;+0.783;078;+000;+000;+056

Byte no.	Data	Encoding	Process value	Unit	Description	Comment
0 1	2#0010_0000 2#0000_0000	Binary Binary	0.5		Duplicated command word	• Bit 0.5 indicates a successful trigger command.
2 3	2#0000_0000 2#0000_0000	Binary Binary			Synchronous / asynchronous message identification	
4 5	2#0000_0000 2#0000_0011	Binary Binary	3		Message counter	• The device has received 3 messages. • Increments by 1 with each action (trigger, message sent etc.).
6 7	2#0000_0000 2#0000_0000	Binary Binary			Reserved	
8 9	0 0	Decimal Decimal	0		Error	Error: 0 = no error 1 = undefined error 2 = no object found
10 11	1 0	Decimal Decimal	01		Number of objects	Number of found objects
12 13	8 0	Decimal Decimal	08		Number of object candidates	Number of found and checked object candidates
14 15	1 0	Binary Binary	1		Object found	0 = no object found 1 = object found
16 17	338 0	Decimal Decimal	338	mm	Width	The broadest dimension of the object surface.
18 19	142 0	Decimal Decimal	142	mm	Height	The object height relative to the base plate.
20 21	452 0	Decimal Decimal	452	mm	Length	The longest dimension of the object surface.
22 23	75 0	Decimal Decimal	75		Centre point X	The X coordinate of the centre point of the object surface (in the user's coordinate system).
24 25	-71 0	Decimal Decimal	-71		Centre point Y	The Y coordinate of the centre point of the object surface (in the user's coordinate system).

Byte no.	Data	Encoding	Process value	Unit	Description	Comment
26 27	783 0	Decimal Decimal	783		Centre point Z	The Z coordinate of the centre point of the object surface (in the user's coordinate system).
28 29	78 0	Decimal Decimal	078		Yaw angle	The yaw angle is between the x axis (world coordinate system) and the vector along the "length" of the object.
30 31	0 0	Decimal Decimal	+000		Rotation X	Rotation about the X axis of the recognised object (in the user's coordinate system).
32 33	0 0	Decimal Decimal	+000		Rotation Y	Rotation about the Y axis of the recognised object (in the user's coordinate system).
34 35	56 0	Decimal Decimal	+056		Rotation Z	Rotation about the Z axis of the recognised object (in the user's coordinate system).

The incorrect execution of a command leads to the following status:

- Error bit = 1
- Duplicated command word is displayed
- Asynchronous message bit = 0
- Asynchronous message identification = 0
- Message counter increments by 1

UK

9.3.11 Transmit process values of the robot pick & place measurement via PROFINET

The device can transmit the process values to a PLC via the PROFINET fieldbus.

Only one fieldbus can be active at a time. The fieldbus is adjustable (→ software manual).

In the output string the process values are separated by a semicolon. The output string is transmitted to a PLC in the displayed sequence.

Observe the following notes to transmit the output string to a PLC:

- Bytes 0 to 7 are part of the output string. They are not displayed in the ifm Vision Assistant.
- Bytes 14 to 35 are repeated for each object set under "Number of objects" (maximum 10 repetitions).
- Semicolons ";" in the output string are not transmitted.
- Float values are converted into 16-bit integers before the transmission.
- All numerical values are converted into 16-bit integers before the transmission.

The output string is as follows:

0;01;08;1;0.338;0.142;0.452;+0.075;-0.071;+0.783;078;+000;+000;+056

Byte no.	Data	Encoding	Process value	Unit	Description	Comment
0 1	2#0010_0000 2#0000_0000	Binary Binary	0.5		Duplicated command word	• Bit 0.5 indicates a successful trigger command.
2 3	2#0000_0000 2#0000_0000	Binary Binary			Synchronous / asynchronous message identification	
4 5	2#0000_0000 2#0000_0011	Binary Binary	3		Message counter	• The device has received 3 messages. • Increments by 1 with each action (trigger, message sent etc.).
6 7	2#0000_0000 2#0000_0000	Binary Binary			Reserved	
8 9	0 0	Decimal Decimal	0		Error	Error: 0 = no error 1 = undefined error 2 = no object found
10 11	1 0	Decimal Decimal	01		Number of objects	Number of found objects
12 13	8 0	Decimal Decimal	08		Number of object candidates	Number of found and checked object candidates
14 15	1 0	Binary Binary	1		Object found	0 = no object found 1 = object found
16 17	338 0	Decimal Decimal	338	mm	Width	The broadest dimension of the object surface.
18 19	142 0	Decimal Decimal	142	mm	Height	The object height relative to the base plate.
20 21	452 0	Decimal Decimal	452	mm	Length	The longest dimension of the object surface.
22 23	75 0	Decimal Decimal	75		Centre point X	The X coordinate of the centre point of the object surface (in the user's coordinate system).
24 25	-71 0	Decimal Decimal	-71		Centre point Y	The Y coordinate of the centre point of the object surface (in the user's coordinate system).

Byte no.	Data	Encoding	Process value	Unit	Description	Comment
26 27	783 0	Decimal Decimal	783		Centre point Z	The Z coordinate of the centre point of the object surface (in the user's coordinate system).
28 29	78 0	Decimal Decimal	078		Yaw angle	The yaw angle is between the x axis (world coordinate system) and the vector along the "length" of the object.
30 31	0 0	Decimal Decimal	+000		Rotation X	Rotation about the X axis of the recognised object (in the user's coordinate system).
32 33	0 0	Decimal Decimal	+000		Rotation Y	Rotation about the Y axis of the recognised object (in the user's coordinate system).
34 35	56 0	Decimal Decimal	+056		Rotation Z	Rotation about the Z axis of the recognised object (in the user's coordinate system).

The incorrect execution of a command leads to the following status:

- Error bit = 1
- Duplicated command word is displayed
- Asynchronous message bit = 0
- Asynchronous message identification = 0
- Message counter increments by 1

UK

9.3.12 Transmit process values of robot pick & place via TCP/IP

The device can transmit the process values to a PLC via the TCP/IP protocol. In the ifm Vision Assistant the process values are displayed as output string as shown below:

```
star;0;01;08;1;0.338;0.142;0.452;+0.075;-0.071;
+0.783;078;+000 ;+000;+056;stop
```

In the output string the process values are separated by a semicolon. The output string is transmitted to a PLC in the displayed sequence.

Observe the following notes to transmit the output string to a PLC:

- Semicolons ";" in the output string are not transmitted.
- The process values "Object found" to "Rotation Z" are repeated for each object set under "Number of objects" (maximum 10 repetitions).
- All numerical values are converted into 16-bit integers before the transmission.

The output string is as follows (data type: ASCII):

star;0;01;08;1;0.338;0.142;0.452;+0.075;-0.071;+0.783;078;+000;+000;+056;stop

Process value	Unit	Description
star		Start string
0		Error
01		Number of objects
08		Number of object candidates
1		1 = no object found 0 = object found
0.338	mm	Width
0.142	mm	Height
0.452	mm	Length
+0.075		Centre point X
-0.071		Centre point Y
+0.783		Centre point Z
078		Yaw angle
+000		Rotation X
+000		Rotation Y
+056		Rotation Z
stop		Stop string

9.3.13 Transmit process values of depalletising via EtherNet/IP

The device can transmit the process values to a PLC via the EtherNet/IP fieldbus.

Only one fieldbus can be active at a time. The fieldbus is adjustable (→ software manual).

In the output string the process values are separated by a semicolon. The output string is transmitted to a PLC in the displayed sequence.

Observe the following notes to transmit the output string to a PLC:

- Bytes 0 to 7 are part of the output string. They are not displayed in the ifm Vision Assistant.
- Semicolons ";" in the output string are not transmitted.
- Float values are converted into 16-bit integers before the transmission.
- All numerical values are converted into 16-bit integers before the transmission.

The output string is as follows:

1;0.200;0.150;0.307;+00.002;-10.044;+03.100;+170;-133;-132;02;1;098;00;1

Byte no.	Data	Encoding	Process value	Unit	Description	Comment
0 1	2#0010_0000 2#0000_0000	Binary Binary	0.5		Duplicated command word	• Bit 0.5 indicates a successful trigger command.
2 3	2#0000_0000 2#0000_0000	Binary Binary			Synchronous / asynchronous message identification	
4 5	2#0000_0000 2#0000_0011	Binary Binary	3		Message counter	• The device has received 3 messages. • Increments by 1 with each action (trigger, message sent etc.).
6 7	2#0000_0000 2#0000_0000	Binary Binary			Reserved	
8 9	1 0	Binary Binary	**1**		Object found	0 = no object found 1 = object found
10 11	200 0	Decimal Decimal	**200**	**mm**	Width	The broadest dimension of the object surface.
12 13	150 0	Decimal Decimal	**150**	**mm**	Height	The object height relative to the base plate.
14 15	307 0	Decimal Decimal	**307**	**mm**	Length	The longest dimension of the object surface.
16 17	2 0	Decimal Decimal	**+2**		Centre point X	The X coordinate of the centre point of the object surface (in the user's coordinate system).
18 19	10044 0	Decimal Decimal	**-10044**		Centre point Y	The Y coordinate of the centre point of the object surface (in the user's coordinate system).
20 21	3100 0	Decimal Decimal	**+3100**		Centre point Z	The Z coordinate of the centre point of the object surface (in the user's coordinate system).
22 23	170 0	Decimal Decimal	**+170**		Rotation X	Rotation about the X axis of the recognised object (in the user's coordinate system)

Byte no.	Data	Encoding	Process value	Unit	Description	Comment
24 25	-133 0	Decimal Decimal	-133		Rotation Y	Rotation about the Y axis of the recognised object (in the user's coordinate system).
26 27	-132 0	Decimal Decimal	-132		Rotation Z	Rotation about the Z axis of the recognised object (in the user's coordinate system).
28 29	02 0	Decimal Decimal	02		Current layer	Current pallet layer, starting with "0". An empty layer is marked with "0".
30 31	1 0	Binary Binary	1		Slip sheet	There is a slip sheet on a pallet layer: 0 = no slip sheet detected 1 = slip sheet detected
32 33	098 0	Decimal Decimal	098		Error	Error: 0 = no error 1 = undefined error 2 = unexpected object recognised Other error codes: (→ 13.1.5).
34 35	00 0	Binary Binary	00		Collision-free	Collision-free depalletising: 0: no 1: yes
36 37	1 0	Decimal Decimal	1		Quality	Quality of object recognition between 0 and 100. The value "100" stands for best possible quality.

The incorrect execution of a command leads to the following status:

- Error bit = 1
- Duplicated command word is displayed
- Asynchronous message bit = 0
- Asynchronous message identification = 0
- Message counter increments by 1

9.3.14 Transmit process values of depalletising via PROFINET

The device can transmit the process values to a PLC via the PROFINET fieldbus.

Only one fieldbus can be active at a time. The fieldbus is adjustable (→ software manual).

In the output string the process values are separated by a semicolon. The output string is transmitted to a PLC in the displayed sequence.

Observe the following notes to transmit the output string to a PLC:

- Bytes 0 to 7 are part of the output string. They are not displayed in the ifm Vision Assistant.
- Semicolons ";" in the output string are not transmitted.
- Float values are converted into 16-bit integers before the transmission.
- All numerical values are converted into 16-bit integers before the transmission.

The output string is as follows:

1;0.200;0.150;0.307;+00.002;-10.044;+03.100;+170;-133;-132;02;1;098;00;1

Byte no.	Data	Encoding	Process value	Unit	Description	Comment
0 1	2#0010_0000 2#0000_0000	Binary Binary	0.5		Duplicated command word	• Bit 0.5 indicates a successful trigger command.
2 3	2#0000_0000 2#0000_0000	Binary Binary			Synchronous / asynchronous message identification	
4 5	2#0000_0000 2#0000_0011	Binary Binary	3		Message counter	• The device has received 3 messages. • Increments by 1 with each action (trigger, message sent etc.).
6 7	2#0000_0000 2#0000_0000	Binary Binary			Reserved	
8 9	1 0	Binary Binary	1		Object found	0 = no object found 1 = object found
10 11	200 0	Decimal Decimal	200	mm	Width	The broadest dimension of the object surface.
12 13	150 0	Decimal Decimal	150	mm	Height	The object height relative to the base plate.
14 15	307 0	Decimal Decimal	307	mm	Length	The longest dimension of the object surface.
16 17	2 0	Decimal Decimal	+2		Centre point X	The X coordinate of the centre point of the object surface (in the user's coordinate system).
18 19	10044 0	Decimal Decimal	-10044		Centre point Y	The Y coordinate of the centre point of the object surface (in the user's coordinate system).
20 21	3100 0	Decimal Decimal	+3100		Centre point Z	The Z coordinate of the centre point of the object surface (in the user's coordinate system).
22 23	170 0	Decimal Decimal	+170		Rotation X	Rotation about the X axis of the recognised object (in the user's coordinate system).

Byte no.	Data	Encoding	Process value	Unit	Description	Comment
24 25	-133 0	Decimal Decimal	-133		Rotation Y	Rotation about the Y axis of the recognised object (in the user's coordinate system).
26 27	-132 0	Decimal Decimal	-132		Rotation Z	Rotation about the Z axis of the recognised object (in the user's coordinate system).
28 29	02 0	Decimal Decimal	02		Current layer	Current pallet layer, starting with "0". An empty layer is marked with "0".
30 31	1 0	Binary Binary	1		Slip sheet	There is a slip sheet on a pallet layer: 0 = no slip sheet detected 1 = slip sheet detected
32 33	098 0	Decimal Decimal	098		Error	Error: 0 = no error 1 = undefined error 2 = unexpected object recognised Other error codes: (→ 13.1.5).
34 35	00 0	Binary Binary	00		Collision-free	Collision-free depalletising: 0: no 1: yes
36 37	1 0	Decimal Decimal	1		Quality	Quality of object recognition between 0 and 100. The value "100" stands for best possible quality.

The incorrect execution of a command leads to the following status:

- Error bit = 1
- Duplicated command word is displayed
- Asynchronous message bit = 0
- Asynchronous message identification = 0
- Message counter increments by 1

9.3.15 Transmit process values of depalletising via TCP/IP

The device can transmit the process values to a PLC via the TCP/IP protocol. In the ifm Vision Assistant the process values are displayed as output string as shown below:

```
star;1;0.200;0.150;0.307;+00.002;-10.044;
+03.100;+170;-133;-132;02;1;098;00;1;stop
```

In the output string the process values are separated by a semicolon. The output string is transmitted to a PLC in the displayed sequence.

Observe the following notes to transmit the output string to a PLC:

- Semicolons ";" in the output string are not transmitted.
- All numerical values are converted into 16-bit integers before the transmission.

The output string is as follows (data type: ASCII):

star;1;0.200;0.150;0.307;+00.002;-10.044;+03.100;+170;-133;-132;02;1;098;00;1;stop

Process value	Unit	Description
star		Start string
1		1 = no object found 0 = object found
0.200		Width
0.150		Height
0.307		Length
+00.002		Centre point X
-10.044		Centre point Y
+03.100		Centre point Z
+170		Rotation X
-133		Rotation Y
-132		Rotation Z
02		Current layer
1		0 = no slip sheet detected 1 = slip sheet detected
098		Error
00		0 = no collision-free depalletising 1 = collision-free depalletising
1		Quality of object recognition (0 to 100).
stop		Stop string

10. Maintenance, repair and disposal

Observe the following instructions:

► Do not open the device as it does not contain any components which can be maintained by the user. The device must only be repaired by the manufacturer.

► Dispose of the sensor in accordance with the national environmental regulations.

10.1 Clean

Observe the following instructions before cleaning the sensor:

► Use clean and lint-free cloth.

► Use glass cleaner as cleaning agent.

If the instructions are not observed, scratches on the lens window may cause measurement errors.

10.2 Update firmware

With the software ifm Vision Assistant the firmware of the sensor can be updated.

Parameters saved in the sensor get lost by the firmware update. Create a backup copy of the parameters before updating the firmware:

► Before updating the firmware export parameters.

► Import parameters after updating the firmware.

Firmware updates are available on our website: www.ifm.com

10.3 Replace device

The parameters are lost when a device is replaced. Create a backup copy of the parameters before replacing the device:

► Export the parameters of the old device before replacement.

► Import the parameters into the new device after replacement.

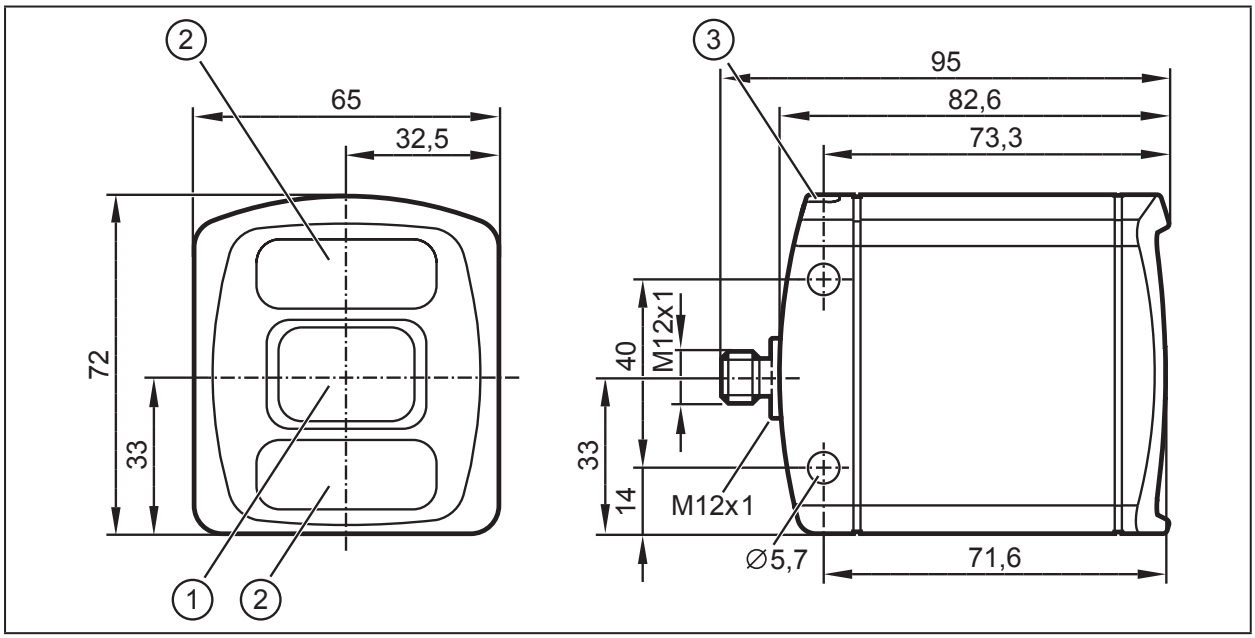
With the export and import of parameters several devices can be quickly provided with the same parameters.

11. Approvals/standards

The EU declaration of conformity is available at: www.ifm.com

12. Scale drawings

12.1 O3D302 / O3D312

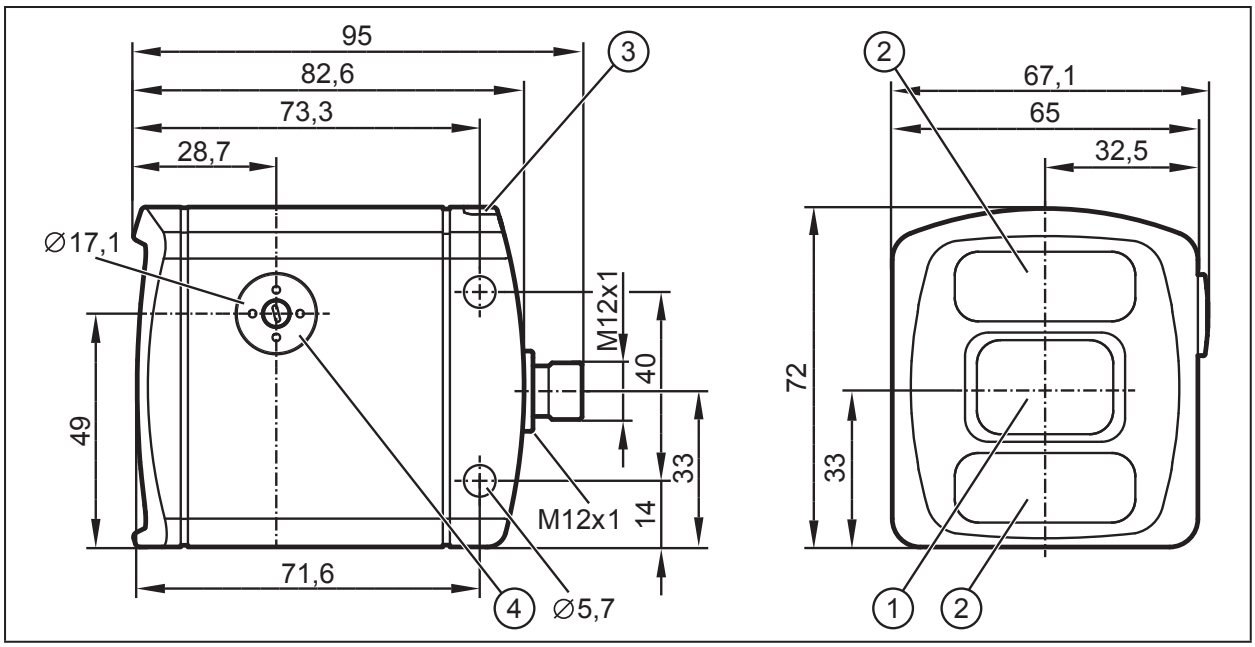


① Lens

② Illumination unit

③ LED 2 colours (yellow/green)

12.2 O3D300 / O3D310



① Lens

② Illumination unit

③ LED 2 colours (yellow/green)

④ Focal setter

13. Appendix

13.1 Process Interface

The process interface is used during the normal operation mode to get operational data (e.g. 3D images, process values) from the O3D3xx.

13.1.1 Sending Commands

For sending commands via the process interface the commands have to be sent with a special protocol and as ASCII character strings. This protocol conforms to the version 3 of the O2V/O2D products.

Structure of the protocol:

<Ticket><length>CR LF <Ticket><content>CR LF

Abbreviation	Description	ASCII code (dec)	ASCII code (hex)
CR	Carriage Return	13	D
LF	Linefeed	10	A
< >	Marking of a placeholder (e.g. <code> is a placeholder for code)		
[ ]	Optional argument (possible but not required)		

Command	Description
<content>	It is the command to the device (e.g. trigger the unit).
<ticket>	It is a character string of 4 digits between 0-9. If a message with a specific ticket is sent to the device, it will reply with the same ticket. A ticket number must be > 0999. Use a ticket number from the range 1000 - 9999.
<length>	It is a character string beginning with the letter 'L' followed by 9 digits. It indicates the length of the following data (<ticket><content>CR LF) in bytes.

They are different protocol versions available:

Version	Input format	Output format
V1	<Content>CR LF	As input
V2	<Ticket><Content>CR LF	As input
V3	<Ticket><Length>CR LF<Ticket><Content>CR LF	As input
V4	<Content>CR LF	<length>CR LF<Content>CR LF

The default protocol version is "V3". It is recommended to use protocol version 3 for machine to machine communication. This is due to the fact that only version 3 supports asynchronous messages and provides length information.

Ticket numbers for asynchronous messages:

Ticket number	Description
0000	Asynchronous results
0001	Asynchronous error messages / codes
0010	Asynchronous notifications / message codes

Format of asynchronous notifications

The format of the asynchronous notifications is a combination of the unique message ID and a JSON formatted string containing the notification details: <unique message ID>:<JSON content>

Example for protocol version 3:

<ticket=0010>L<length>CR LF<ticket=0010><unique message ID>:<JSON content>CR LF

Result:

0010L000000045\r\n0010000500000:{"ID": 1034160761,"Index":1,"Name": "Pos 1"}\r\n

Explanation of the result:

Command	Result
<ticket=0010>	0010
L<length>	L000000045
CR LF	\r\n
<ticket=0010>	0010
<unique message ID>	000500000
<JSON content>	{"ID": 1034160761,"Index":1,"Name": "Pos 1"}
CR LF	\r\n

Asynchronous message IDs

Asynchronous message ID	Description	Example	Description
000500000	Application changed	{"ID": 1034160761,"Index":1,"Name": "Pos 1","valid":true}	
000500001	Application is not valid	{"ID": 1034160761,"Index":1,"Name": "Pos 1","valid":false}	If a application exists on given index but it is invalid, the ID and Name are filled accoring to the application. If there is no application on given index, the application ID will contain 0 and the name an empty string "".
000500002	image acquisition finished	{}	This message signals the reciever, that the device has finished the image acquistion. This can be used for cascading multiple devices with a software trigger.

13.1.2 Receiving Images

For receiving the image data a TCP/IP socket communication is established. The default port number is 50010. The port number may differ based on the configuration. After opening the socket communication, the O3D3XX device will automatically (if the device is in free run mode) send the data through this socket to the TCP/IP client (PC).

PCIC output per frame. The following data is submitted in this sequence:

Component	Content
Ticket and length information	(→ 13.2.15)
Ticket	"0000"
Start sequence	String "star" (4 bytes)
Normalised amplitude image Output format: 16-bit unsigned integer	1 image
Distance image Output format: 16-bit unsigned integer. Unit: mm	1 image
X image Output format: 16-bit signed integer. Unit: mm	1 image
Y image Output format: 16-bit signed integer. Unit: mm	1 image
Z image Output format: 16-bit signed integer. Unit: mm	1 image
Confidence image Output format: 8-bit unsigned integer	1 image
Diagnostic data	
Stop sequence	String "stop" (4 bytes)
Ticket signature	<CR><LF>

13.1.3 Image data

For every image there will be a separate chunk. The chunk is part of the response frame data of the process interface.

The header of each chunk contains different kinds of information. This information is separated into bytes. The information contains e.g. the kind of image which will be in the “PIXEL_DATA” and the size of the chunk.

Offset	Name	Description	Size [byte]
0x0000	CHUNK_TYPE	Defines the type of the chunk. For each distinct chunk an own type is defined.	4
0x0004	CHUNK_SIZE	Size of the whole image chunk in bytes. After this count of bytes the next chunk starts.	4
0x0008	HEADER_SIZE	Number of bytes starting from 0x0000 until PIXEL_DATA.	4
0x000C	HEADER_VERSION	Version number of the header	4
0x0010	IMAGE_WIDTH	Image width in pixel	4
0x0014	IMAGE_HEIGTH	Image height in pixel	4

Offset	Name	Description	Size [byte]
0x0018	PIXEL_FORMAT	Pixel format	4
0x001C	TIME_STAMP	Time stamp in microseconds (deprecated)	4
0x0020	FRAME_COUNT	Frame counter	4
0x0024	STATUS_CODE	Errors of the device	4
0x0028	TIME_STAMP_SEC	Time stamp in seconds	4
0x002C	TIME_STAMP_NSEC	Time stamp in nanoseconds	4
0x0030	PIXEL_DATA	The pixel data in the given type and dimension of the image. Padded to 4-byte boundary.	4

Available chunk types:

Constant	Value	Description
RADIAL_DISTANCE_IMAGE	100	Each pixel of the distance matrix denotes the ToF distance measured by the corresponding pixel or group of pixels of the imager. The distance value is corrected by the camera's calibration, excluding effects caused by multipath and multiple objects contributions (e.g. "flying pixels"). Reference point is the optical centre of the camera inside the camera housing. Invalid PMD pixels (e.g. due to saturation) have a value of zero. Data type: 16-bit unsigned integer (little endian) Unit: millimetres
NORM_AMPLITUDE_IMAGE	101	Each pixel of the normalized amplitude image denotes the raw amplitude (see amplitude image below for further explanation), normalized to exposure time. Furthermore, vignetting effects are compensated, ie the darkening of pixels at the image border is corrected. The visual impression of this grayscale image is comparable to that of a common 2D camera. Invalid PMD pixels (e.g. due to saturation) have an amplitude value of 0. Data type: 16-bit unsigned integer
AMPLITUDE_IMAGE	103	Each pixel of the amplitude matrix denotes the amount of modulated light (i.e. the light from the camera's active illumination) which is reflected by the appropriate object. Higher values indicate higher PMD signal strengths and thus a lower amount of noise on the corresponding distance measurements. The amplitude value is directly derived from the PMD phase measurements without normalisation to exposure time. In multiple exposure mode, the lack of normalisation may lead (depending on the chosen exposure times) to inhomogeneous amplitude image impression, if a certain pixel is taken from the short exposure time and some of its neighbours are not. Invalid PMD pixels (e.g. due to saturation) have an amplitude value of 0. Data type: 16-bit unsigned integer
GRAYSCALE_IMAGE	104	Each pixel of the amplitude matrix denotes the amount of modulated light which is reflected by the appropriate object (i.e. the light from the camera's active illumination). Higher values indicate higher PMD signal strengths and thus a lower amount of noise on the corresponding distance measurements. The amplitude value is directly derived from the PMD phase measurements without normalisation to exposure time.

Constant	Value	Description
CARTESIAN_X_ COMPONENT	200	The X matrix denotes the X component of the Cartesian coordinate of a PMD 3D measurement. The origin of the camera's coordinate system is in the middle of the lens' front glass, if the extrinsic parameters are all set to 0. Data type: 16-bit signed integer Unit: millimetres
CARTESIAN_Y_ COMPONENT	201	The Y matrix denotes the Y component of the Cartesian coordinate of a PMD 3D measurement. The origin of the camera's coordinate system is in the middle of the lens' front glass, if the extrinsic parameters are all set to 0. Data type: 16-bit signed integer Unit: millimetres
CARTESIAN_Z_ COMPONENT	202	The Z matrix denotes the Z component of the Cartesian coordinate of a PMD 3D measurement. The origin of the camera's coordinate system is in the middle of the lens' front glass, if the extrinsic parameters are all set to 0. Data type: 16-bit signed integer Unit: millimetres
CARTESIAN_ALL	203	CARTESIAN_X_COMPONENT, CARTESIAN_Y_COMPONENT, CARTESIAN_Z_COMPONENT
UNIT_VECTOR_ALL	223	The unit vector matrix contains 3 values [ex, ey, ez] for each PMD pixel, i.e. the data layout is [ex_1,ey_1,ez_1, ... ex_N, ey_N, ez_N], where N is the number of PMD pixels. Data type: 32-bit floating point number (3x per pixel)
CONFIDENCE_IMAGE	300	See Additional Information for Image Data (→ 13.1.4)
DIAGNOSTIC	302	See Receiving Images (→ 13.1.2)

JSON_DIAGNOSTIC	305	Items with JSON formatted diagnostic data is formated like this:
		`{`
		`"AcquisitionDuration": 20.391,`
		`"EvaluationDuration": 37.728,`
		`"FrameDuration": 37.728,`
		`"FrameRate": 15.202,`
		`"TemperatureIllu": 52.9`
		`}`
		Unit for durations: millimetres
		Unit for framerates: Hz
		Unit for temperature: °C

Constant	Value	Description
EXTRINSIC_CALIB	400	The transformation from one cartesian coordinate system to another is defined by a 6 degrees of freedom vector (DOF): [trans_x, trans_y, trans_z, rot_x, rot_y, rot_z]. Let R be the product of the common "clockwise" 3D-rotation matrices: R = Rx*Ry*Rz The transformation of a point P is specified by P_t = R*P + [trans_x, trans_y, trans_z]'. The device extrinisic calibration can be set by the user, but it may be changed by an automatic calibration feature of the device. Data type: 32-bit floating point number (little endian) Unit for trans_x, trans_y, trans_z: millimetres Unit for rot_x, rot_y, rot_z: °
JSON_MODEL	500	Model data in JSON
MODEL_ROIMASK	501	ROI mask for internal debugging purposes
SNAPSHOT_IMAGE	600	Snapshot image

Pixel format:

Constant	Value	Description
FORMAT_8U	0	8-bit unsigned integer
FORMAT_8S	1	8-bit signed integer
FORMAT_16U	2	16-bit unsigned integer
FORMAT_16S	3	16-bit signed integer
FORMAT_32U	4	32-bit unsigned integer
FORMAT_32S	5	32-bit signed integer
FORMAT_32F	6	32-bit floating point number
FORMAT_64U	7	64-bit unsigned integer
FORMAT_64F	8	64-bit floating point number
Reserved	9	N/A
FORMAT_32F_3	10	Vector with 3x32-bit floating point number

13.1.4 Additional Information for CONFIDENCE_IMAGE

Further information for the confidence image:

Bit	Value	Description
0	**1** = pixel invalid	Pixel invalid The pixel is invalid. To determine whether a pixel is valid or not only this bit needs to be checked. The reason why the bit is invalid is recorded in the other confidence bits.
1	**1** = pixel saturated	Pixel is saturated Contributes to pixel validity: yes
2	**1** = bad A-B symmetry	A-B pixel symmetry The A-B symmetry value of the four phase measurements is above threshold. Remark: This symmetry value is used to detect motion artefacts. Noise (e.g. due to strong ambient light or very short integration times) or PMD interference may also contribute. Contributes to pixel validity: yes
3	**1** = amplitude below minimum amplitude threshold	Amplitude limits The amplitude value is below minimum amplitude threshold. Contributes to pixel validity: yes
4+5	**Bit 5, bit 4** **0 0** = unused **0 1** = shortest exposure time (only used in 3 exposure mode) **1 0** = middle exposure time in 3 exposure mode, short exposure in double exposure mode **1 1** = longest exposure time (always 1 in single exposure mode)	Exposure time indicator The two bits indicate which exposure time was used in a multiple exposure measurement. Contributes to pixel validity: no
6	1 = pixel is clipped	Clipping box on 3D data If clipping is active this bit indicates that the pixel coordinates are outside the defined volume. Contributes to pixel validity: yes
7	1 = suspect/defective pixel	Suspect pixel This pixel has been marked as "suspect" or "defective" and values have been replaced by interpolated values from the surroundings. Contributes to pixel validity: no

13.1.5 Configuration of PCIC Output

The user has the possibility to define his own PCIC output. This configuration is only valid for the current PCIC connection. It does not affect any other connection and will get lost after disconnecting.

For configuring the PCIC output a “flexible” layouter concept is used, represented by a JSON string. The format of the default configuration is as follows:

```
{
  "layouter": "flexible",
  "format": { "dataencoding": "ascii" },
  "elements": [
    { "type": "string", "value": "star", "id": "start_string" },
    { "type": "blob", "id": "normalized_amplitude_image" },
    { "type": "blob", "id": "x_image" },
    { "type": "blob", "id": "y_image" },
    { "type": "blob", "id": "z_image" },
    { "type": "blob", "id": "confidence_image" },
    { "type": "blob", "id": "diagnostic_data" },
    { "type": "string", "value": "stop", "id": "end_string" }
  ]
}
```

This string can be retrieved by the C? command, altered and sent back using the c command.

The layout software has the following main object properties:

Name	Description	Details
layouter	Defines the basic data output format. So far only “flexible” is supported	Type: string
format	Defines format details, the definitions in the main object are the defaults for any of the following data elements (e.g. if it says dataencoding=binary, all data elements should be binary encoded instead of ASCII).	Type: object
elements	List of data elements which must be written.	Type: array of objects

The actual data is defined within the “elements” properties and may consist of these settings:

Name	Description	Details
type	Defines the type of data which must be written. The data might be stored in a different type (e.g. stored as integer but should be output as Float32) The type "records" will need some special handling.	Type: string
id	Defines an identifier for this data element. If there is no fixed value (property "value"), the data should be retrieved via id.	Type: string
value	Optional property for defining a fixed output value.	Type: any JSON value
format	Type-depending option for fine-tuning the output format. E.g. cut an integer to less than 4 bytes.	Type: object

Available values for the type property:

Type	Description
records	Defines that this element represents a list of records. If type is set to "records", there must be an "elements" property. The "elements" property defines which data should be written per record.
string	Data is written as string. Most of the time this will be used with "value" property to write fixed start, end or delimiter text. Text encoding should be UTF8 if there is nothing else specified in format properties.
float32	Data is written as floating point number. This has a lot of formatting options (at least with "flexible" layout software) See following section about format properties.
uint32	Data is written as integer. This has a lot of formatting options (at least with "flexible" layout software) See following section about format properties.
int32	Data is written as integer. This has a lot of formatting options (at least with "flexible" layout software) See following section about format properties.
uint16	Limits the output to two bytes in binary encoding, besides the binary limitation it acts like uint32.
int16	Limits the output to two bytes in binary encoding, besides the binary limitation it acts like int32.
uint8	Limits the output to one byte in binary encoding, besides the binary limitation it acts like uint32.
int8	Limits the output to one byte in binary encoding, besides the binary limitation it acts like int32.
blob	Data is written as a BLOB (byte by byte as if it came from the data provider). (Binary Large Object)

Depending on the desired data format the user may tune his output data with further “format” properties.

Common format properties:

Format properties	Allowed values	Default
dataencoding	"ascii" or "binary" can be defined in top-level-object and overwritten by element objects.	"ascii"
scale	"float value with decimal separator" to scale the results for output byte width	1.0
offset	"float value with decimal separator"	0.0

Binary format properties:

Format properties	Allowed values	Default
order	Little, big and network	Little

ASCII format properties:

Format properties	Allowed values	Default
width	Output width. If the resulting value exceeds the width field the result will not be truncated.	0
fill	Fill character	" "
precision	Precision is the number of digits behind the decimalseparator.	6
displayformat	Fixed, scientific	Fixed
alignment	Left, right	Right
decimalseparator	7-bit characters for e.g. "."	"."
base	Defines if the output should be: ● binary (2) ● octal (8) ● decimal (10) ● hexadecimal (16)	10

Example of a format configuration of the temperature (id: temp_illu) element.

1. Illumination temperature like this "33,5___":

```
c000000226{ "layouter": "flexible", "format": { "dataencoding": "ascii" },
"elements": [ { "type": "float32", "id": "temp_illu", "format": { "width": 7,
"precision": 1, "fill": "_",  "alignment": "left",  "decimalseparator": "," }
} ] }
```

2. Illumination temperature as binary (16-bit integer, 1/10 °C):

```
c000000194{ "layouter": "flexible", "format": { "dataencoding": "ascii"
}, "elements": [ { "type": "int16", "id": "temp_illu", "format": {
"dataencoding": "binary", "order": "network", "scale": 10 } } ] }
```

3. Illumination temperature in °F (e.g. "92.3 Fahrenheit"):

```
c000000227{ "layouter": "flexible", "format": { "dataencoding": "ascii" },
"elements": [ { "type": "float32", "id": "temp_illu", "format": { "precision":
1, "scale": 1.8, "offset": 32 } }, { "type": "string", "value": " Fahrenheit"
} ] }
```

The following element IDs are available:

ID	Description	Native data type
activeapp_id	Active application, shows which of the 32 application-configurations is currently active	32-bit unsigned integer
all_cartesian_vector_matrices	All Cartesian images (X+Y+Z) concatenated to one package	16-bit signed integer
all_unit_vector_matrices	Matrix of unit vectors. Each element consists of a 3 component vector [e_x, e_y, e_z]	Float32
amplitude_image	PMD raw amplitude image	16-bit unsigned integer
confidence_image	Confidence image	8-bit unsigned integer
distance_image	Radial distance image	16-bit unsigned integer unit: millimetres
evaltime	Evaluation time for current frame in milliseconds	32-bit unsigned integer
extrinsic_calibration	Extrinsic calibration, constisting of 3 translation parameters (unit: millimeters) and 3 angles (unit: degree): [t_x, t_y, t_z, alpha_x, alpha_y, alpha_z]	Float32
framerate	Current frame rate in Hz	Float32
normalized_amplitude_image	Normalized amplitude image	16-bit unsigned integer
temp_front1	Invalid temperature, the output is 3276.7	Float32, unit: °C
temp_illu	Temperature measured in the device while capturing this result Measured on the illumination board	Float32, unit: °C
x_image y_image z_image	Cartesian coordinates for each pixel Each dimension is a separate image	16-bit signed integer

For completeness, level, distance and dimensioning application the following IDs are available:

ID	Description	Native data type
id	ID of the model	int32
rois.count	Number of records in "roi"	int32
rois	List of all ROIs (ROIgroup) of this model	records
SP1 SP2	SwitchingPoint1 and 2 if the model is a Level- or Distance-type. If it is not a Level-/Distance-type, it shall output a null-value.	float32
boxFound	These results are available for a dimensioning application. If the model is not oft the type dimensioning, the IDs shall output a null-value.	int8
length		float
width		float
height		float
qualityLength		float
qualityWidth		float
qualityHeight		float
xMidTop		float
yMidTop		float
zMidTop		float
yawAngle		float
backgroundPlaneDistance		float
numGood	These results are available for a completeness, level and distance applications. If the model is not oft one of these types, the IDs shall output a null-value.	int
numUnderSP1		int
numOverSP2		int
numInvalid		int
allROIsGood		bool
anchorFound		bool
hasAnchorTracking		bool

For ROIs of completeness, level or distance application the following IDs are available:

ID	Description	Native data type
id	unique ID of the ROI within the Model	int32
procval	per ROI process value	float 32Bit
state	per ROI state (if ROI procval is valid or not) • ROI_PROCESS_VALUE_VALID = 0 • ROI_PROCESS_VALUE_REFIMAGE_SET_NOT_TEACHED = 1 • ROI_PROCESS_VALUE_TEACHING_FAILED = 2 • ROI_PROCESS_VALUE_REFIMAGE_INVALID = 3 • ROI_PROCESS_VALUE_NO_VALID_PIXEL = 4 • ROI_PROCESS_VALUE_REFIMAGE_NO_VALID_PIXEL = 5 • ROI_PROCESS_VALUE_OVERFILL = 6 • ROI_PROCESS_VALUE_UNDERFILL = 7	uint32
quality	0..1	float32

For the main object on devices with statistics feature the following IDs are available:

ID	Description	Native data type
statistics_overall_count	Allows the user to output the statistics value with the result of the frame, maps to ModelResults: adv_statistics.number_of_frames	uint32
statistics_passed_count	Allows the user to output the statistics value with the result of the frame, maps to ModelResults: adv_statistics.number_of_passed_frames	uint32
statistics_failed_count	Allows the user to output the statistics value with the result of the frame, maps to ModelResults: adv_statistics.number_of_failed_frames	uint32
statistics_aborted_count	Allows the user to output the statistics value with the result of the frame, maps to ModelResults: adv_statistics.number_of_aborted_frames	uint32
statistics_acquisition_time_min	Allows the user to output the statistics value with the result of the frame,maps to ModelResults: adv_statistics.frame_acquisition.min	float32
statistics_acquisition_time_mean	Allows the user to output the statistics value with the result of the frame,maps to ModelResults: adv_statistics.frame_acquisition.mean	float32
statistics_acquisition_time_max	Allows the user to output the statistics value with the result of the frame,maps to ModelResults: adv_statistics.frame_acquisition.max	float32
statistics_evaluation_time_min	Allows the user to output the statistics value with the result of the frame,maps to ModelResults: adv_statistics.frame_evaluation.min	float32
statistics_evaluation_time_mean	Allows the user to output the statistics value with the result of the frame,maps to ModelResults: adv_statistics.frame_evaluation.mean	float32
statistics_evaluation_time_max	Allows the user to output the statistics value with the result of the frame,maps to ModelResults: adv_statistics.frame_evaluation.max	float32
statistics_frame_duration_min	Allows the user to output the statistics value with the result of the frame,maps to ModelResults: adv_statistics.frame_duration.min	float32
statistics_frame_duration_mean	Allows the user to output the statistics value with the result of the frame,maps to ModelResults: adv_statistics.frame_duration.mean	float32
statistics_frame_duration_max	Allows the user to output the statistics value with the result of the frame,maps to ModelResults: adv_statistics.frame_duration.max	float32

For model records of type "DimensioningV2" (Robot Pick & Place) the following IDs are available:

Length values are given in unit [m].

Rotation values are given in unit [°].

ID	Description	Native data type
numberOfObjects	Number of found objects.	uint32
numberOfObjectCandidates	Number of found object candidates that have been inspected.	uint32
error	Dimensioning error: 0: no error 1: undefined error 2: no object found	uint32
maximumNumberOfObjectsToMeasure	Maximum number of objects to measure.	uint32
objectGeometry	Geometry type of object: 0: Box 1: Circle 2: Ellipse	uint32
objects[maximumNumberOfObjectsToMeasure] {	This structure is provided for each object defined by maximumNumberOfObjectsToMeasure. If not all objects have been found, the values are also provided for the number of missing objects.	
objectFound	Object can be successfully measured (0 if false, 1 if true).	uint32
length	Object length is the longest dimension of the object.	float32
width	Object width is the shortest dimension of the object.	float32
height	Object height is the object height relative to the ground plane.	float32
xMidTop	Cartesian X coordinates of middle point on the top surface of the detected object.	float32
yMidTop	Cartesian Y coordinates of middle point on the top surface of the detected object.	float32
zMidTop	Cartesian Z coordinates of middle point on the top surface of the detected object.	float32
yawAngle	Yaw angle is defined as the angle between the world coordinate x-axis and the vector along the object "length".	float32
circleThickness	The thickness of the circle.	float32
centerPointX	X coordinate of the top center point from the detected object (user frame coordinate system).	float32
centerPointY	Y coordinate of the top center point from the detected object (user frame coordinate system).	float32
centerPointZ	Z coordinate of the top center point from the detected object (user frame coordinate system).	float32
rotationX	X rotation of the detected object (user frame coordinate system).	float32
rotationY	Y rotation of the detected object (user frame coordinate system).	float32
rotationZ	Z rotation of the detected object (user frame coordinate system).	float32
}		

ID	Description	Native data type
	For compatibility reasons the following values are provided for the first detected object.	
boxFound	Object can be successfully measured (0 if false, 1 if true).	uint32
length	Object length is the longest dimension of the object.	float32
width	Object width is the shortest dimension of the object.	float32
height	Object height is the object height relative to the ground plane.	float32
xMidTop	Cartesian X coordinates of middle point on the top surface of the detected object.	float32
yMidTop	Cartesian Y coordinates of middle point on the top surface of the detected object.	float32
zMidTop	Cartesian Z coordinates of middle point on the top surface of the detected object.	float32
yawAngle	Yaw angle is defined as the angle between the world coordinate x-axis and the vector along the object "length".	float32
circleThickness	The thickness of the circle.	float32
centerPointX	X coordinate of the top center point from the detected object (user frame coordinate system).	float32
centerPointY	Y coordinate of the top center point from the detected object (user frame coordinate system).	float32
centerPointZ	Z coordinate of the top center point from the detected object (user frame coordinate system).	float32
rotationX	X rotation of the detected object (user frame coordinate system).	float32
rotationY	Y rotation of the detected object (user frame coordinate system).	float32
rotationZ	Z rotation of the detected object (user frame coordinate system).	float32
backgroundPlaneDistance	Distance of the background at background teach.	float32
objectType	Type of the detected object: 1: box 2: true bounding box 3: circle 4: enclosing circle 5: ellipse 6: enclosing ellipse	uint32
UFCThreeMarkerTeach["A"..."C"]	Coordinates of the available UFC markers.	
{		
x		float32
y		float32
}		

For model records of type "Depalletizing" the following IDs are available:

Length values are given in unit [m].

Rotation values are given in unit [°].

<table>
<tr><th>ID</th><th>Description</th><th>Native data type</th></tr>
<tr><td>error</td><td>Errors in the algorithm:
<table>
<tr><th>Value</th><th>Name</th><th>Description</th></tr>
<tr><td>0</td><td>Depalletizing_Error_None</td><td>No error detected.</td></tr>
<tr><td>1</td><td>Depalletizing_Error_Unknown</td><td>Unknown error detected.</td></tr>
<tr><td>2</td><td>Depalletizing_Error_ UnexpectedObject</td><td>Unexpected object detected.</td></tr>
<tr><td>3</td><td>Depalletizing_Error_StackEmpty</td><td>Stack is empty.</td></tr>
<tr><td>4</td><td>Depalletizing_Error_NoObjectSizes</td><td>No box dimensions provided on the input.</td></tr>
<tr><td>5</td><td>Depalletizing_Error_NoObjectMatch</td><td>No matching object found.</td></tr>
<tr><td>6</td><td>Depalletizing_Error_DataInvalid</td><td>Too many pixels are invalid.</td></tr>
<tr><td>7</td><td>Depalletizing_Error_Background TeachingStatus_ErrorNot EnoughValidPixels</td><td>Background estimation only: not enough valid pixels.</td></tr>
<tr><td>8</td><td>Depalletizing_Error_Background TeachingStatus_ErrorStdTooHigh</td><td>Background estimation only: standard deviation too high.</td></tr>
<tr><td>9</td><td>Depalletizing_Error_Background TeachingStatus_ErrorPlaneFitFailed</td><td>Background estimation only: estimation of the plane failed.</td></tr>
<tr><td>10</td><td>Depalletizing_Error_Background TeachingStatus_ErrorPlane AngleTooHigh</td><td>Background estimation only: plane angle too high.</td></tr>
<tr><td>11</td><td>Depalletizing_Error_Background TeachingStatus_ErrorRotation CalculationFailed</td><td>Background estimation only: internal numerical error in calculation of rotation.</td></tr>
<tr><td>12</td><td>Depalletizing_Error_ InvalidReferenceTeach</td><td>Invalid background teach.</td></tr>
<tr><td>13</td><td>Depalletizing_Error_ InvalidVOITeach</td><td>Invalid VOI teach.</td></tr>
<tr><td>14</td><td>Depalletizing_Error_Insufficient MarginToImageBorder</td><td>Not enough space between segmented layer and image border.</td></tr>
<tr><td>15</td><td>Depalletizing_Error_ IncorrectObjectSizes</td><td>Provided box dimensions are invalid.</td></tr>
<tr><td>16</td><td>Depalletizing_Error_Underfill</td><td>Measurements are below the background level.</td></tr>
</table></td><td>uint32</td></tr>
<tr><td>objectFound</td><td>Object can be successfully measured (0 if false, 1 if true).</td><td>uint32</td></tr>
<tr><td>objectQuality</td><td>Quality of the object detection between 0 and 100.</td><td>float32</td></tr>
<tr><td>objectLength</td><td>Object length is the longest dimension of the top surface of the object.</td><td>float32</td></tr>
<tr><td>objectWidth</td><td>Object width is the shortest dimension of the top surface of the object.</td><td>float32</td></tr>
<tr><td>objectHeight</td><td>Object height is the object height relative to the ground plane.</td><td>float32</td></tr>
<tr><td>centerPointX</td><td>X coordinate of the top center point from the detected object (user frame coordinate system).</td><td>float32</td></tr>
<tr><td>centerPointY</td><td>Y coordinate of the top center point from the detected object (user frame coordinate system).</td><td>float32</td></tr>
<tr><td>centerPointZ</td><td>Z coordinate of the top center point from the detected object (user frame coordinate system).</td><td>float32</td></tr>
<tr><td>rotationX</td><td>X rotation of the detected object (user frame coordinate system).</td><td>float32</td></tr>
<tr><td>rotationY</td><td>Y rotation of the detected object (user frame coordinate system).</td><td>float32</td></tr>
</table>

ID	Description	Native data type
rotationZ	Z rotation of the detected object (user frame coordinate system).	float32
layerLevel	Current pallet layer for depalletization, starting with "0". An empty stack is indicated by "0".	uint32
sensorMountingHeight	Recommended height of the sensor above the palette. Values "<=0": invalid input parameters (e.g. invalid palette dimensions).	float32
isSlipSheet	A slipsheet is on top of the stack: 0: no 1: yes	uint32
backgroundPlaneDistance	Distance of the background plane in positive direction of the z-axis.	float32
isCollisionFree	Collision free depalletization: 0: false 1: true	uint32
centerPoint2DX	The top center X coordinate of the detected box object (projected into the 2D image).	float32
centerPoint2DY	The top center Y coordinate of the detected box object (projected into the 2D image).	float32

The following IDs can be changed with the f command (→ 13.2.6):

ID	Name	Description	Values
0000000001	DepalSlipSheetDetection	Depalletizing: slip sheet detection on/off	1/0
0000000002	DepalSlipObjectType	Depalletizing: type of the object to be detected	0: box 1: bag
0000000003	DepalWidth	Depalletizing: width of the objects to be detected	mm
0000000004	DepalHeight	Depalletizing: length of the objects to be detected	mm
0000000005	DepalLength	Depalletizing: heigth of the objects to be detected	mm

13.2 Process Interface Command Reference

All received messages which are sent because of the following commands will be sent without “start”/”stop” at the beginning or ending of the string.

13.2.1 a Command (activate application)

Command	a<application number>	
Description	Activates the selected application	
Type	Action	
Reply	*	
	!	• Application not available • <application number> contains wrong value • External application switching activated • Device is in an invalid state for this command, e.g. configuration mode
	?	Invalid command length
Note	<application number> 2 digits for the application number as decimal value	

13.2.2 A? Command (occupancy of application list)

Command	A?	
Description	Requests the occupancy of the application list	
Type	Request	
Reply	<amount><t><number active application><t> ... <number><t><number>	
	?	Invalid command length
	!	Invalid state (e.g. no application active)
Note	<amount> char string with 3 digits for the amount of applications saved on the device as decimal number <t> tabulator (0x09) <number active application> 2 digits for the active application <number> 2 digits for the application number	The active application is repeated within the application list.

13.2.3 c Command (upload PCIC output configuration)

Command	`c<length><configuration>`	
Description	Uploads a PCIC output configuration lasting this session	
Type	Action	
Reply	*	
	!	● Error in configuration ● Wrong data length
	?	Invalid command length
Note	<length> 9 digits as decimal value for the data length <configuration> configuration data	

13.2.4 C? Command (retrieve current PCIC configuration)

Command	`C?`	
Description	Retrieves the current PCIC configuration	
Type	Request	
Reply	<length><configuration>	
	?	Invalid command length
Note	<length> 9 digits as decimal value for the data length <configuration> configuration data	

13.2.5 E? Command (request current error state)

Command	`E?`	
Description	Requests the current error state	
Type	Request	
Reply	<code>	
	!	Invalid state (e.g. configuration mode)
	?	Invalid command length
Note	● <code> Error code with 8 digits as a decimal value. It contains leading zeros.	

13.2.6 f Command (set temporary application parameter)

Command	f<Parameter-ID> <reserved><value>	
Description	Set temporary application parameter	
	<Parameter-id>	Id of parameter to be set Fixed 5 bytes decimal ASCII padded with "0", e.g. "00003".
	<reserved>	Fixed to "#00000"
	<value>	Fixed 5 bytes signed decimal ASCII padded with "0" and sign, e.g. "+00777"
Type	Action	
Reply	*	Parameter successfully set
	!	Parameter-id invalid or syntax error
	?	Invalid command length
Note	Example: f00003#00000+00777	

13.2.7 G? Command (request device information)

Command	G?	
Description	Requests device information	
Type	Request	
Reply	<vendor><t><article number><t><name><t><location><t><description><t><ip> <subnet mask><t><gateway><t><MAC><t><DHCP><t><port number>	
Note	● <vendor> IFM ELECTRONIC ● <t> Tabulator (0x09) ● <article number> e.g. O3D300 ● <name> UTF8 Unicode string ● <location> UTF8 Unicode string ● <description> UTF8 Unicode string ● <ip> IP address of the device as ASCII character sting e.g. 192.168.0.96 ● <port number> port number of the XML-RPC ● <subnet mask> subnet mask of the device as ASCII e.g. 192.168.0.96 ● <gateway> gateway of the device as ASCII e.g 192.168.0.96 ● <MAC> MAC adress of the device as ASCII e.g. AA:AA:AA:AA:AA:AA ● <DHCP> ASCII string "0" for off and "1" for on	

13.2.8 H? Command (return a list of available commands)

Command	H?	
Description	Returns a list of available commands	
Type	Request	
Reply	H? - show this list t - execute Trigger T? - execute Trigger and wait for data o<io-id><io-state> - sets IO state O<io-id>? - get IO state I<image-id>? - get last image of defined type A? - get application list p<state> - activate / deactivate data output a<application number> - set active application E? - get last error V? - get current protocol version v<version> - sets protocol version c<length of configuration file><configuration file> - configures process date formatting C? - show current configuration G? - show device information S? - show statistics L? - retrieves the connection ID f<id><reserved><value> - set parameter value	

13.2.9 I? Command (request last image taken)

Command	I<image-ID>?	
Description	Request last image taken	
Type	Request	
Reply	<length><image data>	
	!	● No image available ● Wrong ID
	?	● Invalid command length
Note	<image-ID> 2 digits for the image type <length> char string with exactly 9 digits as decimal number for the image data size in bytes <image data> image data	Valid image ID: 01 - amplitude image 02 - normalised amplitude image 03 - distance image 04 - X image (distance information) 05 - Y image (distance information) 06 - Z image (distance information) 07 - confidence image (status information) 08 - extrinsic calibration 09 - unit_vector_matrix_ex, ey,ez 10 - last result output as formatted for this connection 11 - all distance images: X, Y, and Z

13.2.10 o Command (set logic state of a ID)

Command	o<IO-ID><IO-state>	
Description	Sets the logic state of a specific ID	
Type	Action	
Reply	*	
	!	Invalid state (e.g. configuration mode)
	?	Invalid command length
Note	● <IO-ID> 2 digits for digital output: "01" for IO1 "02" for IO2 "03" for IO3 ● <IO-state> 1 digit for the state: "0" for logic state low "1" for logic state high	

13.2.11 O? Command (request state of a ID)

Command	O<IO-ID>?	
Description	Requests the state of a specific ID	
Type	Request	
Reply	<IO-ID><IO-state>	
	!	• Invalid state (e.g. configuration mode) • Wrong ID
	?	Invalid command length
Note	• <IO-ID> 2 digits for digital output: "01" for IO1 "02" for IO2 "03" for IO3 • <IO-state> 1 digit for the state: "0" for logic state low "1" for logic state high	The camera supports ID 1 and ID 2. The sensor supports ID 1, ID 2 and ID 3.

13.2.12 p Command (turn PCIC output on or off)

Command	p<state>	
Description	Turns the PCIC output on or off	
Type	Action	
Reply	*	
	!	<state> contains wrong value
	?	Invalid command length
Note	<state> 1 digit 0: deactivates all asynchronous output 1: activates asynchronous result output 2: activates asynchronous error output 3: activates asynchronous error and data output 4: activates asynchronous notifications 5: activates asynchronous notifications and asynchronous result 6: activates asynchronous notifications and asynchronous error output 7: activates all outputs	On device restart the value configured within the application is essential for the output of data. This command can be executed in any device state. By default the error codes will not be provided by the device.

13.2.13 S? Command (request current decoding statistics)

Command	S?	
Description	Requests current decoding statistics	
Type	Request	
Reply	<number of results><t><number of positive decodings><t><number of false decodings>	
	!	No application active
Note	<t> tabulator (0x09) <number of results> Images taken since application start. 10 digits decimal value with leading 0s <number of positive decodings> Number of decodings leading to a positive result. 10 digits decimal value with leading 0s <number of false decodings> Number of decodings leading to a negative result. 10 digits decimal value with leading 0s	

13.2.14 t Command (execute asynchronous trigger)

Command	t	
Description	Executes trigger. The result data is send asynchronously	
Type	Action	
Reply	*	Trigger was executed, the device captures an image and evaluates the result.
	!	● Device is busy with an evaluation ● Device is in an invalid state for this command, e.g. configuration mode ● Device is set to a different trigger source ● No active application

13.2.15 T? Command (execute synchronous trigger)

Command	T?	
Description	Executes trigger. The result data is send synchronously	
Type	Request	
Reply	Process data within the configured layout	Trigger was executed, the device captures an image, evaluates the result and sends the process data.
	!	● Device is busy with an evaluation ● Device is in an invalid state for this command, e.g. configuration mode ● Device is set to a different trigger source ● No active application
Note	Result data can be sent via EtherNet/IP, PROFINET or TCP/IP (→ 9.3).	

13.2.16 v Command (set current protocol version)

Command	v<version>	
Description	Sets the current protocol version. The device configuration is not affected	
Type	Action	
Reply	*	
	!	Invalid version
	?	Invalid command length
Note	<version> 2 digits for the protocol version	(→ 13.1.1)

The default protocol version is „V3“.

13.2.17 V? Command (request current protocol version)

Command	V?	
Description	Requests current protocol version	
Type	Request	
Reply	<current version><empty><min version><empty><max version>	
Note	<current version> 2 digits for the currently set version <empty> space sign: 0x20 <min/max version> 2 digits for the available min and max version that can be set	

13.3 Error codes

By default the error codes will not be provided by the device. The p command can activate their provision (→ 13.2.12).

Error code ID	Description
100000001	Maximum number of connections exceeded
110001001	Boot timeout
110001002	Fatal software error
110001003	Unknown hardware
110001006	Trigger overrun
110002000	Short circuit on Ready for Trigger
110002001	Short circuit on OUT1
110002002	Short circuit on OUT2
110002003	Reverse feeding
110003000	Vled overvoltage
110003001	Vled undervoltage
110003002	Vmod overvoltage
110003003	Vmod undervoltage
110003004	Mainboard overvoltage
110003005	Mainboard undervoltage
110003006	Supply overvoltage
110003007	Supply undervoltage
110003008	VFEMon alarm
110003009	PMIC supply alarm
110004000	Illumination overtemperature

13.4 EtherNet/IP

13.4.1 Data structures for consuming and producing assemblies

Assemblies

Instance	Bytes	Type
100	8	Consuming (from device point of view: databuffer for receiving from PLC)
101	450	Producing (from device point of view: databuffer for sending to PLC)

Consuming assembly data layout

Byte	0-1	2-7
Description	Command word	Command data

Layout of producing assembly

Byte	0-1	2-3	4-5	6-7	8-15	16-449
Description	Command word for mirroring	Synchronous / asynchronous message identifier	Message counter	Reserved	Mandatory message data (e.g. error code)	Non mandatory data fields

Layout of command word

Bit	0	1-15
Description	Error bit This bit has no meaning in the consuming assembly. It is used for signaling an occured error to the PLC	Command bits Each bit represents a specific command

Command word

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Description	Error bit	N.a.	N.a.	N.a.	N.a.	N.a.	Get last error	Get connection ID	Get statistics	Activate application	Get application list	Get IO state	Set IO state	Execute synchronous trigger	Activate asynchronous PCIC output	Use extended command

Synchronous / asynchronous message identifier

Bit	0	1-15
Description	Asynchronous message bit	Bits for asynchrounous message identifier

Data to send exceeds processing assembly data section size

If the size of the data exceeds the size of the configured processing assembly data section size, the data is truncated. No error is risen.

13.4.2 Functionality of the Ethernet/IP application

The chapter describes the initialization of assembly buffers.

On initialization all buffers are set to 0.

State change 0 -> 1 of a command bit in consuming assembly

If the state of one command bit switches from 0 to 1, the according command is executed passing the information within the command data section.

Multiple state changes

If multiple bits have a transition from 0 -> 1 the event is handled as an error.

Reset of command bit state by PLC

The PLC has to reset the command bit from 1 -> 0 before it can execute a new command again. The device has to reset the command word and increase the message counter within the producing assembly.

Blocking of asynchronous messages

As long as the command handshake procedure has not been finished, no asynchronous message is allowed to be sent via the Ethernet/IP interface.

Client disconnect

If the client is disconnecting before finishing the handshake procedure, the handshake procedure is canceled and all buffers are reset.

General reply to an implemented command

If the command is implemented, the data in the data section is applicable and the execution of the command does not lead to an error. The producing assembly is filled as follows:

- Error bit = 0
- Command bits = mirror of the command within the consuming assembly
- Asynchronous message bit = 0
- Asynchronous message identifier = 0
- Message counter increased by 1
- Message data set to 0

Reply to an implemented command - reply contains specific data

If the command is implemented, the data in the data section is applicable and the execution of the command does not lead to an error. The producing assembly is filled as follows:

- Error bit = 0
- Command bits = mirror of the command within the consuming assembly
- Asynchronous message bit = 0
- Asynchronous message identifier = 0
- Message counter increased by 1
- Message data set according to the command definition

Reply to an implemented command with error in data section

If the content of the data section is not suitable to the command, the message is handled as an error. The producing assembly contains the following data:

- Error bit = 1
- Command bits = mirror of the command within the consuming assembly
- Asynchronous message bit = 0
- Asynchronous message identifier = 0
- Message counter increased by 1

No error code is sent in the data section. The error code is polled with the "get last error" command.

Reply to an implemented command that leads to an error

If the execution of the command leads to an error, the producing assembly contains the following data:

- Error bit = 1
- Command bits = mirror of the command within the consuming assembly
- Asynchronous message bit = 0
- Asynchronous message identifier = 0
- Message counter increased by 1

No error code is sent in the data section. The error code is polled with the "get last error" command.

Reply to a not implemented command

If a command bit with no functionality is received, it undergoes a transition from 0 -> 1 and the message is handled as an error. The producing assembly contains the following data:

- Error bit = 1
- Command bits = mirror of the command within the consuming assembly
- Asynchronous message bit = 0
- Asynchronous message identifier = 0
- Message counter increased by 1

No error code is sent in the data section. The error code is polled with the "get last error" command.

Reset of error bit

The error bit will be resetted to 0, if

- the error code caused by an command is retrieved from the client
- a system error is not present anymore.

Functionality of asynchronous message bit

If the message contain asynchronous data (frame results, system errors, etc.), the asynchronous message bit must be set to 1.

Bits for asynchronous message identifier

If the message contains asynchronous data, the identifier represents the asynchronous message type.

The ticket number for asynchronous results is 0.

The ticket number for asynchronous error codes is 1.

Message counter

For each message sent via the producing assembly, the message counter is increased. The counter starts with the value 1. If the maximum counter is reached, it starts with 1 again.

Get last error

This command is used to reset the error bit.

Get connection ID

This command retrieves the connection ID of the current Ethernet/IP connection. The content of the producing assembly mandatory data section is:

- Bytes 0-3: connection ID, 32 bit unsigned integer

Get statistics

This command retrieves the current statistics. The content of the producing assembly mandatory data section is:

- Bytes 0-3: total readings since application start
- Bytes 4-7: passed readings
- Bytes 8-11: failed readings

All values are 32 bit unsigned integers.

Default endianness

The default endianness is in little-endian format.

Activate application

This command activates the application defined by the bytes 6 and 7 of the consuming assembly data section. The bytes 2-5 have to be set to 0. An error is risen if bytes 2-5 are not set to 0.

The data content of the processing assembly is set to 0.

Get application list

This command retrieves the current configuration list. The content of the producing assembly mandatory data section is:

- Bytes 0-3: total number of saved applications, 32 bit unsigned integer
- Bytes 4-7: number of active application, 32 bit unsigned integer
- Bytes 8-n: always a 32 bit unsigned integer for an application number in use

Get IO state

Retrieves the logic state of the given IO identifier. Bytes 4 and 5 of the consuming assembly data section defines the IO ID as a 16 bit unsigned integer value:

- 1 -> IO1
- 2 -> IO2
- 3 -> IO3

The bytes 2-3 and 6-7 have to be set to 0. An error is risen if bytes 2-3 or 6-7 are not set to 0.

The data content of the processing assembly is:

- Bytes 0-3: logic state of the IO, 1 for high, 0 for low, 32 bit unsigned integer

Set IO state

This command sets the given state of the given IO. Bytes 4 and 5 of the consuming assembly data section defines the IO ID as a 16 bit unsigned integer value:

- 1 -> IO1
- 2 -> IO2
- 3 -> IO3

The bytes 6 and 7 define the logic state of the IO as 16 bit unsigned integer value.

The bytes 2-3 have to be set to 0. An error is risen if bytes 2-3 are not set to 0.

The data content of the processing assembly is set to 0.

Execute synchronous trigger

This command executes a synchronous trigger. The content of the producing assembly data section depends on the user defined PCIC output for Ethernet/IP.

Activate asynchronous PCIC output

This command activates or deactivates the asynchronous PCIC output for this connection. The bytes 6 and 7 of the consuming assembly data section define the on/off state as a 16 bit unsigned integer value:

- 0 = off
- 1 = on

The bytes 2-5 have to be set to 0. An error is risen if bytes 2-5 are not set to 0.

The data content of the processing assembly is set to 0.

For the Ethernet/IP interface the user shall only be able to select the binary representation of result data.

13.4.3 Extended commands

Use of extended command

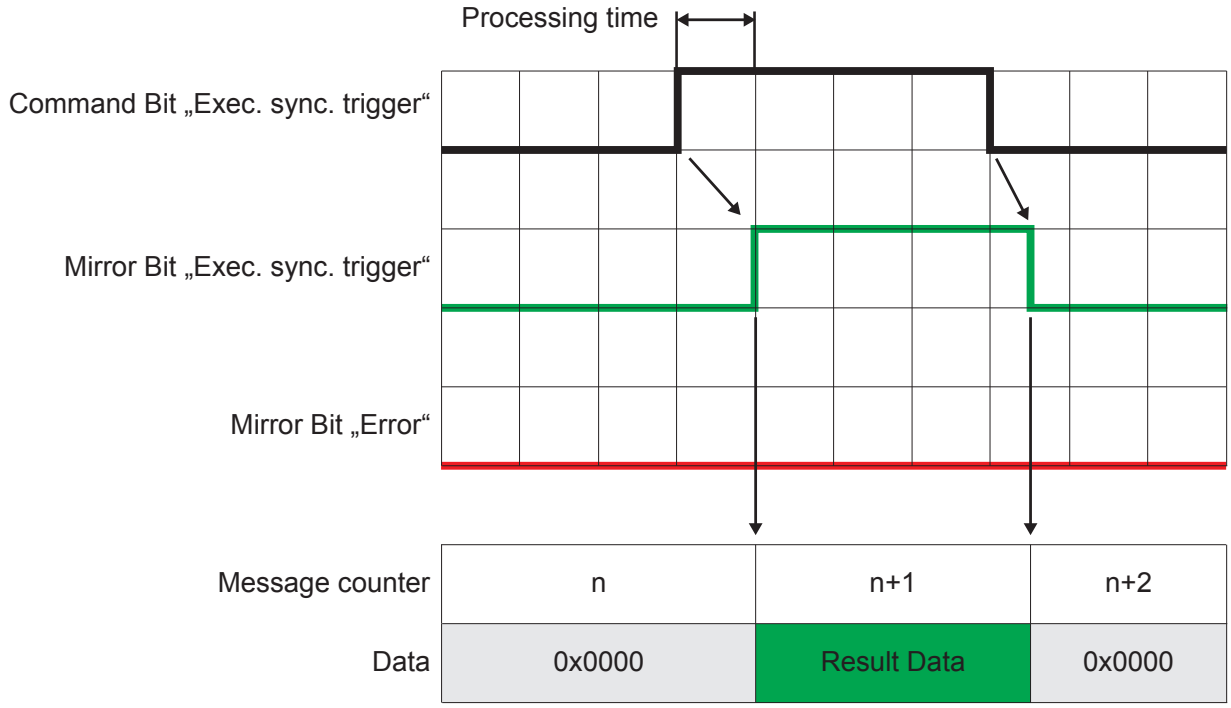
The following command executes an extended command. The ID of the extended command is stored as 16 bit integer in bytes 2-3. The remaining data depends on the extended command.

ID	Description
1	Set temporary application parameter The ID of the parameter to be changed is stored as unsigned 16 bit integer in bytes 4-5. The value of the parameter is stored as signed 16 bit integer in bytes 6-7.

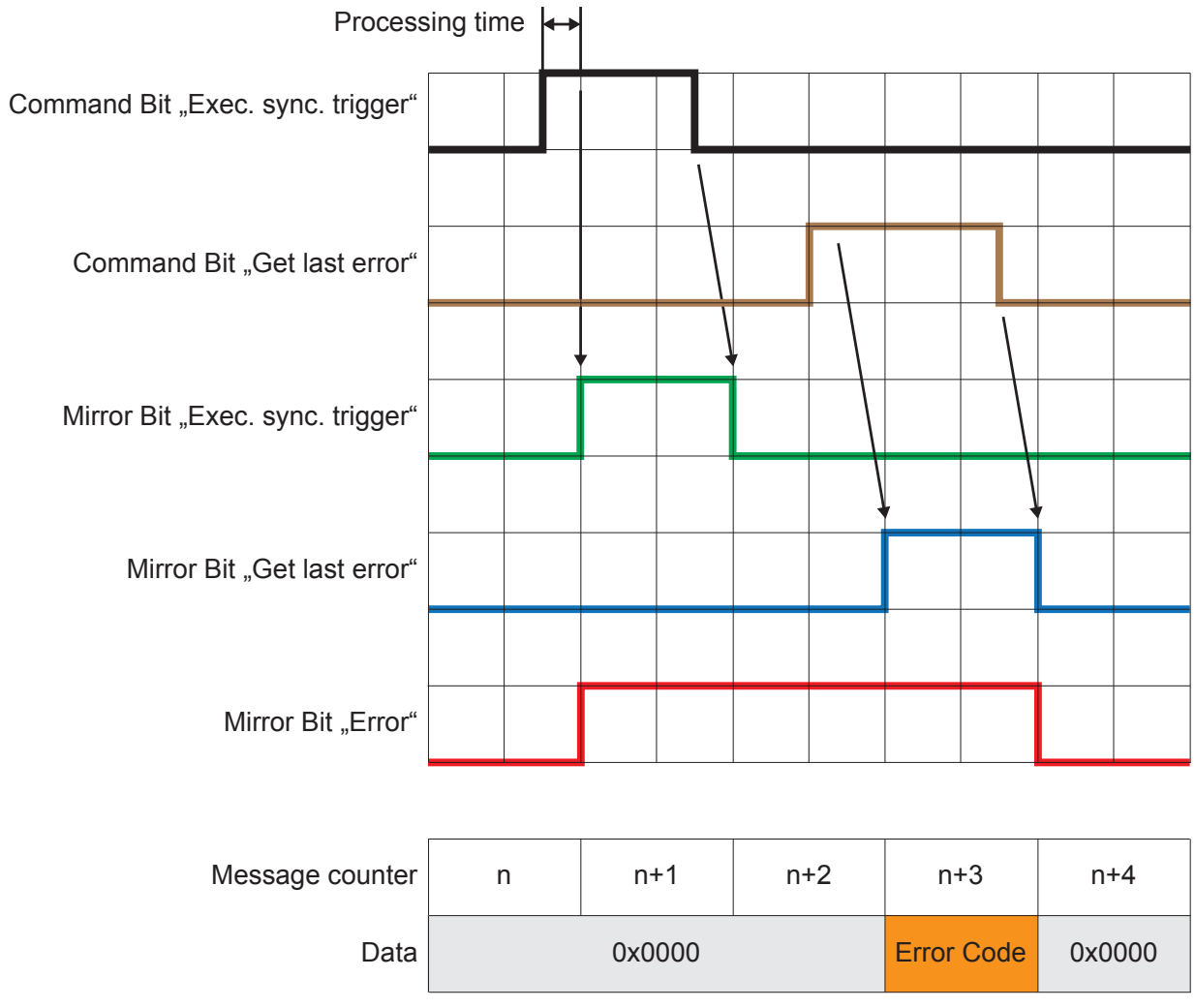
Use of extended command with the depalletising application

Byte	1 (Bit 7)	2-3	4-5	6-7
Description	Use extended command high / low	Extended command ID 1 = set temporary application parameter	Parameter ID 1 = DepalSlipSheetDetection 2 = Type of the object to detect 3 = DepalWidth 4 = DepalHeight 5 = DepalLength	Parameter value 1 = on / 0 = off 1 = bag / 0 = box value [mm] value [mm] value [mm]

13.4.4 Signal sequence with synchronous trigger



13.4.5 Signal sequence with failed trigger



13.5 PROFINET IO

13.5.1 Data structures for output and input frame

Size of output frame

Every output frame sent by the controller contains 8 bytes of data, which consists of command word and command data.

Size of input frame

Every Input frame contains 16 - 450 bytes of data, which are generated by the device in response to the commands received in the output frames. The size of non mandatory data is adjustable by changing the size of the input data in the GSDML file.

Byte	0-1	2-3	4-5	6-7	8-15	16-449
Description	Command word for mirroring	Synchronous / asynchronous message identifier	Message counter	Reserved	Mandatory data	Non mandatory data

Layout of command word

Bit	0	1-15
Description	Error bit This bit has no meaning in the consuming assembly. It is used for signaling an occured error to the PLC	Command bits Each bit represents a specific command

Command word

Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Description	Error bit	N.a.	N.a.	N.a.	N.a.	N.a.	Get last error	Get connection ID	Get statistics	Activate application	Get application list	Get IO state	Set IO state	Execute synchronous trigger	Activate asynchronous PCIC output	Use extended command

Synchronous / asynchronous identifier

Bit	0	1-15
Description	Asynchronous message bit	Bits for asynchrounous message identifier

13.5.2 Functionality of PROFINET IO application

This section describes how to handle the commands sent by the controller. The PLC sends the commands to the device in the output frames by setting the appropriate bit in the command word. The current value of the command word and command data is obtained from the output module by the application.

After detecting that one of the command bits changed the state from 0 to 1, the PROFINET application executes the corresponding command and sets the response in the input frames.

Number of supported PROFINET connections

The O3D3xx running a PROFINET application supports one connection with a single controller.

Initialisation of input and output buffers

After the connection is established, the input and output buffers are initialised with 0 s.

Command execution triggering

As soon as the command bit in the output frame changes from 0 to 1, the corresponding command will be executed.

Handling of multiple command bits

If more than one command bit is set to 1, an error will be reported.

Command execution completion

The PLC has to reset the command bit from 1 to 0 before a new command can be executed. The device has to reset the command word and increase the message counter within the input frame. Mandatory and non mandatory data in the response frame is set to 0x0.

Blocking of asynchronous messages

As long as the command handshake procedure has not been finished, no asynchronous message will be sent by the device.

Client disconnect

If the client is disconnecting before finishing the handshake procedure, the handshake procedure is canceled and all buffers are reset.

General reply to an implemented command

If the command is implemented, the data in the data section is applicable and the execution of the command does not lead to an error. The input frame contains the following data:

- Error bit = 0
- Command bits = mirror of the command within the output frame
- Asynchronous message bit = 0
- Asynchronous message identifier = 0
- Message counter increased by 1
- Message data set to 0

Reply to an implemented command - reply contains specific data

If the command is implemented, the data in the data section is applicable and the execution of the command does not lead to an error. The input frame contains the following data:

- Error bit = 0
- Command bits = mirror of the command within the output frame
- Asynchronous message bit = 0
- Asynchronous message identifier = 0
- Message counter increased by 1
- Message data set according to the command definition

Reply to an implemented command with error in data section

If the content of the data section is not suitable to the command, the message is handled as an error. The input frame contains the following data:

- Error bit = 1
- Command bits = mirror of the command within the output frame
- Asynchronous message bit = 0
- Asynchronous message identifier = 0
- Message counter increased by 1

No error code is sent in the data section. The error code is polled with the "get last error" command. Mandatory and non mandatory data in the response frame will be set to 0x0.

Reply to an implemented command that leads to an error

If the execution of the command leads to an error, the input frame contains the following data:

- Error bit = 1
- Command bits = mirror of the command within the output frame
- Asynchronous message bit = 0
- Asynchronous message identifier = 0
- Message counter increased by 1

No error code is sent in the data section. The error code is polled with the "get last error" command. Mandatory and non mandatory data in the response frame will be set to 0x0.

Reply to a not implemented command

If a command bit with no functionality is received, it undergoes a transition from 0 -> 1 and the message is handled as an error. The input frame contains the following data:

- Error bit = 1
- Command bits = mirror of the command within the output frame
- Asynchronous message bit = 0
- Asynchronous message identifier = 0
- Message counter increased by 1

No error code is sent in the data section. The error code is polled with the "get last error" command. Mandatory and non mandatory data in the response frame will be set to 0x0.

Reset of error bit

The error bit will be resetted to 0, if

- the error code caused by an command is sent to the controller
- a system error is not present anymore

Queuing of error codes

The Profinet application is able to buffer one system error (the last one) and one command error (also the last one). The buffered system error and PCIC command error will be cleared, after they are read by the PLC with the "get last error" command.

Functionality of asynchronous message bit

If the message contain asynchronous data (frame results, system errors, etc.), the asynchronous message bit must be set to 1.

Bits for asynchronous message identifier

If the message contains asynchronous data, the identifier represents the asynchronous message type:

- The ticket number for asynchronous results is 0
- The ticket number for asynchronous error codes is 1
- The reserved ticket numbers for asynchronous messages are in the range 0-99

Message counter

For each command response sent in the input frame the message counter is increased. The counter starts with value 1. If the maximum counter is reached, it starts with 1 again.

Get last error

This command retrieves the current command and system error. The content of the mandatory data section sent in the input frame is:

- Bytes 0-3 : command error code, 32 bit unsigned integer
- Bytes 4-7: system error code, 32 bit unsigned integer

Get connection ID

This command retrieves the connection ID of the current Profinet connection. The response sent in the input frame contains 16 Bytes of the AR UUID.

Get statistics

This command retrieves the current statistics. The content of the mandatory data section sent in the input frame is:

- Bytes 0-3: total readings since application start
- Bytes 4-7: passed readings
- Bytes 8-11: failed readings

All values are 32 bit unsigned integers.

Default endianness

The default endianness is in little-endian format.

Activate application

This command activates the application defined by the bytes 6 and 7 of the output frame data section. The bytes 2-5 have to be set to 0. An error is risen if bytes 2-5 are not set to 0.

The data content of the input frame is set to 0, after receiving the "Activate application" command.

Get application list

This command retrieves the current configuration list. The content of the response sent in the input frame mandatory data section is:

- Byte 0-3: total number of saved applications, 32 bit unsigned integer
- Bytes 4-7: number of active application, 32 bit unsigned integer
- Bytes 8-n: always a 32 bit unsigned integer for an application number in use

Get IO state

Retrieves the logic state of the given IO identifier. Bytes 4 and 5 of the output frame data section defines the IO ID as a 16 bit unsigned integer value:

- 1 -> IO1
- 2 -> IO2
- 3 -> IO3

The bytes 2-3 and 6-7 have to be set to 0. An error is risen if bytes 2-3 or 6-7 are not set to 0.

The data sent in the input frame is:

- Byte 0-3: logic state of the requested IO, 1 for high, 0 for low, 32 bit unsigned integer

Set IO state

This command sets the given state of the given IO. Bytes 4 and 5 of the output frame data section defines the IO ID as a 16 bit unsigned integer value:

- 1 -> IO1
- 2 -> IO2
- 3 -> IO3

The bytes 6 and 7 define the logic state of the IO as 16 bit unsigned integer value.

The bytes 2-3 have to be set to 0. An error is risen if bytes 2-3 are not set to 0.

The data content of the input frame is set to 0, after receiving the "Set IO state" command.

Execute synchronous trigger

This command executes a synchronous trigger. The content of the input frame data section depends on the user defined PCIC output for PROFINET.

Activate asynchronous PCIC output

This command activates or deactivates the asynchronous PCIC output for this connection. The bytes 6 and 7 of the output frame data section define the on/off state as a 16 bit unsigned integer value:

- 0 = off
- 1 = on

The bytes 2-5 have to be set to 0. An error is risen if bytes 2-5 are not set to 0.

The data content of the input frame is set to 0, after receiving the "Activate asynchronous PCIC output" command.

13.5.3 Extended commands

Use of extended command

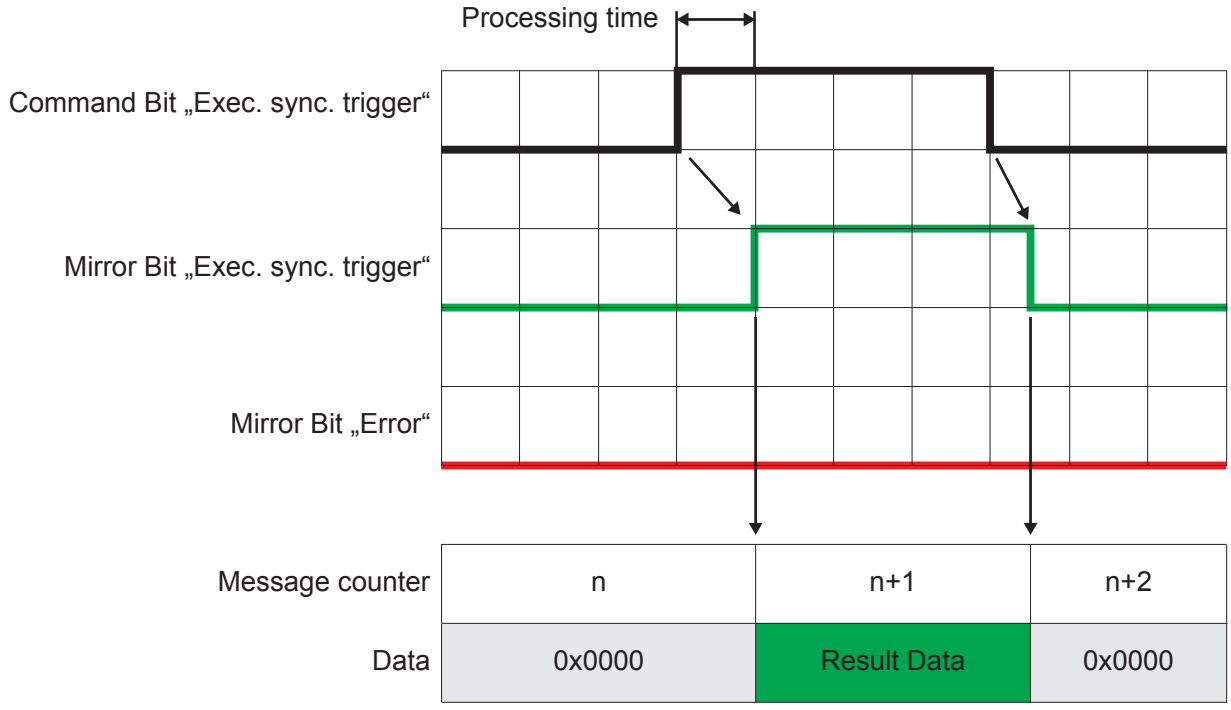
The following command executes an extended command. The ID of the extended command is stored as 16 bit integer in bytes 2-3. The remaining data depends on the extended command.

ID	Description
1	Set temporary application parameter The ID of the parameter to be changed is stored as unsigned 16 bit integer in bytes 4-5. The value of the parameter is stored as signed 16 bit integer in bytes 6-7.

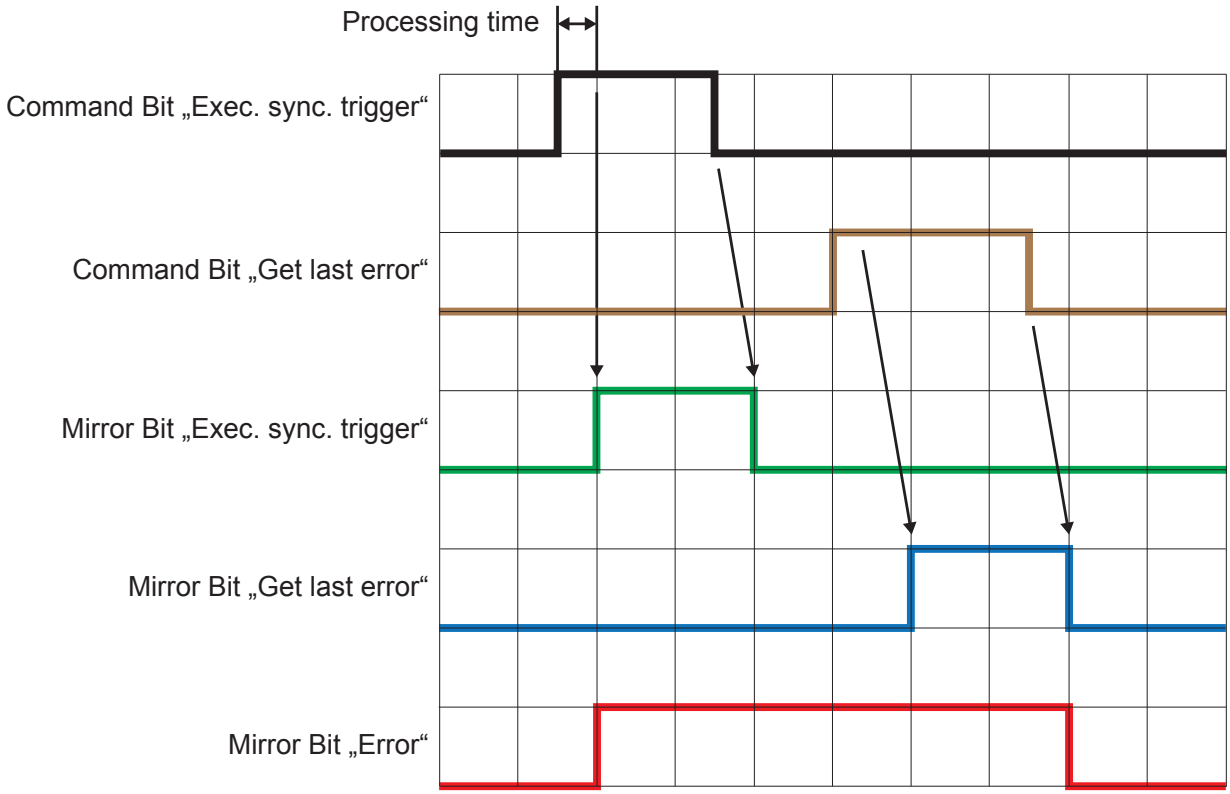
Use of extended command with the depalletising application

Byte	0 (Bit 7)	2-3	4-5	6-7
Description	Use extended command high / low	Extended command ID 1 = set temporary application parameter	Parameter ID 1 = DepalSlipSheetDetection 2 = Type of the object to detect 3 = DepalWidth 4 = DepalHeight 5 = DepalLength	Parameter value 1 = on / 0 = off 1 = bag / 0 = box value [mm] value [mm] value [mm]

13.5.4 Signal sequence with synchronous trigger



13.5.5 Signal sequence with failed trigger



Message counter	n	n+1	n+2	n+3	n+4
Data	0x0000			Error Code	0x0000