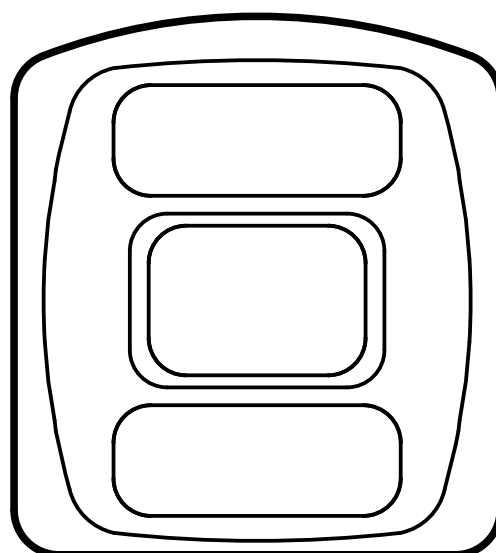




Bedienungsanleitung 3D Sensor

DE

O3D300
O3D302
O3D310
O3D312



Inhalt

1. Vorbemerkung	4
1.1 Verwendete Symbole	4
1.2 Verwendete Warnhinweise	4
1.3 Open source information	5
2. Sicherheitshinweise	6
2.1 Allgemein	6
2.2 Zielgruppe	6
2.3 Elektrischer Anschluss	6
2.4 Eingriffe in das Gerät	6
3. Bestimmungsgemäße Verwendung	7
4. Lieferumfang	7
5. Zubehör	7
6. Montage	8
6.1 Montageort wählen	8
6.2 Gerät auf Inbetriebnahme vorbereiten	9
6.2.1 Typische Warngrenzen für O3D300 / O3D302	9
6.2.2 Typische Warngrenzen für O3D310 / O3D312	10
6.2.3 Reduzieren der Oberflächentemperatur	10
6.3 Gerät montieren	11
6.4 Montagezubehör	11
7. Elektrischer Anschluss	12
7.1 Anschlussbelegung	12
7.1.1 Pin 1 / 3 (24 V / GND)	13
7.1.2 Pin 2 (Triggereingang)	13
7.1.3 Pin 4 / 5 / 6 (Schaltausgänge)	13
7.1.4 Pin 4 (Analogausgang)	14
7.1.5 Pin 7 / 8 (Schalteingänge)	14
7.2 Verdrahtungsbeispiele	15
7.2.1 Bildaufnahme mit Nährungsschalter triggern	15
7.2.2 Mehrere Geräte nebeneinander verwenden	16
7.3 Statische Anwendungsumschaltung	17
7.4 Pulsgesteuerte Anwendungsumschaltung	18
8. Anzeigeelemente	19
9. Inbetriebnahme	20
9.1 Gerät parametrieren	20
9.2 Objekt detektieren	20
9.3 Prozesswerte senden	21
9.3.1 Prozesswerte der Vollständigkeitsüberwachung über EtherNet/IP senden	21
9.3.2 Prozesswerte der Vollständigkeitsüberwachung über PROFINET senden	23
9.3.3 Prozesswerte der Vollständigkeitsüberwachung über TCP/IP senden	25
9.3.4 Prozesswerte der Objektvermessung über EtherNet/IP senden	26
9.3.5 Prozesswerte der Objektvermessung über PROFINET senden	28
9.3.6 Prozesswerte der Objektvermessung über TCP/IP senden	30
9.3.7 Prozesswerte der Füllstandsmessung über EtherNet/IP senden	31
9.3.8 Prozesswerte der Füllstandsmessung über PROFINET senden	32
9.3.9 Prozesswerte der Füllstandsmessung über TCP/IP senden	33
9.3.10 Prozesswerte der Roboter Greifernavigation über EtherNet/IP senden	34
9.3.11 Prozesswerte der Roboter Greifernavigation über PROFINET senden	36
9.3.12 Prozesswerte der Roboter Greifernavigation über TCP/IP senden	38
9.3.13 Prozesswerte der Depalettierung über EtherNet/IP senden	39
9.3.14 Prozesswerte der Depalettierung über PROFINET senden	41

9.3.15	Prozesswerte der Depalettierung über TCP/IP senden	43
10.	Wartung, Instandsetzung und Entsorgung	44
10.1	Reinigung	44
10.2	Firmware aktualisieren	44
10.3	Gerät austauschen	44
11.	Zulassungen/Normen	44
12.	Maßzeichnungen	45
12.1	O3D302 / O3D312	45
12.2	O3D300 / O3D310	45
13.	Appendix	46
13.1	Process Interface	46
13.1.1	Sending Commands	46
13.1.2	Receiving Images	48
13.1.3	Image data	48
13.1.4	Additional Information for CONFIDENCE_IMAGE	52
13.1.5	Configuration of PCIC Output	53
13.2	Process Interface Command Reference	63
13.2.1	a Command (activate application)	63
13.2.2	A? Command (occupancy of application list)	63
13.2.3	c Command (upload PCIC output configuration)	64
13.2.4	C? Command (retrieve current PCIC configuration)	64
13.2.5	E? Command (request current error state)	64
13.2.6	f Command (set temporary application parameter)	65
13.2.7	G? Command (request device information)	66
13.2.8	H? Command (return a list of available commands)	67
13.2.9	I? Command (request last image taken)	68
13.2.10	o Command (set logic state of a ID)	68
13.2.11	O? Command (request state of a ID)	69
13.2.12	p Command (turn PCIC output on or off)	69
13.2.13	S? Command (request current decoding statistics)	70
13.2.14	t Command (execute asynchronous trigger)	70
13.2.15	T? Command (execute synchronous trigger)	71
13.2.16	v Command (set current protocol version)	71
13.2.17	V? Command (request current protocol version)	71
13.3	Error codes	72
13.4	EtherNet/IP	73
13.4.1	Data structures for consuming and producing assemblies	73
13.4.2	Functionality of the Ethernet/IP application	74
13.4.3	Extended commands	78
13.4.4	Signal sequence with synchronous trigger	79
13.4.5	Signal sequence with failed trigger	79
13.5	PROFINET IO	80
13.5.1	Data structures for output and input frame	80
13.5.2	Functionality of PROFINET IO application	80
13.5.3	Extended commands	85
13.5.4	Signal sequence with synchronous trigger	85
13.5.5	Signal sequence with failed trigger	86

Copyright

Microsoft®, Windows®, Windows Vista®, Windows 7®, Windows 8®, Windows 8.1® und Windows 10® sind eingetragene Warenzeichen der Microsoft Corporation.

Adobe® und Acrobat® sind eingetragene Warenzeichen der Adobe Systems Inc.

Alle benutzten Warenzeichen und Firmenbezeichnungen unterliegen dem Copyright der jeweiligen Firmen.

1. Vorbemerkung

Das Dokument richtet sich an Fachkräfte. Dabei handelt es sich um Personen, die aufgrund ihrer einschlägigen Ausbildung und ihrer Erfahrung befähigt sind, Risiken zu erkennen und mögliche Gefährdungen zu vermeiden, die der Betrieb oder die Instandhaltung des Gerätes verursachen kann. Das Dokument enthält Angaben zum korrekten Umgang mit dem Gerät.

Lesen Sie dieses Dokument vor dem Einsatz, damit Sie mit Einsatzbedingungen, Installation und Betrieb vertraut werden. Bewahren Sie das Dokument während der gesamten Einsatzdauer des Gerätes auf.

1.1 Verwendete Symbole

- ▶ Handlungsanweisung
- > Reaktion, Ergebnis
- [...] Bezeichnung von Tasten, Schaltflächen oder Anzeigen
- Querverweis
-  Wichtiger Hinweis
Fehlfunktionen oder Störungen sind bei Nichtbeachtung möglich.
-  Information
Ergänzender Hinweis

1.2 Verwendete Warnhinweise

ACHTUNG
Warnung vor Sachschäden.

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 ("This product includes software developed by the University of California, Berkeley and its contributors"), The Academic Free License version 2.1. 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 Lesser General Public License in their respective versions: The Free Software is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even 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 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.

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 GLP/LGPL covered code).

2. Sicherheitshinweise

2.1 Allgemein

Diese Beschreibung ist Bestandteil des Gerätes. Sie enthält Texte und Abbildungen zum korrekten Umgang mit dem Gerät und muss vor einer Installation oder dem Einsatz gelesen werden.

Befolgen Sie die Angaben dieser Anleitung. Nichtbeachten der Hinweise, Betrieb außerhalb der nachstehend bestimmungsgemäßen Verwendung, falsche Installation oder fehlerhafte Handhabung können schwerwiegende Beeinträchtigungen der Sicherheit von Menschen und Anlagen zur Folge haben.

2.2 Zielgruppe

Die Anleitung richtet sich an Personen, die im Sinne der EMV- und der Niederspannungsrichtlinie als fachkundig angesehen werden. Das Gerät darf nur von einer Elektrofachkraft eingebaut, angeschlossen und in Betrieb gesetzt werden.

2.3 Elektrischer Anschluss

Schalten Sie das Gerät extern spannungsfrei bevor Sie irgendwelche Arbeiten vornehmen.

An den Anschlusspins dürfen nur die in den technischen Daten bzw. auf dem Geräteaufdruck angegebenen Signale eingespeist und die zugelassenen Zubehörkomponenten der ifm angeschlossen werden.

2.4 Eingriffe in das Gerät

Bei Fehlfunktionen oder Unklarheiten mit dem Hersteller in Verbindung setzen. Eingriffe in das Gerät können schwerwiegende Beeinträchtigungen der Sicherheit von Menschen und Anlagen zur Folge haben. Sie sind nicht zulässig und führen zum Haftungs- und Gewährleistungsausschluss.

3. Bestimmungsgemäße Verwendung

Der O3D3xx 3D-Sensor ist ein optischer Sensor, der punktwise den Abstand zwischen Sensor und der nächsten Oberfläche per Lichtlaufzeitverfahren misst. Der O3D3xx 3D-Sensor beleuchtet die Szene mit einer internen Infrarot-Lichtquelle und berechnet die Entfernung anhand des von der Oberfläche reflektierten Lichts.

Aus den Bilddaten werden mit der internen Bildverarbeitung Prozesswerte generiert und mit Schwellenwerten verglichen. Die Vergleichs- und Prozesswerte werden mit den digitalen Ausgängen verknüpft. Dadurch können die folgenden Applikationen gelöst werden:

- Vollständigkeitsüberwachung
- Füllstandsmessung
- Abstandsüberwachung
- Vermessung von quaderförmigen Objekten
- Sortierung von quaderförmigen Objekten

Die Messdaten und Prozesswerte können über Ethernet ausgegeben und vom Anwender ausgewertet werden. Die Parametrierung des O3D3xx 3D-Sensors erfolgt ebenfalls über Ethernet.

Der O3D3xx 3D-Sensor darf nur unter den im Datenblatt angegebenen Umgebungsbedingungen verwendet werden.

Die Gerätesicherheit ist für die Nutzung unter folgenden Umgebungsbedingungen ausgelegt:

- Nutzung im Innenbereich
- Höhenlagen bis zu 2000 m
- Relative Luftfeuchtigkeit bis maximal 90%, nicht kondensierend
- Verschmutzungsgrad 3

Aufgrund der Anforderungen für elektromagnetische Störaussendungen ist das Gerät für den Einsatz in Industrieumgebungen bestimmt. Für den Einsatz in Wohnbereichen ist das Gerät nicht geeignet.



Das Gerät darf nur unter den im Datenblatt angegebenen Umgebungsbedingungen verwendet werden.

4. Lieferumfang

- O3D3xx 3D Sensor
- Kurzanleitung



Das Datenblatt und weitere Dokumentation (Softwarehandbuch etc.) stehen im Internet zu Verfügung: www.ifm.com

5. Zubehör

Zum Betrieb des Gerätes wird das folgende Zubehör benötigt:

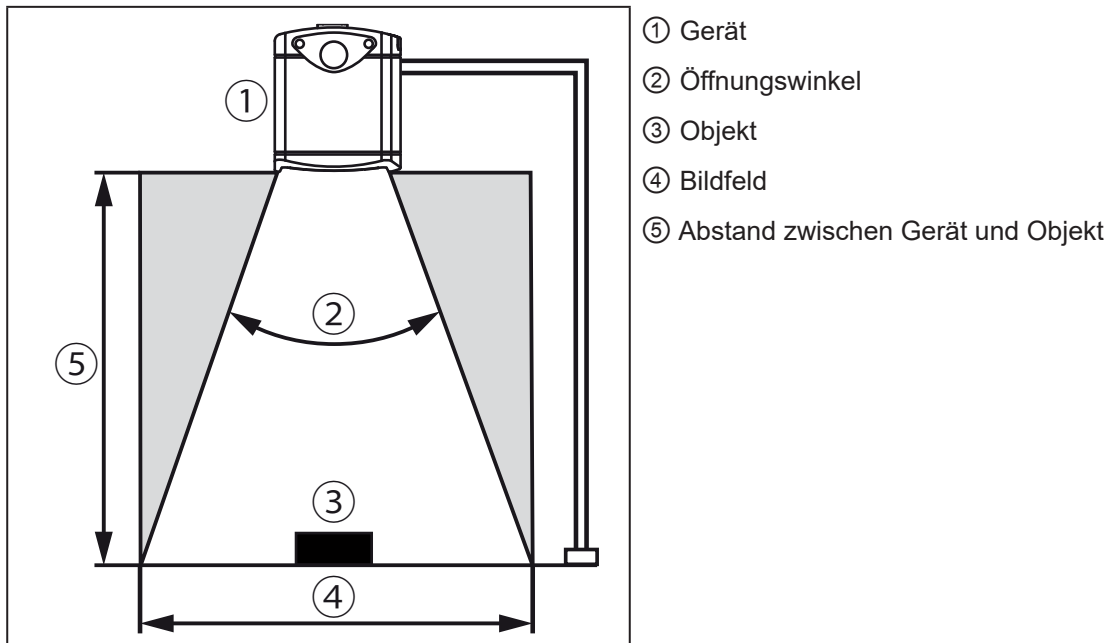
Artikelnummer	Bezeichnung
E11950	Stromversorgungskabel für Kamera/Sensor
E11898	M12-Industrial-Ethernet-Verbindungskabel



Die Software ifm Vision Assistant steht kostenlos im Internet zur Verfügung: www.ifm.com

6. Montage

Das Kapitel beschreibt, was vor der Montage beachtet werden muss und wie das Gerät montiert wird.



6.1 Montageort wählen

Beachten Sie die folgenden Anweisungen bei der Wahl des Montageortes:

- ▶ Das Objekt ③ muss sich vollständig im Bildfeld ④ befinden.
- > Die Größe des Bildfeldes hängt vom Gerätetyp ab und ist im Datenblatt angegeben. Die Größe des Bildfeldes hängt zusätzlich vom Abstand des Gerätes zum Objekt ⑤ ab: Bei steigendem Abstand wird das Bildfeld größer.
- ▶ Toleranzen beim Platzieren des Objektes berücksichtigen.
- ▶ Beim Festlegen des Abstandes zwischen Gerät und Objekt ⑤ Messbereich des Gerätes berücksichtigen.
- > Der Messbereich ist im Datenblatt des Gerätes angegeben.
- ▶ Abstand zwischen Gerät und Objekt ⑤ möglichst klein wählen.
- > Bei einem möglichst kleinen Abstand wird das Objekt mit der maximalen Auflösung erfasst.
- ▶ Am Montageort helles Umgebungslicht und Sonneneinstrahlung vermeiden.
- > Ein Fremdlichtniveau über 8 klx verursacht Messfehler (bei zugrunde gelegtem Sonnenspektrum). Tatsächlich ist nur der Infrarotanteil zwischen 800 und 900 nm störend.
- ▶ Stark verschmutzte Bereiche als Montageort vermeiden.
- > In stark verschmutzten Bereichen verschmutzt das Objektiv trotz der Ausrichtung nach unten ①.
- ▶ Transparente Scheiben zwischen Gerät ① und Objekt ③ vermeiden.
- > Transparente Scheiben reflektieren einen Teil des Lichtes, selbst beim Verwenden einer sehr sauberen Glasplatte.



Werden die Anweisungen nicht eingehalten, können Messfehler auftreten.

6.2 Gerät auf Inbetriebnahme vorbereiten

Die Oberflächentemperatur des Gerätes ist abhängig vom Betriebsmodus, der Parameterwahl und der thermischen Anbindung des Gerätes an die Umgebung.



Stellen Sie sicher, dass das Gerät die folgende Anforderung erfüllt:

Die Oberflächentemperatur für leicht berührbare Oberflächen darf max. 25°C höher sein als die Umgebungstemperatur (gemäß IEC61010-2-201).

Die folgenden Diagramme enthalten typische Warngrenzen, an denen sich der Installateur orientieren kann.



Die Diagramme sind für die folgenden Belichtungsmodi gültig:

- eine Belichtungszeit
- zwei Belichtungszeiten
- drei Belichtungszeiten

Bei zwei und drei Belichtungszeiten müssen die typischen Warngrenzen über die Summe der Belichtungszeiten ermittelt werden. Die Belichtungszeiten werden in der Software ifm Vision Assistant angezeigt.

Befolgen Sie eine der Anweisungen, wenn die Warngrenzen überschritten werden:

- Oberflächentemperatur reduzieren (→ 6.2.3).
- Berührschutz montieren, ohne die Konvektion (Luftbewegung) einzuschränken.
- > Durch den montierten Berührschutz soll sich die Oberflächentemperatur nicht erhöhen.

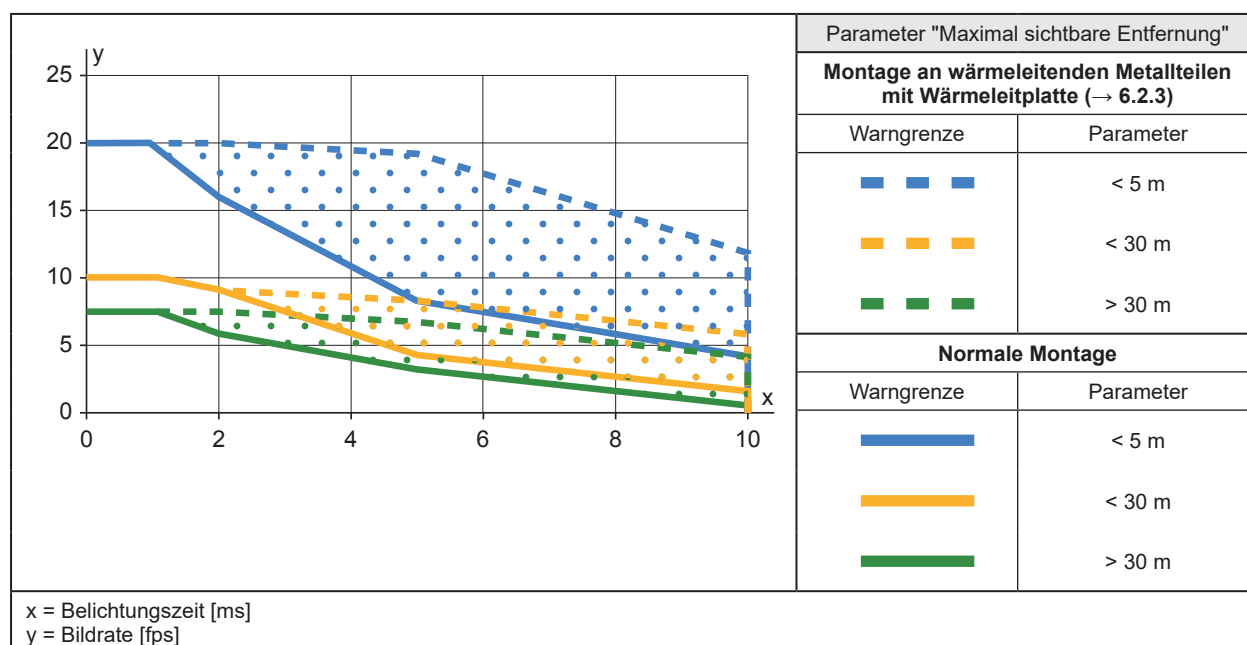


Der Parameter "Maximal sichtbare Entfernung" wird in ifm Vision Assistant eingestellt. In den Diagrammen werden die Warngrenzen des Parameters mit gestrichelten und durchgezogenen Linien dargestellt.

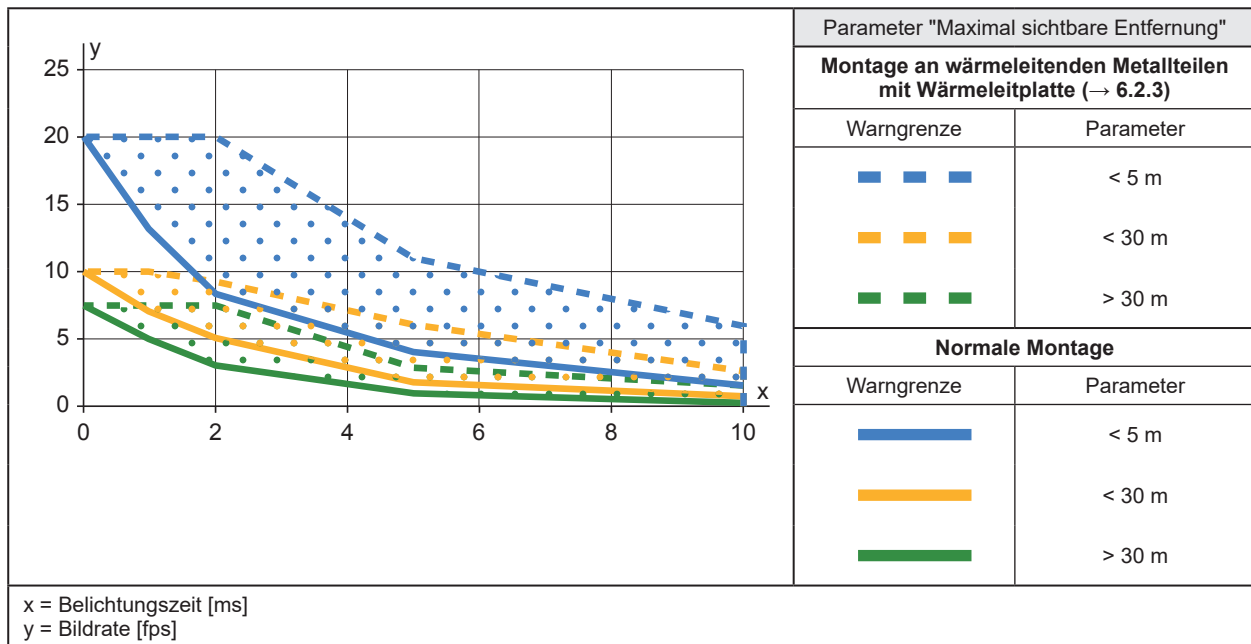
Befindet sich das Gerät in einem der punktierten Bereiche, muss die Oberflächentemperatur reduziert werden (→ 6.2.3). Wird die Warngrenze trotz wärmeableitender Montage überschritten, kann zusätzlich der Berührschutz montiert werden.

Werden die typischen Warngrenzen bei normaler Montage unterschritten, sind keine Maßnahmen notwendig.

6.2.1 Typische Warngrenzen für O3D300 / O3D302



6.2.2 Typische Warngrenzen für O3D310 / O3D312



6.2.3 Reduzieren der Oberflächentemperatur

Mit den folgenden Maßnahmen kann die Oberflächentemperatur reduziert werden:

- ▶ Gerät an wärmeleitenden Metallteilen montieren.
 - > Ein großflächiger Kontakt des Gerätes mit Metallteilen erhöht die Wärmeabfuhr (z.B. Aluminium).
- ▶ Wärmeleitplatte bei Montage auf Metallteilen verwenden.
 - > Der wärmeleitende Effekt wird durch die Wärmeleitplatte erhöht. Die Wärmeleitplatte ist als Zubehör erhältlich (→ 6.4).
- ▶ Aufbauten im Umfeld des Gerätes und Packungsdichte von Objekten reduzieren.
 - > Aufbauten im Geräteumfeld und eine erhöhte Packungsdichte können sich negativ auf die Konvektion auswirken (Luftbewegung).
- ▶ Ein oder zwei Kühlkörper am Gerät montieren.
 - > Die Kühlkörper erhöhen die Oberfläche des Gerätes, wodurch die Oberflächentemperatur reduziert wird. Die Kühlkörper sind als Zubehör erhältlich (→ 6.4).
- ▶ Belichtungszeit, Bildrate oder Maximal sichtbare Entfernung reduzieren.
 - > Der verwendete Betriebsmodus und die Parameter können die Oberflächentemperatur erhöhen.

6.3 Gerät montieren

Beachten Sie die folgenden Anweisungen bei der Montage des Gerätes:

- ▶ Gerät mit 2x M5-Schrauben oder Montageset montieren.
- > Die Bohrmaße für die M5-Schrauben sind im Datenblatt angegeben.
- > Das Montageset ist als Zubehör erhältlich (→ 6.4).
- ▶ Zugentlastungen für alle am Gerät verbundene Leitungen verwenden.

Beachten Sie die folgenden Hinweise bei der Montage eines O3D300 und O3D310:

- ▶ Gerät so montieren, dass der Fokuseinsteller mit einem Schraubendreher erreichbar ist.
- > Die Position des Fokuseinstellers ist in der Maßzeichnung angegeben (→ 12).



Bei permanenten Einsatz des Gerätes im Nassbereich, kann die Überwurfmutter des M12-Industrial-Ethernet-Verbindungskabel (z.B. E11898) korrodieren. Verwenden Sie für den permanenten Einsatz im Nassbereich ein Verbindungskabel mit einer Überwurfmutter aus V4A.

6.4 Montagezubehör

Abhängig vom Montageort und von der Montage kann das folgende Montagezubehör verwendet werden:

Artikelnummer	Bezeichnung
E3D301	Montageset Smart Camera
E3D302	Kühlkörper Smart Camera
E3D303	Wärmeleitplatte Smart Camera
E3D304	2x Kühlkörper Smart Camera



Informationen zum Zubehör unter: www.ifm.com

7. Elektrischer Anschluss

Beachten Sie die folgenden Hinweise vor der elektrischen Installation.

ACHTUNG

Das Gerät darf nur von einer Elektrofachkraft installiert werden. Elektrische Daten im Datenblatt beachten.

Gerät der Schutzklasse III (SK III)

Die elektrische Versorgung darf nur über PELV-Stromkreise erfolgen.

Die elektrische Versorgung muss der UL61010-1, Kap. 9.4 - Limited Energy entsprechen:

Die Überstromschutzeinrichtung muss einen Strom von 6,6 A in 120 s abschalten. Bei der Dimensionierung der Überstromschutzeinrichtung technische Daten des Gerätes und der Verkabelung berücksichtigen.

Die Isolierung externer Stromkreise muss der UL61010-2-201, Abb. 102 entsprechen.

Für Kabellängen > 30 m einen zusätzlichen Schutz gegen Stoßspannungen nach IEC6100-4-5 verwenden.

Stromversorgung vor elektrischen Anschluss spannungsfrei schalten.



Für den Gültigkeitsbereich cULus:

Mindesttemperaturfestigkeit des Kabels zum Anschluss an Feldanschlussklemmen: 70 °C.

7.1 Anschlussbelegung

	① Ethernet
	M12-Buchse, D-kodiert, 4-polig
	1 TD + 2 RD + 3 TD - 4 RD - S Shield
	② Stromversorgung
	M12-Stecker, A-kodiert, 8-polig
	1 U+ 2 Triggereingang 3 GND 4 Schaltausgang 1 - (digital oder analog) 5 Schaltausgang 3 - Ready 6 Schaltausgang 2 - (digital) 7 Schalteingang 1 8 Schalteingang 2



Unbenutzten Ethernet-Anschluss mit Verschlusskappe (E73004) verschließen.
Anzugsdrehmoment 0,6...0,8 Nm.



Das Verhalten der Schaltein- und -gänge kann mit der Software im Vision Assistant eingestellt werden. Die Einstellung PNP- oder NPN-schaltend gilt immer für alle Schaltein- und -gänge.

Achten Sie bei der Installation von Aktuatoren und Sensoren auf die richtige Einstellung (z.B. Lichtschranken zum Triggern).

Die Schaltausgänge können auch als Pulsausgänge betrieben werden, die ihr Schaltsignal nach einstellbarer Zeit zurücksetzen.

Der Analogausgang liefert Strom bzw. Spannung gegen GND.

7.1.1 Pin 1 / 3 (24 V / GND)

Der zulässige Spannungsbereich ist im Datenblatt des Gerätes angegeben.

7.1.2 Pin 2 (Triggereingang)

Die Bildaufnahme des Gerätes kann über den Triggereingang mit einem Schaltsignal getriggert werden.

Die folgenden Triggerflanken können verwendet werden:

- fallende Flanke triggert Bildaufnahme
- steigende Flanke triggert Bildaufnahme
- fallende und steigende Flanke triggern Bildaufnahme



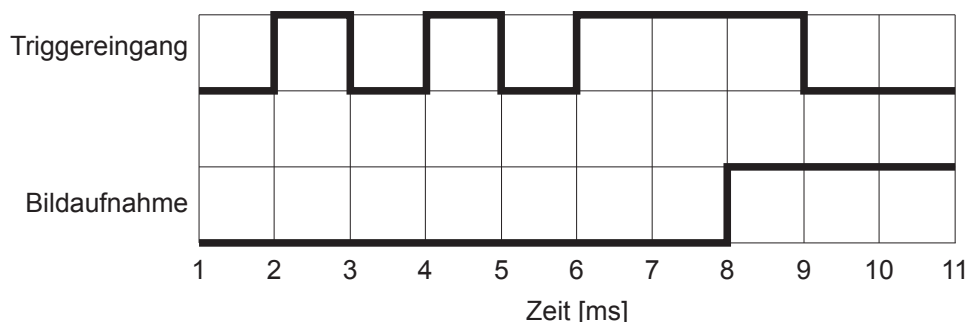
Weitere Möglichkeiten zum Triggern des Gerätes:

- Prozessschnittstellen-Kommando (→ 13.2)
- kontinuierliche Bildaufnahme mit fest einstellbarer Bildrate



Der Triggereingang ist intern entprellt. Abhängig von der elektrischen Installation kann auf das Entprellen der Triggerleitung verzichtet werden.

Die interne Entprellung verhindert, dass mehrere kurze Pulse eine Triggerung auslösen. Der Puls muss mindestens 2 ms lang sein, damit er als Triggerung erkannt wird.



7.1.3 Pin 4 / 5 / 6 (Schaltausgänge)

Die Schaltausgänge 1 bis 3 geben die verschiedenen Gerätestatus aus. Neben dem Gerätestatus können die Schaltausgänge auch die zur Lösung der Applikation notwendigen Vergleichswerte ausgeben.

Die elektrischen Spezifikationen der Schaltausgänge 1 bis 3 sind im Datenblatt angegeben.

Der Schaltausgang 3 gibt in der Voreinstellung den Gerätestatus "Bereit für Trigger" aus.



"Schaltausgang geschaltet" bedeutet, dass der entsprechende Gerätestatus eingetreten ist.

Der Gerätestatus kann je nach Einstellung einen der folgenden Werte annehmen:

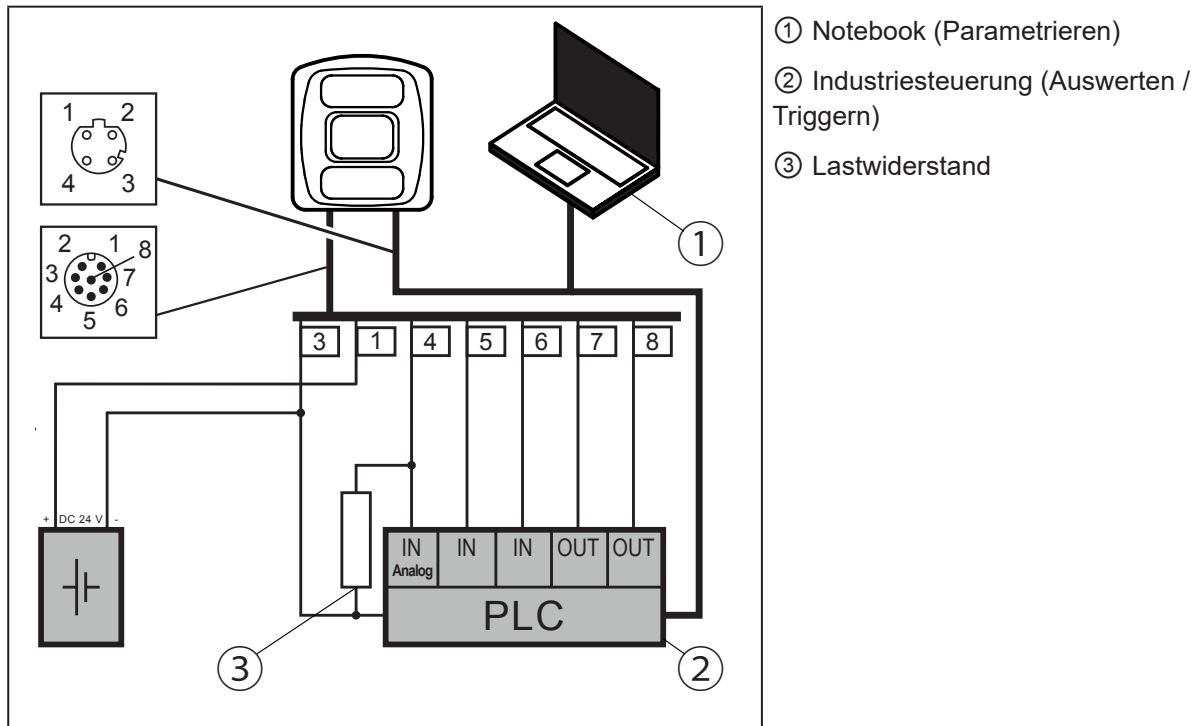
- "Bereit für Trigger"
Das Gerät meldet, dass ein neues Bild aufgenommen werden kann. Nur bei diesem Gerätestatus werden Triggerungen verarbeitet. Bei kontinuierlicher Bildaufnahme wird der Gerätestatus "Bereit für Trigger" nicht ausgegeben.
- "Bildaufnahme beendet"
Das Gerät meldet, dass die Bildaufnahme beendet ist. Der Gerätestatus kann zum Kaskadieren von Geräten verwendet werden.
- "Auswertung fertig"
Das Gerät meldet, dass die Bildverarbeitung abgeschlossen ist. Zu dem Zeitpunkt sind die Schaltausgänge bereits aktualisiert. Die Bilddaten werden über Ethernet übertragen.
- "Fehler"
Das Gerät meldet, dass intern ein Fehler vorliegt. Detaillierte Informationen zum Fehler können über Ethernet abgefragt werden.

7.1.4 Pin 4 (Analogausgang)

Der Schaltausgang 1 / Analogausgang kann als Schaltausgang oder analoger Stromausgang (4-20 mA) / analoger Spannungsausgang (0-10 V) verwendet werden.

Der analoge Stromausgang bietet gegenüber dem analogen Spannungsausgang eine höhere Übertragungssicherheit. Der analoge Stromausgang ist unabhängig von der Leitungslänge und gewährleistet eine höhere Signalqualität in Richtung Industriesteuerung.

In der Industriesteuerung wird über einen Lastwiderstand (Bürde) gegen GND der Analogstrom in eine Analogspannung umgesetzt. Der Lastwiderstand wird nach den Angaben im Datenblatt gewählt. Hochohmige Lastwiderstände sind wegen der geringeren Wärmeentwicklung im Gerät den niederohmigen Lastwiderständen vorzuziehen.



Mit der Software ifm Vision Assistant kann dem Startwert (4 mA / 0 V) und Endwert (20 mA / 10 V) des Analogausganges jeweils ein Prozesswert zugeordnet werden.

7.1.5 Pin 7 / 8 (Schalteingänge)

Die Schalteingänge stellen die folgenden Funktionen bereit:

- aktive Anwendung wählen (→ 7.3)



Die verschiedenen Parametrierungen der Funktionen sind im Softwarehandbuch angegeben.



Die elektrischen Spezifikationen von Schalteingang 1 und Schalteingang 2 sind im Datenblatt des Gerätes angegeben.

7.2 Verdrahtungsbeispiele

Im folgenden sind Verdrahtungsbeispiele des Gerätes abgebildet.

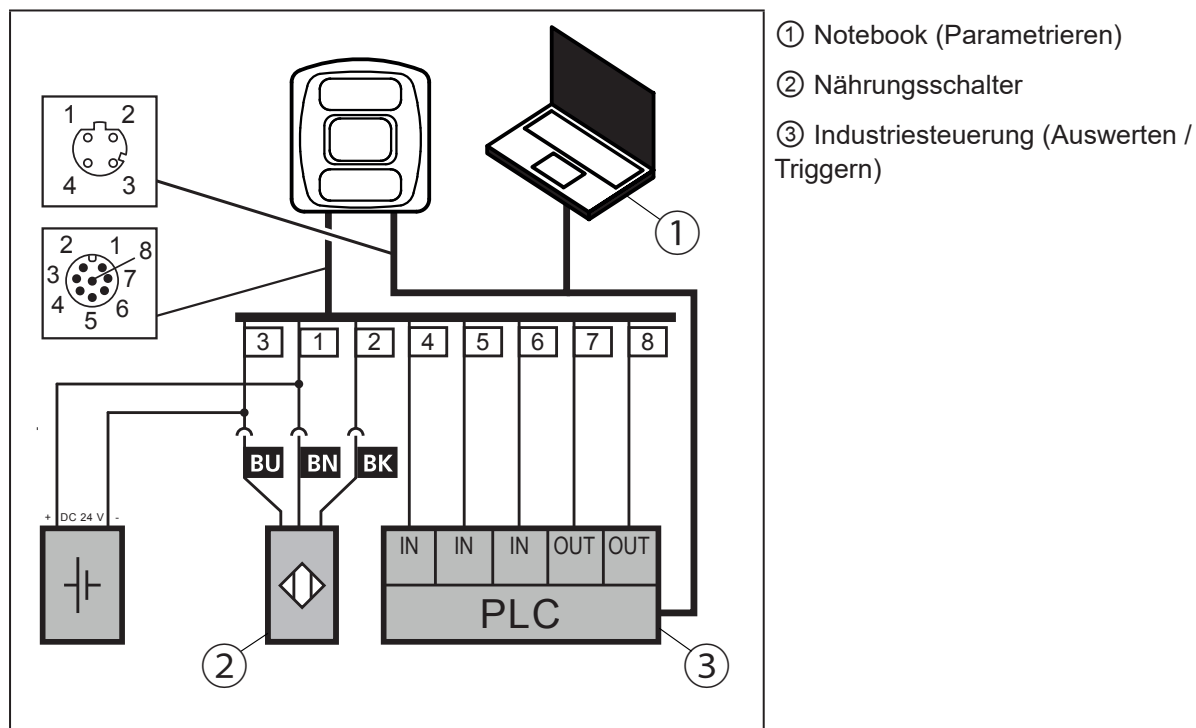
7.2.1 Bildaufnahme mit Nährungsschalter triggern

Das Gerät kann extern getriggert werden:

- über Ethernet
- über einen Nährungsschalter, verbunden mit dem Triggereingang

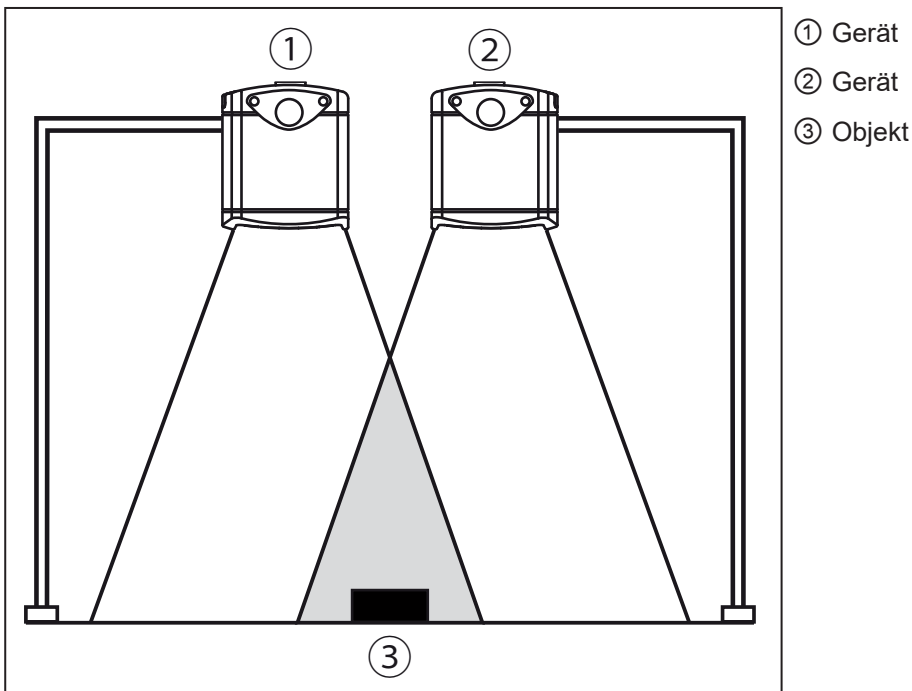
DE

Die folgende Abbildung zeigt die Verdrahtung des Gerätes mit einem Nährungsschalter.



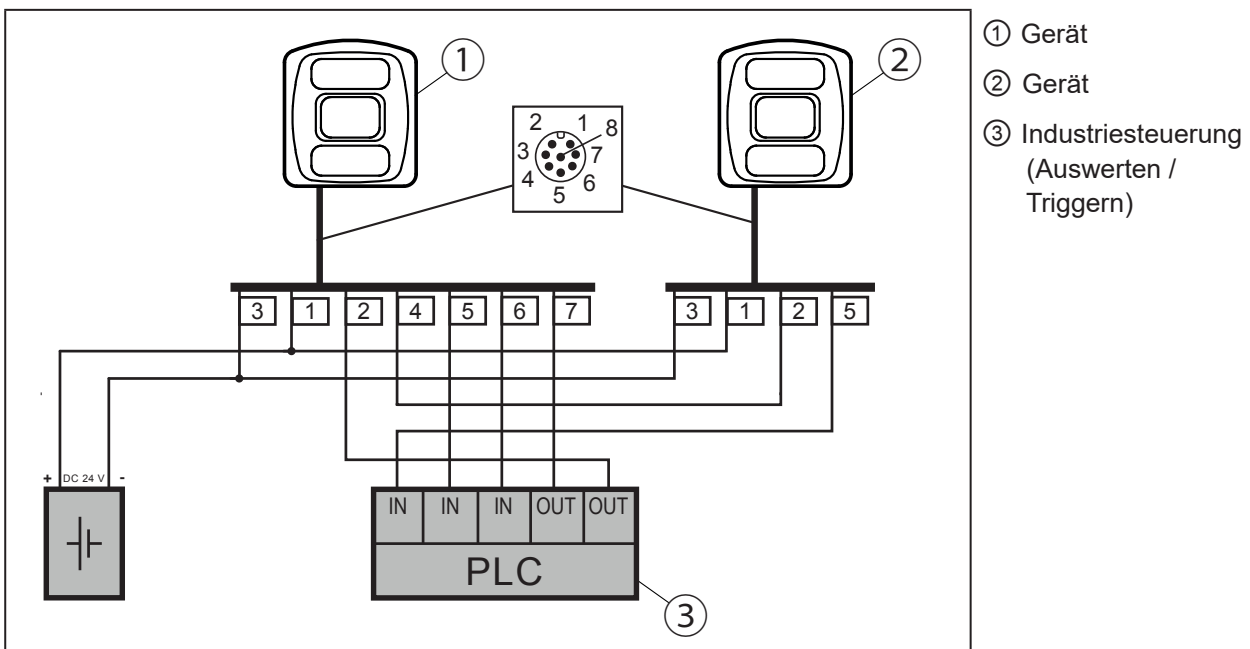
7.2.2 Mehrere Geräte nebeneinander verwenden

Nebeneinander montierte Geräte können Messfehler verursachen durch das zeitgleiche Belichten.



Die Messfehler können auf zwei Arten vermieden werden:

- Geräte über HW-Trigger kaskadieren
Beim Kaskadieren triggert eine Steuerung die Bildaufnahme von Gerät ① (siehe Abb. unten). Nach Abschluss der Bildaufnahme triggert Gerät ① selbstständig Gerät ②. Dabei gibt Pin 4 von Gerät ① den Gerätestatus "Bildaufnahme beendet" aus. Gerät ② meldet den Sequenzabschluss der Industriesteuerung ③.



- Unterschiedliche Frequenzkanäle verwenden
Mit der Software ifm Vision Assistant kann jedem Gerät ein eigener Frequenzkanal zugeordnet werden. Die unterschiedlichen Frequenzkanäle reduzieren das Auftreten von Messfehlern.

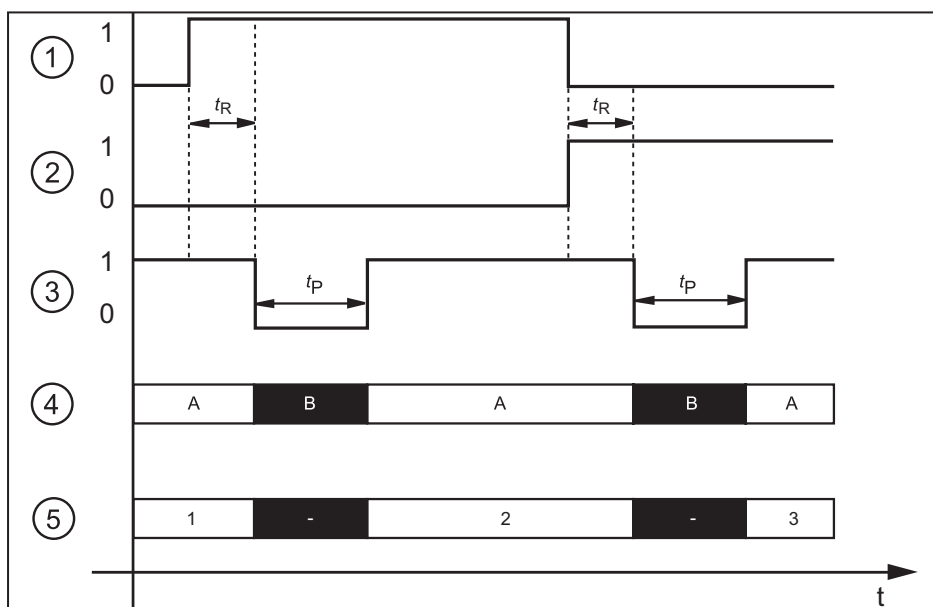


Die Software ifm Vision Assistant steht kostenlos im Internet zur Verfügung: www.ifm.com

7.3 Statische Anwendungsumschaltung

Auf das Gerät lassen sich bis zu 32 verschiedene Inspektionsaufgaben speichern. Bei entsprechender Konfiguration können die ersten vier Anwendungen über die beiden Schalteingänge ausgewählt werden.

Eingang 2	Eingang 1	Anwendung Nr.
0	0	1
0	1	2
1	0	3
1	1	4



Beispiel: Umschaltung Anwendung 1 → Anwendung 2 → Anwendung 3

①	Schalteingang 1 = 0 → 1 → 0
②	Schalteingang 2 = 0 → 0 → 1
③	Ausgang READY
④	Triggereingang
	A: Trigger erlaubt
	B: Trigger gesperrt
⑤	ID-Nummer der aktiven Anwendung

Beim Umschalten der Anwendungen sind die Überwachungszeit t_R und die Trigger-Sperrzeit t_P zu berücksichtigen.

Überwachungszeit t_R : Die Anwendungsumschaltung beginnt erst, wenn nach einer Flankenänderung der Zustand an beiden Schalteingängen für 20 ms stabil bleibt.

Trigger-Sperrzeit t_P : Während der Anwendungsumschaltung ist der Triggereingang gesperrt. Die Sperrzeit ist abhängig von:

- der Anzahl der Anwendungen auf dem Gerät
- der Anzahl der Modelle in der zu aktivierenden Anwendung



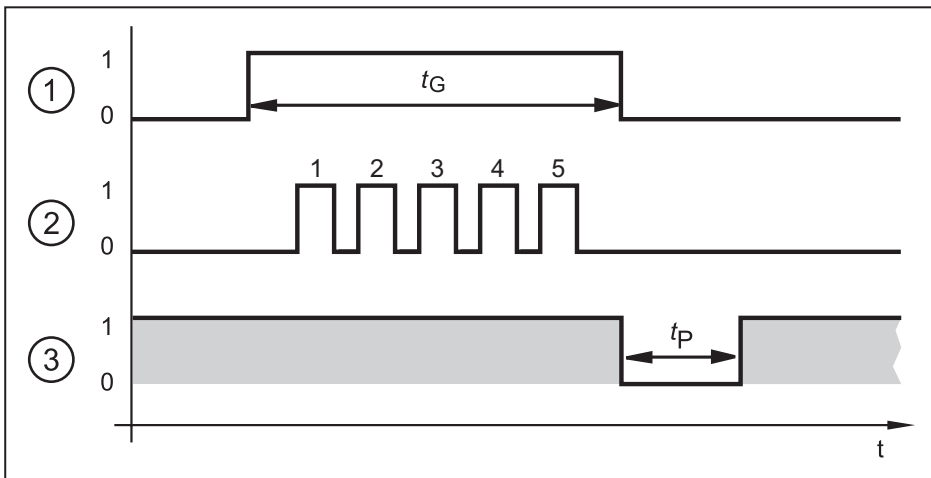
In der Abbildung oben ist die Ausgangslogik PNP eingestellt (Voreinstellung). Die Ausgangslogiken PNP und NPN verhalten sich umgekehrt zueinander:

- Ausgangslogik PNP: Bei einem High-Signal (1) liegt Spannung an.
- Ausgangslogik NPN: Bei einem Low-Signal (0) liegt Spannung an.

Nähere Informationen zur Konfiguration der Anwendungsumschaltung finden Sie im Softwarehandbuch des Gerätes. www.ifm.com

7.4 Pulsgesteuerte Anwendungsumschaltung

Alternativ zur statischen Umschaltung kann die Auswahl der Anwendung pulsgesteuert erfolgen.



①	Gate-Signal, Schalteingang 1 = 0 → 1 → 0 (t_G = Signal aktiv)
②	Puls-Signal, Schalteingang 2 oder Triggereingang = 0 → 5 Pulse → 0
③	Ausgang READY

Während an Schalteingang 1 ein aktives Signal anliegt (Gate-Signal), zählt das Gerät eingehende Pulse und aktiviert die entsprechende Anwendung.

Anzahl der Pulse = ID-Nummer der Anwendung.

Als Pulseingang kann entweder der Schalteingang 2 oder der Triggereingang des Gerätes verwendet werden.



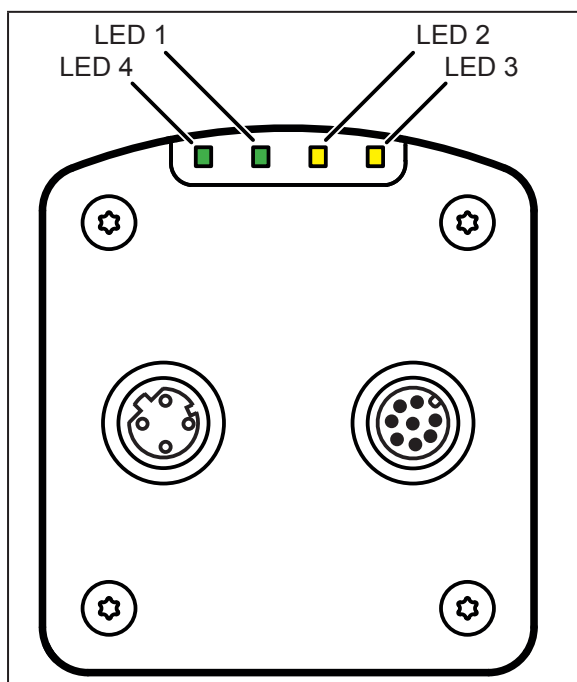
In der Abbildung oben ist die Ausgangslogik PNP eingestellt (Voreinstellung). Die Ausgangslogiken PNP und NPN verhalten sich umgekehrt zueinander:

- Ausgangslogik PNP: Bei einem High-Signal (1) liegt Spannung an.
- Ausgangslogik NPN: Bei einem Low-Signal (0) liegt Spannung an.

Nähere Informationen zur Konfiguration der Anwendungsumschaltung finden Sie im Softwarehandbuch des Gerätes. www.ifm.com

8. Anzeigeelemente

Das Gerät signalisiert über die Anzeigeelemente LED 1 - 4 den aktuellen Betriebszustand.



DE

LED 4 (Ethernet)	LED 1 (Power)	LED 2 (Out 1)	LED 3 (Out 2)	Beschreibung
	leuchtet			Gerät ist betriebsbereit, Versorgungsspannung liegt an
	blinkt mit 0,5 Hz			Gerät ist nicht parametrierbar oder die Parametrierung wurde nicht auf das Gerät geladen On Off
	blinkt 2x mit 0,5 Hz			Gerät ist im Parametriemodus On Off
		leuchtet		Schaltausgang 1 ist geschaltet
		blinkt mit 8 Hz		Schaltausgang 1 hat einen Kurzschluss
			leuchtet	Schaltausgang 2 ist geschaltet
			blinkt mit 8 Hz	Schaltausgang 2 hat einen Kurzschluss
leuchtet				Ethernet ist verbunden
blinkt				Ethernet überträgt Daten
aus				Ethernet ist nicht verbunden
		blinkt mit 8 Hz	blinkt mit 8 Hz	Gerät meldet internen Fehler
		blinkt mit 2 Hz	blinkt mit 2 Hz	Gerät meldet behebbaren Fehler. Die Fehlermeldung kann über Ethernet ausgelesen werden
	Lauflicht ⇒			Gerät bootet
	Lauflicht ⇐			Gerät führt Firmware-Update aus

9. Inbetriebnahme

Durch Einschalten der Versorgungsspannung wird das Gerät in Betrieb genommen. Nach 15 Sekunden befindet sich das Gerät im Auswertebetrieb, in dem gespeicherte Anwendungen ausgeführt werden. Die Anzeigeelemente signalisieren den aktuellen Betriebszustand (→ 8).



Auf dem Gerät können bis zu 32 Anwendungen gespeichert werden. Eine Anwendung kann auf verschiedene Arten aktiviert werden:

- Software ifm Vision Assistant
- Prozessschnittstellen-Kommando
- Schalteingang 1 und 2
- Schalteingang 1 und Triggereingang

9.1 Gerät parametrieren

Das Gerät wird mit der Software ifm Vision Assistant parametriert (→ siehe Softwarehandbuch).



Das Verwenden der Software ifm Vision Assistant und detaillierte Informationen zum Messprinzip des Gerätes sind im Softwarehandbuch beschrieben.

Die Software ifm Vision Assistant steht kostenlos im Internet zur Verfügung:

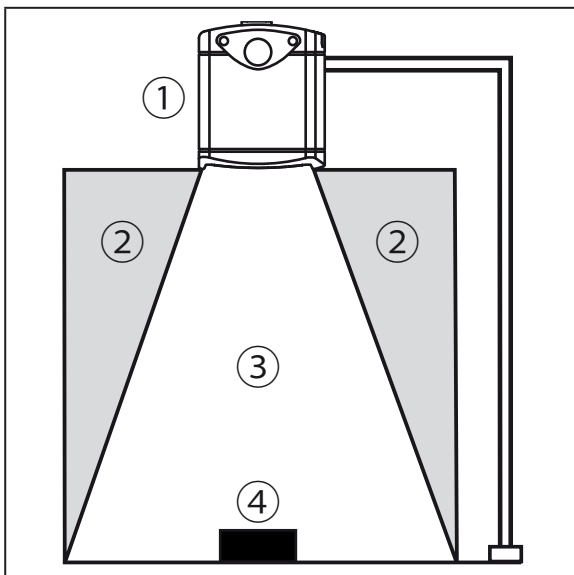
www.ifm.com

Das Softwarehandbuch ist im Internet verfügbar:

www.ifm.com

9.2 Objekt detektieren

Im Folgenden wird beschrieben, welche Bedingungen zu einer hohen Erkennungsrate von Objekten führen.



- ① Gerät
- ② Einflussbereich
- ③ Sichtfeld
- ④ Objekt

Ein Objekt ④ wird optimal detektiert, wenn die folgenden Bedingungen erfüllt werden:

- Objekt ist im Sichtfeld ③ platziert
- Objekt ist vom Gerät ① das nächste sichtbare Objekt
- Einflussbereich ② ist frei von Gegenständen (Aufbauten etc.)
- Frontscheibe des Gerätes ist frei von Verschmutzungen.



Werden die Bedingungen nicht eingehalten, können Messfehler auftreten.

9.3 Prozesswerte senden

9.3.1 Prozesswerte der Vollständigkeitsüberwachung über EtherNet/IP senden

Das Gerät kann die Prozesswerte über den Feldbus EtherNet/IP an eine SPS senden. Die Prozesswerte werden im ifm Vision Assistant als Ausgabe-String wie folgt angezeigt:

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

DE



Es kann immer nur ein Feldbus aktiv sein. Der Feldbus ist einstellbar (→ Softwarehandbuch).

Im Ausgabe-String werden die Prozesswerte durch ein Semikolon getrennt. Der Ausgabe-String wird in der angezeigten Reihenfolge an eine SPS übertragen.



Beachten Sie die folgenden Hinweise beim Übertragen des Ausgabe-Strings an eine SPS:

- Die Bytes 0 bis 7 sind Teil des Ausgabe-Strings. Sie werden nicht im ifm Vision Assistant angezeigt (siehe Screenshot oben).
- Im Ausgabe-String enthaltene Semikolon ";" werden nicht übertragen.
- Float-Werte werden vor dem Übertragen in 16-Bit Integer umgewandelt.
- Alle Zahlenwerte werden vor dem Übertragen in 16-Bit Integer umgewandelt.

Der Ausgabe-String setzt sich wie folgt zusammen:

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

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
0	2#0000_0000	Binär	1.5		Gespiegeltes Kommandowort	• Bit 1.5 zeigt ein erfolgreiches Trigger-Kommando an
1	2#0010_0000	Binär				
2	2#0000_0000	Dezimal			Synchrone / asynchrone Nachrichtenennung	
3	2#0000_0000	Dezimal				
4	30	Dezimal	30		Nachrichtenzähler	• Das Gerät hat 30 Nachrichten empfangen • Zählt um 1 hoch, bei jeder Aktion (Trigger, gesendete Nachricht etc.)
5	0	Dezimal				
6	0	Dezimal			Reserviert	
7	0	Dezimal				
8	s	ASCII	star		Start-String	
9	t	ASCII				
10	a	ASCII				
11	r	ASCII				
12	0	Dezimal	0		Status von allen ROIs (0 = schlecht, 1 = gut)	Zeigt den Status der Vollständigkeitsüberwachung an
13	0	Dezimal				

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
14	0	Dezimal				Bei aktivierter Lagenachführung werden Byte 14 und 15 von jener belegt. 0 = Lage wird nicht nachgeführt 1 = Lage wird nachgeführt Alle folgenden Daten verschieben sich um 2 Bytes; d.h. die 1. ROI ID beginnt mit Byte 16 und 17.
15	0	Dezimal	0		ROI ID	
16	0	Dezimal	0		ROI Status	ROI Status: 0 = gut 1 = Referenzebene nicht eingelernt 2 = Einlernen fehlgeschlagen 3 = Referenzebene ungültig 4 = Keine gültigen Pixel 5 = Referenzebene enthält keine gültigen Pixel 6 = Überfüllt 7 = Unterfüllt
17	0	Dezimal	0			
18	0	Dezimal	0	mm	ROI Wert	
19	0	Dezimal	0			
20	1	Dezimal	1		ROI ID	
21	0	Dezimal				
22	7	Dezimal	7		ROI Status	
23	0	Dezimal				
24	-67	Dezimal	-67	mm	ROI Wert	
25	-1	Dezimal				
26	2	Dezimal	2		ROI ID	
27	0	Dezimal				
28	6	Dezimal	6		ROI Status	
29	0	Dezimal				
30	14	Dezimal	14	mm	ROI Wert	
31	0	Dezimal				
32	3	Dezimal	3		ROI ID	
33	0	Dezimal				
34	0	Dezimal	0		ROI Status	
35	0	Dezimal				
36	0	Dezimal	0	mm	ROI Wert	
37	0	Dezimal				
38	s	ASCII	stop		Stop-String	
39	t	ASCII				
40	o	ASCII				
41	p	ASCII				



Das fehlerhafte Ausführen eines Kommandos führt zu folgendem Zustand:

- Error Bit = 1
- Gespiegeltes Kommandowort wird angezeigt
- Asynchrones Nachrichtenbit = 0
- Asynchrone Nachrichtenkennung = 0
- Nachrichtenzähler zählt um 1 hoch

9.3.2 Prozesswerte der Vollständigkeitsüberwachung über PROFINET senden

Das Gerät kann die Prozesswerte über den Feldbus PROFINET an eine SPS senden. Die Prozesswerte werden im ifm Vision Assistant als Ausgabe-String wie folgt angezeigt:

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

DE



Es kann immer nur ein Feldbus aktiv sein. Der Feldbus ist einstellbar (→ Softwarehandbuch).

Im Ausgabe-String werden die Prozesswerte durch ein Semikolon getrennt. Der Ausgabe-String wird in der angezeigten Reihenfolge an eine SPS übertragen.



Beachten Sie die folgenden Hinweise beim Übertragen des Ausgabe-Strings an eine SPS:

- Die Bytes 0 bis 7 sind Teil des Ausgabe-Strings. Sie werden nicht im ifm Vision Assistant angezeigt (siehe Screenshot oben).
- Im Ausgabe-String enthaltene Semikolon ";" werden nicht übertragen.
- Float-Werte werden vor dem Übertragen in 16-Bit Integer umgewandelt.
- Alle Zahlenwerte werden vor dem Übertragen in 16-Bit Integer umgewandelt.

Der Ausgabe-String setzt sich wie folgt zusammen:

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

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
0	2#0010_0000	Binär	0.5		Gespiegelter Kommandowort	• Bit 0.5 zeigt ein erfolgreiches Trigger-Kommando an
1	2#0000_0000	Binär				
2	2#0000_0000	Dezimal			Synchrone / asynchrone Nachrichtenennung	
3	2#0000_0000	Dezimal				
4	0	Dezimal	30		Nachrichtenzähler	• Das Gerät hat 30 Nachrichten empfangen • Zählt um 1 hoch, bei jeder Aktion (Trigger, gesendete Nachricht etc.)
5	30	Dezimal				
6	0	Dezimal			Reserviert	
7	0	Dezimal				
8	s	ASCII	star		Start-String	
9	t	ASCII				
10	a	ASCII				
11	r	ASCII				
12	0	Dezimal	0		Status von allen ROIs (0 = schlecht, 1 = gut)	Zeigt den Status der Vollständigkeitsüberwachung an
13	0	Dezimal				
14	0	Dezimal	0		ROI ID	Bei aktivierter Lagenachführung werden Byte 14 und 15 von jener belegt. 0 = Lage wird nicht nachgeführt 1 = Lage wird nachgeführt Alle folgenden Daten verschieben sich um 2 Bytes; d.h. die 1. ROI ID beginnt mit Byte 16 und 17.
15	0	Dezimal				

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
16	0	Dezimal	0		ROI Status	ROI Status: 0 = gut 1 = Referenzebene nicht eingelernt 2 = Einlernen fehlgeschlagen 3 = Referenzebene ungültig 4 = Keine gültigen Pixel 5 = Referenzebene enthält keine gültigen Pixel 6 = Überfüllt 7 = Unterfüllt
17	0	Dezimal				
18	0	Dezimal	0	mm	ROI Wert	
19	0	Dezimal				
20	0	Dezimal	1		ROI ID	
21	1	Dezimal				
22	0	Dezimal	7		ROI Status	
23	7	Dezimal				
24	-1	Dezimal	-67	mm	ROI Wert	
25	-67	Dezimal				
26	0	Dezimal	2		ROI ID	
27	2	Dezimal				
28	0	Dezimal	6		ROI Status	
29	6	Dezimal				
30	0	Dezimal	14	mm	ROI Wert	
31	14	Dezimal				
32	0	Dezimal	3		ROI ID	
33	3	Dezimal				
34	0	Dezimal	0		ROI Status	
35	0	Dezimal				
36	0	Dezimal	0	mm	ROI Wert	
37	0	Dezimal				
38	s	ASCII	stop		Stop-String	
39	t	ASCII				
40	o	ASCII				
41	p	ASCII				



Das fehlerhafte Ausführen eines Kommandos führt zu folgendem Zustand:

- Error Bit = 1
- Gespiegeltes Kommandowort wird angezeigt
- Asynchrones Nachrichtenbit = 0
- Asynchrone Nachrichtenkennung = 0
- Nachrichtenzähler zählt um 1 hoch

9.3.3 Prozesswerte der Vollständigkeitsüberwachung über TCP/IP senden

Das Gerät kann die Prozesswerte über das Protokoll TCP/IP an eine SPS senden. Die Prozesswerte werden im ifm Vision Assistant als Ausgabe-String wie folgt dargestellt:

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

DE

Im Ausgabe-String werden die Prozesswerte durch ein Semikolon getrennt. Der Ausgabe-String wird in der angezeigten Reihenfolge an eine SPS übertragen.



Beachten Sie die folgenden Hinweise beim Übertragen des Ausgabe-Strings an eine SPS:

- Im Ausgabe-String enthaltene Semikolon ";" werden nicht übertragen.
- Alle Zahlenwerte werden vor dem Übertragen in 16-Bit Integer umgewandelt.

Der Ausgabe-String setzt sich wie folgt zusammen (Datentyp: ASCII):

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

Prozesswert	Einheit	Beschreibung
star		Start-String
0		Status von allen ROIs (0 = schlecht, 1 = gut)
00 0 +0.000	m	ROI ID ROI Status ROI Wert
01 7 -0.068	m	ROI ID ROI Status ROI Wert
02 6 +0.013	m	ROI ID ROI Status ROI Wert
03 0 +0.001	m	ROI ID ROI Status ROI Wert
stop		Stop-String

ROI Status:

0 = gut

1 = Referenzebene nicht eingelesen

2 = Einlernen fehlgeschlagen

3 = Referenzebene ungültig

4 = Keine gültigen Pixel

5 = Referenzebene enthält keine gültigen Pixel

6 = Überfüllt

7 = Unterfüllt

9.3.4 Prozesswerte der Objektvermessung über EtherNet/IP senden

Das Gerät kann die Prozesswerte über den Feldbus EtherNet/IP an eine SPS senden. Die Prozesswerte werden im ifm Vision Assistant als Ausgabe-String wie folgt angezeigt:

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



Es kann immer nur ein Feldbus aktiv sein. Der Feldbus ist einstellbar (→ Softwarehandbuch).

Im Ausgabe-String werden die Prozesswerte durch ein Semikolon getrennt. Der Ausgabe-String wird in der angezeigten Reihenfolge an eine SPS übertragen.



Beachten Sie die folgenden Hinweise beim Übertragen des Ausgabe-Strings an eine SPS:

- Der Ausgabe-String ist einstellbar. Welche Prozesswerte übertragen werden sollen, kann im ifm Vision Assistant eingestellt werden.
- Die Bytes 0 bis 7 sind Teil des Ausgabe-Strings. Sie werden nicht im ifm Vision Assistant angezeigt (siehe Screenshot oben).
- Im Ausgabe-String enthaltene Semikolon ";" werden nicht übertragen.
- Float-Werte werden vor dem Übertragen in 16-Bit Integer umgewandelt.
- Alle Zahlenwerte werden vor dem Übertragen in 16-Bit Integer umgewandelt.

Der Ausgabe-String setzt sich wie folgt zusammen:

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

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
0	2#0000_0000	Binär	1.5		Gespiegeltes Kommandowort	• Bit 1.5 zeigt ein erfolgreiches Trigger-Kommando an
1	2#0010_0000	Binär				
2	2#0000_0000	Binär			Synchrone / asynchrone Nachrichtenennung	
3	2#0000_0000	Binär				
4	2#0000_0011	Binär	3		Nachrichtenzähler	• Das Gerät hat 3 Nachrichten empfangen • Zählt um 1 hoch, bei jeder Aktion (Trigger, gesendete Nachricht etc.)
5	2#0000_0000	Binär				
6	2#0000_0000	Binär			Reserviert	
7	2#0000_0000	Binär				
8	s	ASCII	star		Start-String	
9	t	ASCII				
10	a	ASCII				
11	r	ASCII				
12	2#0000_0001	Binär	1		Ergebnis-Bit	0 = keine Box gefunden 1 = Box gefunden
13	2#0000_0000	Binär				
14	104	Dezimal	104	mm	Breite	
15	0	Dezimal				
16	88	Dezimal	88	mm	Höhe	
17	0	Dezimal				
18	108	Dezimal	109	mm	Länge	
19	0	Dezimal				
20	21	Dezimal	21		Koordinate x	
21	0	Dezimal				
22	-11	Dezimal	-11		Koordinate y	
23	-1	Dezimal				

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
24	-124	Dezimal	389		Koordinate z	
25	1	Dezimal				
26	-98	Dezimal	158		Rotationsgrad	
27	0	Dezimal				
28	97	Dezimal	97		Qualität Breite	
29	0	Dezimal				
30	93	Dezimal	94		Qualität Höhe	
31	0	Dezimal				
32	97	Dezimal	97		Qualität Länge	
33	0	Dezimal				
34	s	ASCII	stop		Stop-String	
35	t	ASCII				
36	o	ASCII				
37	p	ASCII				

DE



Das fehlerhafte Ausführen eines Kommandos führt zu folgendem Zustand:

- Error Bit = 1
- Gespiegeltes Kommandowort wird angezeigt
- Asynchrones Nachrichtenbit = 0
- Asynchrone Nachrichtenkennung = 0
- Nachrichtenzähler zählt um 1 hoch

9.3.5 Prozesswerte der Objektvermessung über PROFINET senden

Das Gerät kann die Prozesswerte über den Feldbus PROFINET an eine SPS senden. Die Prozesswerte werden im ifm Vision Assistant als Ausgabe-String wie folgt angezeigt:

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



Es kann immer nur ein Feldbus aktiv sein. Der Feldbus ist einstellbar (→ Softwarehandbuch).

Im Ausgabe-String werden die Prozesswerte durch ein Semikolon getrennt. Der Ausgabe-String wird in der angezeigten Reihenfolge an eine SPS übertragen.



Beachten Sie die folgenden Hinweise beim Übertragen des Ausgabe-Strings an eine SPS:

- Der Ausgabe-String ist einstellbar. Welche Prozesswerte übertragen werden sollen, kann im ifm Vision Assistant eingestellt werden.
- Die Bytes 0 bis 7 sind Teil des Ausgabe-Strings. Sie werden nicht im ifm Vision Assistant angezeigt (siehe Screenshot oben).
- Im Ausgabe-String enthaltene Semikolon ";" werden nicht übertragen.
- Float-Werte werden vor dem Übertragen in 16-Bit Integer umgewandelt.
- Alle Zahlenwerte werden vor dem Übertragen in 16-Bit Integer umgewandelt.

Der Ausgabe-String setzt sich wie folgt zusammen:

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

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
0	2#0010_0000	Binär	0.5		Gespiegeltes Kommandowort	• Bit 0.5 zeigt ein erfolgreiches Trigger-Kommando an
1	2#0000_0000	Binär				
2	2#0000_0000	Binär			Synchrone / asynchrone Nachrichtenennung	
3	2#0000_0000	Binär				
4	2#0000_0000	Binär				
5	2#0000_0011	Binär	3		Nachrichtenzähler	• Das Gerät hat 3 Nachrichten empfangen • Zählt um 1 hoch, bei jeder Aktion (Trigger, gesendete Nachricht etc.)
6	2#0000_0000	Binär	star		Start-String	
7	2#0000_0000	Binär				
8	s	ASCII				
9	t	ASCII				
10	a	ASCII	1		Ergebnis-Bit	0 = keine Box gefunden 1 = Box gefunden
11	r	ASCII				
12	2#0000_0000	Binär	104	mm	Breite	
13	2#0000_0001	Binär				
14	0	Dezimal	88	mm	Höhe	
15	104	Dezimal				
16	0	Dezimal	109	mm	Länge	
17	88	Dezimal				
18	0	Dezimal	21		Koordinate x	
19	109	Dezimal				
20	0	Dezimal	-11		Koordinate y	
21	21	Dezimal				
22	-1	Dezimal				
23	-11	Dezimal				

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
24	1	Dezimal	389		Koordinate z	
25	-124	Dezimal				
26	0	Dezimal	158		Rotationsgrad	
27	-98	Dezimal				
28	0	Dezimal	97		Qualität Breite	
29	97	Dezimal				
30	0	Dezimal	94		Qualität Höhe	
31	94	Dezimal				
32	0	Dezimal	97		Qualität Länge	
33	97	Dezimal				
34	s	ASCII	stop		Stop-String	
35	t	ASCII				
36	o	ASCII				
37	p	ASCII				

DE



Das fehlerhafte Ausführen eines Kommandos führt zu folgendem Zustand:

- Error Bit = 1
- Gespiegeltes Kommandowort wird angezeigt
- Asynchrones Nachrichtenbit = 0
- Asynchrone Nachrichtenkennung = 0
- Nachrichtenzähler zählt um 1 hoch

9.3.6 Prozesswerte der Objektvermessung über TCP/IP senden

Das Gerät kann die Prozesswerte über das Protokoll TCP/IP an eine SPS senden. Im ifm Vision Assistant ist wählbar, welche Prozesswerte gesendet werden. Die Prozesswerte werden im ifm Vision Assistant als Ausgabe-String wie folgt dargestellt:

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

Im Ausgabe-String werden die Prozesswerte durch ein Semikolon getrennt. Der Ausgabe-String wird in der angezeigten Reihenfolge an eine SPS übertragen.



Beachten Sie die folgenden Hinweise beim Übertragen des Ausgabe-Strings an eine SPS:

- Im Ausgabe-String enthaltene Semikolon ";" werden nicht übertragen.
- Alle Zahlenwerte werden vor dem Übertragen in 16-Bit Integer umgewandelt.

Der Ausgabe-String setzt sich wie folgt zusammen (Datentyp: ASCII):

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

Prozesswert	Einheit	Beschreibung
star		Start-String
1		Objekt gefunden
0.104	m	Breite
0.088	m	Höhe
0.109	m	Länge
+0.021		Koordinate x
-0.011		Koordinate y
+0.389		Koordinate z
158		Rotationsgrad
097		Qualität Breite
094		Qualität Höhe
097		Qualität Länge
stop		Stop-String

9.3.7 Prozesswerte der Füllstandsmessung über EtherNet/IP senden

Das Gerät kann die Prozesswerte über den Feldbus EtherNet/IP an eine SPS senden. Die Prozesswerte werden im ifm Vision Assistant als Ausgabe-String wie folgt angezeigt:

0070

DE



Es kann immer nur ein Feldbus aktiv sein. Der Feldbus ist einstellbar (→ Softwarehandbuch).

Der Ausgabe-String wird in der angezeigten Reihenfolge an eine SPS übertragen.



Beachten Sie die folgenden Hinweise beim Übertragen des Ausgabe-Strings an eine SPS:

- Die Bytes 0 bis 7 sind Teil des Ausgabe-Strings. Sie werden nicht im ifm Vision Assistant angezeigt (siehe Screenshot oben).
- Im Ausgabe-String enthaltene Semikolon ";" werden nicht übertragen.
- Float-Werte werden vor dem Übertragen in 16-Bit Integer umgewandelt.
- Alle Zahlenwerte werden vor dem Übertragen in 16-Bit Integer umgewandelt.

Der Ausgabe-String setzt sich wie folgt zusammen:

0070

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
0	2#0000_0000	Binär	1.5		Gespiegelter Kommandowort	Bit 1.5 zeigt ein erfolgreiches Trigger-Kommando an
1	2#0010_0000	Binär				
2	2#0000_0000	Dezimal			Synchrone / asynchrone Nachrichtenennung	
3	2#0000_0000	Dezimal				
4	30	Dezimal	30		Nachrichtenzähler	• Das Gerät hat 30 Nachrichten empfangen • Zählt um 1 hoch, bei jeder Aktion (Trigger, gesendete Nachricht etc.)
5	0	Dezimal				
6	0	Dezimal			Reserviert	
7	0	Dezimal				
8	0	Dezimal	0		Status von allen ROIs (0 = schlecht, 1 = gut)	Zeigt den Status der Füllstandsmessung an
9	0	Dezimal				
10	0	Dezimal	0		ROI ID	ROI Status: 0 = gut 6 = Überfüllt 7 = Unterfüllt
11	0	Dezimal				
12	7	Dezimal	7		ROI Status	
13	0	Dezimal				
14	0	Dezimal	0	mm	ROI Wert	
15	0	Dezimal				



Das fehlerhafte Ausführen eines Kommandos führt zu folgendem Zustand:

- Error Bit = 1
- Gespiegelter Kommandowort wird angezeigt
- Asynchrones Nachrichtenbit = 0
- Asynchrone Nachrichtenennung = 0
- Nachrichtenzähler zählt um 1 hoch

9.3.8 Prozesswerte der Füllstandsmessung über PROFINET senden

Das Gerät kann die Prozesswerte über den Feldbus PROFINET an eine SPS senden. Die Prozesswerte werden im ifm Vision Assistant als Ausgabe-String wie folgt angezeigt:




Es kann immer nur ein Feldbus aktiv sein. Der Feldbus ist einstellbar (→ Softwarehandbuch).

Der Ausgabe-String wird in der angezeigten Reihenfolge an eine SPS übertragen.



Beachten Sie die folgenden Hinweise beim Übertragen des Ausgabe-Strings an eine SPS:

- Die Bytes 0 bis 7 sind Teil des Ausgabe-Strings. Sie werden nicht im ifm Vision Assistant angezeigt (siehe Screenshot oben).
- Im Ausgabe-String enthaltene Semikolon ";" werden nicht übertragen.
- Float-Werte werden vor dem Übertragen in 16-Bit Integer umgewandelt.
- Alle Zahlenwerte werden vor dem Übertragen in 16-Bit Integer umgewandelt.

Der Ausgabe-String setzt sich wie folgt zusammen:

0070

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
0	2#0010_0000	Binär	0.5		Gespiegelter Kommandowort	Bit 0.5 zeigt ein erfolgreiches Trigger-Kommando an
1	2#0000_0000	Binär				
2	2#0000_0000	Dezimal			Synchrone / asynchrone Nachrichtenennung	
3	2#0000_0000	Dezimal				
4	0	Dezimal	30		Nachrichtenzähler	• Das Gerät hat 30 Nachrichten empfangen • Zählt um 1 hoch, bei jeder Aktion (Trigger, gesendete Nachricht etc.)
5	30	Dezimal				
6	0	Dezimal			Reserviert	
7	0	Dezimal				
8	0	Dezimal	0		Status von allen ROIs (0 = schlecht, 1 = gut)	Zeigt den Status der Füllstandsmessung an
9	0	Dezimal				
10	0	Dezimal	0		ROI ID	ROI Status: 0 = gut 6 = Überfüllt 7 = Unterfüllt
11	0	Dezimal				
12	0	Dezimal	7		ROI Status	
13	7	Dezimal				
14	0	Dezimal	0	mm	ROI Wert	
15	0	Dezimal				



Das fehlerhafte Ausführen eines Kommandos führt zu folgendem Zustand:

- Error Bit = 1
- Gespiegelter Kommandowort wird angezeigt
- Asynchrones Nachrichtenbit = 0
- Asynchrone Nachrichtenennung = 0
- Nachrichtenzähler zählt um 1 hoch

9.3.9 Prozesswerte der Füllstandsmessung über TCP/IP senden

Das Gerät kann die Prozesswerte über das Protokoll TCP/IP an eine SPS senden. Die Prozesswerte werden im ifm Vision Assistant als Ausgabe-String wie folgt dargestellt:

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

DE

Im Ausgabe-String werden die Prozesswerte durch ein Semikolon getrennt. Der Ausgabe-String wird in der angezeigten Reihenfolge an eine SPS übertragen.



Beachten Sie die folgenden Hinweise beim Übertragen des Ausgabe-Strings an eine SPS:

- Im Ausgabe-String enthaltene Semikolon ";" werden nicht übertragen.
- Alle Zahlenwerte werden vor dem Übertragen in 16-Bit Integer umgewandelt.

Der Ausgabe-String setzt sich wie folgt zusammen (Datentyp: ASCII):

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

Prozesswert	Einheit	Beschreibung
star		Start-String
0		Status von allen ROIs (0 = schlecht, 1 = gut)
00		ROI ID
7		ROI Status
+0.000	m	ROI Wert
stop		Stop-String

ROI Status:
0 = gut
6 = Überfüllt
7 = Unterfüllt

9.3.10 Prozesswerte der Roboter Greifernavigation über EtherNet/IP senden

Das Gerät kann die Prozesswerte über den Feldbus EtherNet/IP an eine SPS senden.



Es kann immer nur ein Feldbus aktiv sein. Der Feldbus ist einstellbar (→ Softwarehandbuch).

Im Ausgabe-String werden die Prozesswerte durch ein Semikolon getrennt. Der Ausgabe-String wird in der angezeigten Reihenfolge an eine SPS übertragen.



Beachten Sie die folgenden Hinweise beim Übertragen des Ausgabe-Strings an eine SPS:

- Die Bytes 0 bis 7 sind Teil des Ausgabe-Strings. Sie werden nicht im ifm Vision Assistant angezeigt.
- Die Bytes 14 bis 35 wiederholen sich für jedes unter "Anzahl der Objekte" eingestellte Objekt (maximal 10 Wiederholungen).
- Im Ausgabe-String enthaltene Semikolon ";" werden nicht übertragen.
- Float-Werte werden vor dem Übertragen in 16-Bit Integer umgewandelt.
- Alle Zahlenwerte werden vor dem Übertragen in 16-Bit Integer umgewandelt.

Der Ausgabe-String setzt sich wie folgt zusammen:

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

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
0	2#0010_0000	Binär	0.5		Gespiegelter Kommandowort	• Bit 0.5 zeigt ein erfolgreiches Trigger-Kommando an
1	2#0000_0000	Binär				
2	2#0000_0000	Binär	3		Synchrone / asynchrone Nachrichtenennung	
3	2#0000_0000	Binär				
4	2#0000_0000	Binär				
5	2#0000_0011	Binär	3		Nachrichtenzähler	• Das Gerät hat 3 Nachrichten empfangen • Zählt um 1 hoch, bei jeder Aktion (Trigger, gesendete Nachricht etc.)
6	2#0000_0000	Binär				
7	2#0000_0000	Binär			Reserviert	
8	0	Dezimal	0		Fehler	Fehler: 0 = kein Fehler 1 = undefinierter Fehler 2 = kein Objekt gefunden
9	0	Dezimal				
10	1	Dezimal	01		Anzahl der Objekte	Anzahl der gefundenen Objekte.
11	0	Dezimal				
12	8	Dezimal	08		Anzahl der Objektkandidaten	Anzahl der gefundenen und geprüften Objektkandidaten.
13	0	Dezimal				
14	1	Binär	1		Objekt gefunden	0 = kein Objekt gefunden 1 = Objekt gefunden
15	0	Binär				
16	338	Dezimal	338	mm	Breite	Die breiteste Abmessung der Objektoberfläche.
17	0	Dezimal				
18	142	Dezimal	142	mm	Höhe	Die Objekthöhe relativ zur Grundplatte.
19	0	Dezimal				
20	452	Dezimal	452	mm	Länge	Die längste Abmessung der Objektoberfläche.
21	0	Dezimal				
22	75	Dezimal	75		Mittelpunkt X	Die X-Koordinate des Mittelpunktes der Objektoberfläche (im Koordinatensystem des Benutzers).
23	0	Dezimal				

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
24	-71	Dezimal	-71		Mittelpunkt Y	Die Y-Koordinate des Mittelpunktes der Objektoberfläche (im Koordinatensystem des Benutzers).
25	0	Dezimal				
26	783	Dezimal	783		Mittelpunkt Z	Die Z-Koordinate des Mittelpunktes der Objektoberfläche (im Koordinatensystem des Benutzers).
27	0	Dezimal				
28	78	Dezimal	078		Gierwinkel	Der Gierwinkel liegt zwischen der x-Achse (Weltkoordinatensystem) und dem Vektor entlang der "Länge" des Objektes.
29	0	Dezimal				
30	0	Dezimal	+000		Rotation X	Rotation um die X-Achse des erkannten Objektes (im Koordinatensystem des Benutzers).
31	0	Dezimal				
32	0	Dezimal	+000		Rotation Y	Rotation um die Y-Achse des erkannten Objektes (im Koordinatensystem des Benutzers).
33	0	Dezimal				
34	56	Dezimal	+056		Rotation Z	Rotation um die Z-Achse des erkannten Objektes (im Koordinatensystem des Benutzers).
35	0	Dezimal				



Das fehlerhafte Ausführen eines Kommandos führt zu folgendem Zustand:

- Error Bit = 1
- Gespiegeltes Kommandowort wird angezeigt
- Asynchrones Nachrichtenbit = 0
- Asynchrone Nachrichtenkennung = 0
- Nachrichtenzähler zählt um 1 hoch

9.3.11 Prozesswerte der Roboter Greifernavigation über PROFINET senden

Das Gerät kann die Prozesswerte über den Feldbus PROFINET an eine SPS senden.



Es kann immer nur ein Feldbus aktiv sein. Der Feldbus ist einstellbar (→ Softwarehandbuch).

Im Ausgabe-String werden die Prozesswerte durch ein Semikolon getrennt. Der Ausgabe-String wird in der angezeigten Reihenfolge an eine SPS übertragen.



Beachten Sie die folgenden Hinweise beim Übertragen des Ausgabe-Strings an eine SPS:

- Die Bytes 0 bis 7 sind Teil des Ausgabe-Strings. Sie werden nicht im ifm Vision Assistant angezeigt.
- Die Bytes 14 bis 35 wiederholen sich für jedes unter "Anzahl der Objekte" eingestellte Objekt (maximal 10 Wiederholungen).
- Im Ausgabe-String enthaltene Semikolon ";" werden nicht übertragen.
- Float-Werte werden vor dem Übertragen in 16-Bit Integer umgewandelt.
- Alle Zahlenwerte werden vor dem Übertragen in 16-Bit Integer umgewandelt.

Der Ausgabe-String setzt sich wie folgt zusammen:

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

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
0	2#0010_0000	Binär	0.5		Gespiegeltes Kommandowort	• Bit 0.5 zeigt ein erfolgreiches Trigger-Kommando an
1	2#0000_0000	Binär				
2	2#0000_0000	Binär	3		Synchrone / asynchrone Nachrichtenennung	
3	2#0000_0000	Binär				
4	2#0000_0000	Binär				
5	2#0000_0011	Binär	3		Nachrichtenzähler	• Das Gerät hat 3 Nachrichten empfangen • Zählt um 1 hoch, bei jeder Aktion (Trigger, gesendete Nachricht etc.)
6	2#0000_0000	Binär				
7	2#0000_0000	Binär			Reserviert	
8	0	Dezimal				
9	0	Dezimal	0		Fehler	Fehler: 0 = kein Fehler 1 = undefinierter Fehler 2 = kein Objekt gefunden
10	1	Dezimal	01		Anzahl der Objekte	Anzahl der gefundenen Objekte.
11	0	Dezimal				
12	8	Dezimal	08		Anzahl der Objektkandidaten	Anzahl der gefundenen und geprüften Objektkandidaten.
13	0	Dezimal				
14	1	Binär	1		Objekt gefunden	0 = kein Objekt gefunden 1 = Objekt gefunden
15	0	Binär				
16	338	Dezimal	338	mm	Breite	Die breiteste Abmessung der Objektoberfläche.
17	0	Dezimal				
18	142	Dezimal	142	mm	Höhe	Die Objekthöhe relativ zur Grundplatte.
19	0	Dezimal				
20	452	Dezimal	452	mm	Länge	Die längste Abmessung der Objektoberfläche.
21	0	Dezimal				
22	75	Dezimal	75		Mittelpunkt X	Die X-Koordinate des Mittelpunktes der Objektoberfläche (im Koordinatensystem des Benutzers).
23	0	Dezimal				

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
24	-71	Dezimal	-71		Mittelpunkt Y	Die Y-Koordinate des Mittelpunktes der Objektoberfläche (im Koordinatensystem des Benutzers).
25	0	Dezimal				
26	783	Dezimal	783		Mittelpunkt Z	Die Z-Koordinate des Mittelpunktes der Objektoberfläche (im Koordinatensystem des Benutzers).
27	0	Dezimal				
28	78	Dezimal	078		Gierwinkel	Der Gierwinkel liegt zwischen der x-Achse (Weltkoordinatensystem) und dem Vektor entlang der "Länge" des Objektes.
29	0	Dezimal				
30	0	Dezimal	+000		Rotation X	Rotation um die X-Achse des erkannten Objektes (im Koordinatensystem des Benutzers).
31	0	Dezimal				
32	0	Dezimal	+000		Rotation Y	Rotation um die Y-Achse des erkannten Objektes (im Koordinatensystem des Benutzers).
33	0	Dezimal				
34	56	Dezimal	+056		Rotation Z	Rotation um die Z-Achse des erkannten Objektes (im Koordinatensystem des Benutzers).
35	0	Dezimal				



Das fehlerhafte Ausführen eines Kommandos führt zu folgendem Zustand:

- Error Bit = 1
- Gespiegeltes Kommandowort wird angezeigt
- Asynchrones Nachrichtenbit = 0
- Asynchrone Nachrichtenkennung = 0
- Nachrichtenzähler zählt um 1 hoch

9.3.12 Prozesswerte der Roboter Greifernavigation über TCP/IP senden

Das Gerät kann die Prozesswerte über das Protokoll TCP/IP an eine SPS senden. Die Prozesswerte werden im ifm Vision Assistant als Ausgabe-String wie folgt dargestellt:

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

Im Ausgabe-String werden die Prozesswerte durch ein Semikolon getrennt. Der Ausgabe-String wird in der angezeigten Reihenfolge an eine SPS übertragen.



Beachten Sie die folgenden Hinweise beim Übertragen des Ausgabe-Strings an eine SPS:

- Im Ausgabe-String enthaltene Semikolon ";" werden nicht übertragen.
- Die Prozesswerte "Objekt gefunden" bis "Rotation Z" wiederholen sich für jedes unter "Anzahl der Objekte" eingestellte Objekt (maximal 10 Wiederholungen).
- Alle Zahlenwerte werden vor dem Übertragen in 16-Bit Integer umgewandelt.

Der Ausgabe-String setzt sich wie folgt zusammen (Datentyp: ASCII):

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

Prozesswert	Einheit	Beschreibung
star		Start-String
0		Fehler
01		Anzahl der Objekte
08		Anzahl der Objektkandidaten
1		0 = kein Objekt gefunden 1 = Objekt gefunden
0.338	mm	Breite
0.142	mm	Höhe
0.452	mm	Länge
+0.075		Mittelpunkt X
-0.071		Mittelpunkt Y
+0.783		Mittelpunkt Z
078		Gierwinkel
+000		Rotation X
+000		Rotation Y
+056		Rotation Z
stop		Stop-String

9.3.13 Prozesswerte der Depalettierung über EtherNet/IP senden

Das Gerät kann die Prozesswerte über den Feldbus EtherNet/IP an eine SPS senden.



Es kann immer nur ein Feldbus aktiv sein. Der Feldbus ist einstellbar (→ Softwarehandbuch).

Im Ausgabe-String werden die Prozesswerte durch ein Semikolon getrennt. Der Ausgabe-String wird in der angezeigten Reihenfolge an eine SPS übertragen.



Beachten Sie die folgenden Hinweise beim Übertragen des Ausgabe-Strings an eine SPS:

- Die Bytes 0 bis 7 sind Teil des Ausgabe-Strings. Sie werden nicht im ifm Vision Assistant angezeigt.
- Im Ausgabe-String enthaltene Semikolon ";" werden nicht übertragen.
- Float-Werte werden vor dem Übertragen in 16-Bit Integer umgewandelt.
- Alle Zahlenwerte werden vor dem Übertragen in 16-Bit Integer umgewandelt.

Der Ausgabe-String setzt sich wie folgt zusammen:

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

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
0	2#0010_0000	Binär	0.5		Gespiegeltes Kommandowort	• Bit 0.5 zeigt ein erfolgreiches Trigger-Kommando an
1	2#0000_0000	Binär				
2	2#0000_0000	Binär	3		Synchrone / asynchrone Nachrichtenennung	• Das Gerät hat 3 Nachrichten empfangen • Zählt um 1 hoch, bei jeder Aktion (Trigger, gesendete Nachricht etc.)
3	2#0000_0000	Binär				
4	2#0000_0000	Binär			Nachrichtenzähler	
5	2#0000_0011	Binär				
6	2#0000_0000	Binär			Reserviert	
7	2#0000_0000	Binär				
8	1	Binär	1		Objekt gefunden	0 = kein Objekt gefunden 1 = Objekt gefunden
9	0	Binär				
10	200	Dezimal	200	mm	Breite	Die breiteste Abmessung der Objektoberfläche.
11	0	Dezimal				
12	150	Dezimal	150	mm	Höhe	Die Objekthöhe relativ zur Grundplatte.
13	0	Dezimal				
14	307	Dezimal	307	mm	Länge	Die längste Abmessung der Objektoberfläche.
15	0	Dezimal				
16	2	Dezimal	+2		Mittelpunkt X	Die X-Koordinate des Mittelpunktes der Objektoberfläche (im Koordinatensystem des Benutzers).
17	0	Dezimal				
18	10044	Dezimal	-10044		Mittelpunkt Y	Die Y-Koordinate des Mittelpunktes der Objektoberfläche (im Koordinatensystem des Benutzers).
19	0	Dezimal				
20	3100	Dezimal	+3100		Mittelpunkt Z	Die Z-Koordinate des Mittelpunktes der Objektoberfläche (im Koordinatensystem des Benutzers).
21	0	Dezimal				
22	170	Dezimal	+170		Rotation X	Rotation um die X-Achse des erkannten Objektes (im Koordinatensystem des Benutzers).
23	0	Dezimal				

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
24	-133	Dezimal	-133		Rotation Y	Rotation um die Y-Achse des erkannten Objektes (im Koordinatensystem des Benutzers).
25	0	Dezimal				
26	-132	Dezimal	-132		Rotation Z	Rotation um die Z-Achse des erkannten Objektes (im Koordinatensystem des Benutzers).
27	0	Dezimal				
28	02	Dezimal	02		Aktuelle Ebene	Aktuelle Palettenebene, beginnend mit "0". Eine leere Ebene wird durch "0" gekennzeichnet.
29	0	Dezimal				
30	1	Binär	1		Trennblatt	Ein Trennblatt befindet sich auf einer Palettenebene: 0 = kein Trennblatt erkannt 1 = Trennblatt erkannt
31	0	Binär				
32	098	Dezimal	098		Fehler	Fehler: 0 = kein Fehler 1 = undefinierter Fehler 2 = unerwartetes Objekt erkannt Weitere Fehlercodes: (→ 13.1.5).
33	0	Dezimal				
34	00	Binär	00		Kollisionsfrei	Kollisionsfreies Depalettieren: 0: nein 1: ja
35	0	Binär				
36	1	Dezimal	1		Qualität	Qualität der Objekterkennung zwischen 0 und 100. Der Wert "100" steht für bestmögliche Qualität.
37	0	Dezimal				



Das fehlerhafte Ausführen eines Kommandos führt zu folgendem Zustand:

- Error Bit = 1
- Gespiegeltes Kommandowort wird angezeigt
- Asynchrones Nachrichtenbit = 0
- Asynchrone Nachrichtenkennung = 0
- Nachrichtenzähler zählt um 1 hoch

9.3.14 Prozesswerte der Depalettierung über PROFINET senden

Das Gerät kann die Prozesswerte über den Feldbus PROFINET an eine SPS senden.



Es kann immer nur ein Feldbus aktiv sein. Der Feldbus ist einstellbar (→ Softwarehandbuch).

Im Ausgabe-String werden die Prozesswerte durch ein Semikolon getrennt. Der Ausgabe-String wird in der angezeigten Reihenfolge an eine SPS übertragen.



Beachten Sie die folgenden Hinweise beim Übertragen des Ausgabe-Strings an eine SPS:

- Die Bytes 0 bis 7 sind Teil des Ausgabe-Strings. Sie werden nicht im ifm Vision Assistant angezeigt.
- Im Ausgabe-String enthaltene Semikolon ";" werden nicht übertragen.
- Float-Werte werden vor dem Übertragen in 16-Bit Integer umgewandelt.
- Alle Zahlenwerte werden vor dem Übertragen in 16-Bit Integer umgewandelt.

Der Ausgabe-String setzt sich wie folgt zusammen:

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

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
0	2#0010_0000	Binär	0.5		Gespiegeltes Kommandowort	• Bit 0.5 zeigt ein erfolgreiches Trigger-Kommando an
1	2#0000_0000	Binär				
2	2#0000_0000	Binär			Synchrone / asynchrone Nachrichtenennung	
3	2#0000_0000	Binär				
4	2#0000_0000	Binär	3		Nachrichtenzähler	• Das Gerät hat 3 Nachrichten empfangen • Zählt um 1 hoch, bei jeder Aktion (Trigger, gesendete Nachricht etc.)
5	2#0000_0011	Binär				
6	2#0000_0000	Binär			Reserviert	
7	2#0000_0000	Binär				
8	1	Binär	1		Objekt gefunden	0 = kein Objekt gefunden 1 = Objekt gefunden
9	0	Binär				
10	200	Dezimal	200	mm	Breite	Die breiteste Abmessung der Objektoberfläche.
11	0	Dezimal				
12	150	Dezimal	150	mm	Höhe	Die Objekthöhe relativ zur Grundplatte.
13	0	Dezimal				
14	307	Dezimal	307	mm	Länge	Die längste Abmessung der Objektoberfläche.
15	0	Dezimal				
16	2	Dezimal	+2		Mittelpunkt X	Die X-Koordinate des Mittelpunktes der Objektoberfläche (im Koordinatensystem des Benutzers).
17	0	Dezimal				
18	10044	Dezimal	-10044		Mittelpunkt Y	Die Y-Koordinate des Mittelpunktes der Objektoberfläche (im Koordinatensystem des Benutzers).
19	0	Dezimal				
20	3100	Dezimal	+3100		Mittelpunkt Z	Die Z-Koordinate des Mittelpunktes der Objektoberfläche (im Koordinatensystem des Benutzers).
21	0	Dezimal				
22	170	Dezimal	+170		Rotation X	Rotation um die X-Achse des erkannten Objektes (im Koordinatensystem des Benutzers).
23	0	Dezimal				

Byte-Nr.	Daten	Kodierung	Prozesswert	Einheit	Beschreibung	Kommentar
24	-133	Dezimal	-133		Rotation Y	Rotation um die Y-Achse des erkannten Objektes (im Koordinatensystem des Benutzers).
25	0	Dezimal				
26	-132	Dezimal	-132		Rotation Z	Rotation um die Z-Achse des erkannten Objektes (im Koordinatensystem des Benutzers).
27	0	Dezimal				
28	02	Dezimal	02		Aktuelle Ebene	Aktuelle Palettenebene, beginnend mit "0". Eine leere Ebene wird durch "0" gekennzeichnet.
29	0	Dezimal				
30	1	Binär	1		Trennblatt	Ein Trennblatt befindet sich auf einer Palettenebene: 0 = kein Trennblatt erkannt 1 = Trennblatt erkannt
31	0	Binär				
32	098	Dezimal	098		Fehler	Fehler: 0 = kein Fehler 1 = undefinierter Fehler 2 = unerwartetes Objekt erkannt Weitere Fehlercodes: (→ 13.1.5).
33	0	Dezimal				
34	00	Binär	00		Kollisionsfrei	Kollisionsfreies Depalettieren: 0: nein 1: ja
35	0	Binär				
36	1	Dezimal	1		Qualität	Qualität der Objekterkennung zwischen 0 und 100. Der Wert "100" steht für bestmögliche Qualität.
37	0	Dezimal				



Das fehlerhafte Ausführen eines Kommandos führt zu folgendem Zustand:

- Error Bit = 1
- Gespiegeltes Kommandowort wird angezeigt
- Asynchrones Nachrichtenbit = 0
- Asynchrone Nachrichtenkennung = 0
- Nachrichtenzähler zählt um 1 hoch

9.3.15 Prozesswerte der Depalettierung über TCP/IP senden

Das Gerät kann die Prozesswerte über das Protokoll TCP/IP an eine SPS senden. Die Prozesswerte werden im ifm Vision Assistant als Ausgabe-String wie folgt dargestellt:

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

Im Ausgabe-String werden die Prozesswerte durch ein Semikolon getrennt. Der Ausgabe-String wird in der angezeigten Reihenfolge an eine SPS übertragen.



Beachten Sie die folgenden Hinweise beim Übertragen des Ausgabe-Strings an eine SPS:

- Im Ausgabe-String enthaltene Semikolon ";" werden nicht übertragen.
- Alle Zahlenwerte werden vor dem Übertragen in 16-Bit Integer umgewandelt.

Der Ausgabe-String setzt sich wie folgt zusammen (Datentyp: ASCII):

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

Prozesswert	Einheit	Beschreibung
star		Start-String
1		0 = kein Objekt gefunden 1 = Objekt gefunden
0.200		Breite
0.150		Höhe
0.307		Länge
+00.002		Mittelpunkt X
-10.044		Mittelpunkt Y
+03.100		Mittelpunkt Z
+170		Rotation X
-133		Rotation Y
-132		Rotation Z
02		Aktuelle Ebene
1		0 = kein Trennblatt erkannt 1 = Trennblatt erkannt
098		Fehler
00		0 = kein kollisionsfreies Depalettieren 1 = kollisionsfreies Depalettieren
1		Qualität der Objekterkennung (0 bis 100)
stop		Stop-String

10. Wartung, Instandsetzung und Entsorgung

Beachten Sie die folgenden Anweisungen:

- ▶ Das Gerät nicht öffnen. Innerhalb des Gerätes befinden sich keine vom Anwender zu wartenden Bauteile. Die Instandsetzung des Gerätes darf nur durch den Hersteller erfolgen.
- ▶ Das Gerät gemäß den nationalen Umweltvorschriften entsorgen.

10.1 Reinigung

Beachten Sie die folgenden Anweisungen vor dem Reinigen des Gerätes:

- ▶ Sauberes fusselfreies Tuch verwenden.
- ▶ Glasreiniger als Reinigungsmittel verwenden.



Werden die Anweisungen nicht eingehalten, können Messfehler durch Kratzer auf der Frontscheibe auftreten.

10.2 Firmware aktualisieren

Mit der Software ifm Vision Assistant kann die Firmware des Gerätes aktualisiert werden.



Im Gerät gespeicherte Parameter gehen durch das Aktualisieren der Firmware verloren. Erstellen Sie vor dem Aktualisieren der Firmware eine Sicherungskopie der Parameter:

- ▶ Vor dem Aktualisieren der Firmware Parameter exportieren.
- ▶ Nach dem Aktualisieren der Firmware Parameter importieren.



Firmware-Updates sind im Internet verfügbar: www.ifm.com

10.3 Gerät austauschen

Beim Austausch eines Gerätes gehen die Parameter verloren. Erstellen Sie vor dem Austausch des Gerätes eine Sicherungskopie der Parameter:

- ▶ Vor dem Austausch die Parameter des alten Gerätes exportieren.
- ▶ Nach dem Austausch die Parameter auf das neue Gerät importieren.



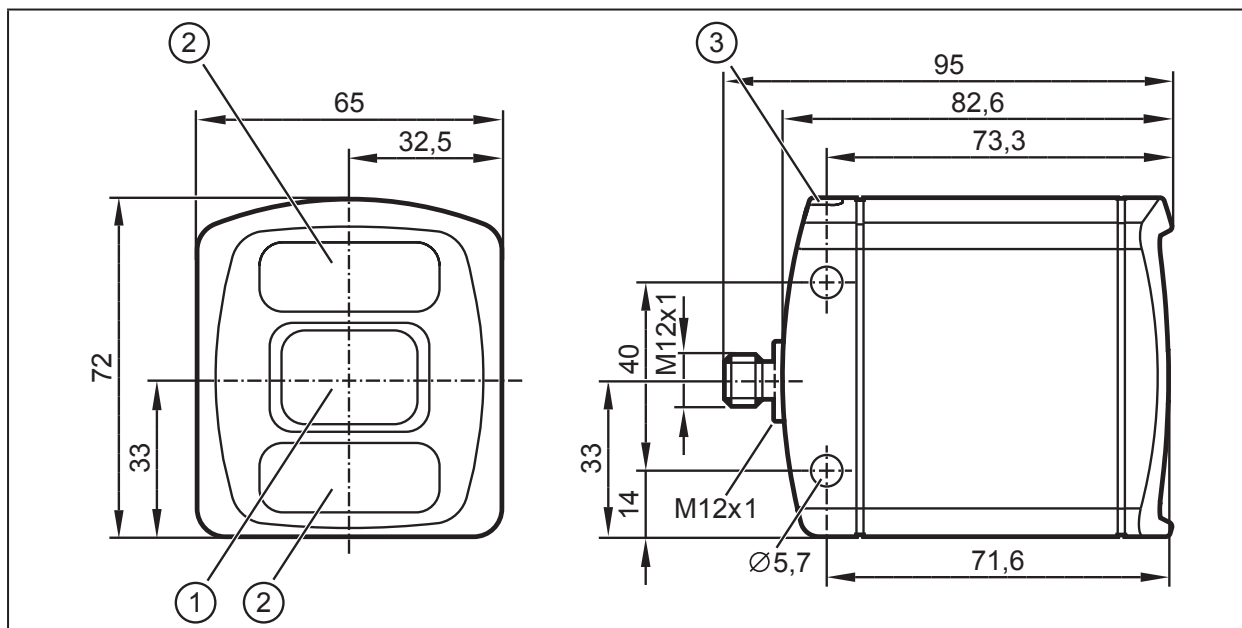
Mit den Ex- und Importieren von Parametern können mehrere Geräte schnell mit den selben Parametern ausgestattet werden.

11. Zulassungen/Normen

Die EU-Konformitätserklärung ist abrufbar unter: www.ifm.com

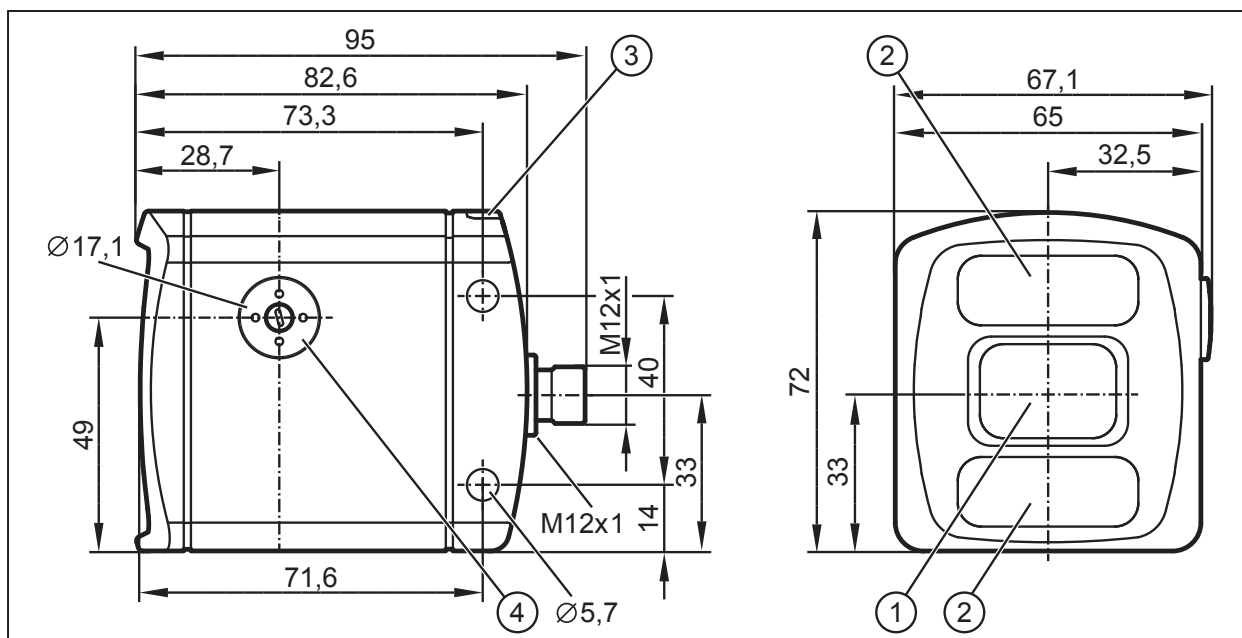
12. Maßzeichnungen

12.1 O3D302 / O3D312



- ① Objektiv
- ② Beleuchtungseinheit
- ③ LED 2-farbig (gelb/grün)

12.2 O3D300 / O3D310



- ① Objektiv
- ② Beleuchtungseinheit
- ③ LED 2-farbig (gelb/grün)
- ④ Fokuseinsteller

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\n00100005000000:{"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 acquisition. 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_HEIGHT	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 formatted like this: <pre>{ "AcquisitionDuration": 20.391, "EvaluationDuration": 37.728, "FrameDuration": 37.728, "FrameRate": 15.202, "TemperatureIllu": 52.9 }</pre> 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: $R = R_x \cdot R_y \cdot R_z$ The transformation of a point P is specified by $P_t = R \cdot P + [trans_x, trans_y, trans_z]$. The device extrinsic 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-dependent 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, consisting 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 length width height qualityLength qualityWidth qualityHeight xMidTop yMidTop zMidTop yawAngle backgroundPlaneDistance	These results are available for a dimensioning application. If the model is not of the type dimensioning, the IDs shall output a null-value.	int8 float float float float float float float float float float float
numGood numUnderSP1 numOverSP2 numInvalid allROIsGood anchorFound hasAnchorTracking	These results are available for a completeness, level and distance applications. If the model is not of one of these types, the IDs shall output a null-value.	int int int int bool bool 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_OVERFLOW = 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"] { x y }	Coordinates of the available UFC markers.	float32 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 [°].

ID	Description	Native data type																																																						
error	Errors in the algorithm:	uint32																																																						
	<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_BackgroundTeachingStatus_ErrorNotEnoughValidPixels</td><td>Background estimation only: not enough valid pixels.</td></tr><tr><td>8</td><td>Depalletizing_Error_BackgroundTeachingStatus_ErrorStdTooHigh</td><td>Background estimation only: standard deviation too high.</td></tr><tr><td>9</td><td>Depalletizing_Error_BackgroundTeachingStatus_ErrorPlaneFitFailed</td><td>Background estimation only: estimation of the plane failed.</td></tr><tr><td>10</td><td>Depalletizing_Error_BackgroundTeachingStatus_ErrorPlaneAngleTooHigh</td><td>Background estimation only: plane angle too high.</td></tr><tr><td>11</td><td>Depalletizing_Error_BackgroundTeachingStatus_ErrorRotationCalculationFailed</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_InsufficientMarginToImageBorder</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>		Value	Name	Description	0	Depalletizing_Error_None	No error detected.	1	Depalletizing_Error_Unknown	Unknown error detected.	2	Depalletizing_Error_UnexpectedObject	Unexpected object detected.	3	Depalletizing_Error_StackEmpty	Stack is empty.	4	Depalletizing_Error_NoObjectSizes	No box dimensions provided on the input.	5	Depalletizing_Error_NoObjectMatch	No matching object found.	6	Depalletizing_Error_DataInvalid	Too many pixels are invalid.	7	Depalletizing_Error_BackgroundTeachingStatus_ErrorNotEnoughValidPixels	Background estimation only: not enough valid pixels.	8	Depalletizing_Error_BackgroundTeachingStatus_ErrorStdTooHigh	Background estimation only: standard deviation too high.	9	Depalletizing_Error_BackgroundTeachingStatus_ErrorPlaneFitFailed	Background estimation only: estimation of the plane failed.	10	Depalletizing_Error_BackgroundTeachingStatus_ErrorPlaneAngleTooHigh	Background estimation only: plane angle too high.	11	Depalletizing_Error_BackgroundTeachingStatus_ErrorRotationCalculationFailed	Background estimation only: internal numerical error in calculation of rotation.	12	Depalletizing_Error_InvalidReferenceTeach	Invalid background teach.	13	Depalletizing_Error_InvalidVOITeach	Invalid VOI teach.	14	Depalletizing_Error_InsufficientMarginToImageBorder	Not enough space between segmented layer and image border.	15	Depalletizing_Error_IncorrectObjectSizes	Provided box dimensions are invalid.	16	Depalletizing_Error_Underfill	Measurements are below the background level.
	Value		Name	Description																																																				
	0		Depalletizing_Error_None	No error detected.																																																				
	1		Depalletizing_Error_Unknown	Unknown error detected.																																																				
	2		Depalletizing_Error_UnexpectedObject	Unexpected object detected.																																																				
	3		Depalletizing_Error_StackEmpty	Stack is empty.																																																				
	4		Depalletizing_Error_NoObjectSizes	No box dimensions provided on the input.																																																				
	5		Depalletizing_Error_NoObjectMatch	No matching object found.																																																				
	6		Depalletizing_Error_DataInvalid	Too many pixels are invalid.																																																				
	7		Depalletizing_Error_BackgroundTeachingStatus_ErrorNotEnoughValidPixels	Background estimation only: not enough valid pixels.																																																				
	8		Depalletizing_Error_BackgroundTeachingStatus_ErrorStdTooHigh	Background estimation only: standard deviation too high.																																																				
	9		Depalletizing_Error_BackgroundTeachingStatus_ErrorPlaneFitFailed	Background estimation only: estimation of the plane failed.																																																				
	10		Depalletizing_Error_BackgroundTeachingStatus_ErrorPlaneAngleTooHigh	Background estimation only: plane angle too high.																																																				
	11		Depalletizing_Error_BackgroundTeachingStatus_ErrorRotationCalculationFailed	Background estimation only: internal numerical error in calculation of rotation.																																																				
	12		Depalletizing_Error_InvalidReferenceTeach	Invalid background teach.																																																				
	13		Depalletizing_Error_InvalidVOITeach	Invalid VOI teach.																																																				
14	Depalletizing_Error_InsufficientMarginToImageBorder	Not enough space between segmented layer and image border.																																																						
15	Depalletizing_Error_IncorrectObjectSizes	Provided box dimensions are invalid.																																																						
16	Depalletizing_Error_Underfill	Measurements are below the background level.																																																						
objectFound	Object can be successfully measured (0 if false, 1 if true).	uint32																																																						
objectQuality	Quality of the object detection between 0 and 100.	float32																																																						
objectLength	Object length is the longest dimension of the top surface of the object.	float32																																																						
objectWidth	Object width is the shortest dimension of the top surface of the object.	float32																																																						
objectHeight	Object height is the object height relative to the ground plane.	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																																																						

DE

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: height 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.

DE

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	

DE

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	

DE

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)

DE

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 occurred 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 asynchronous 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 reset 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

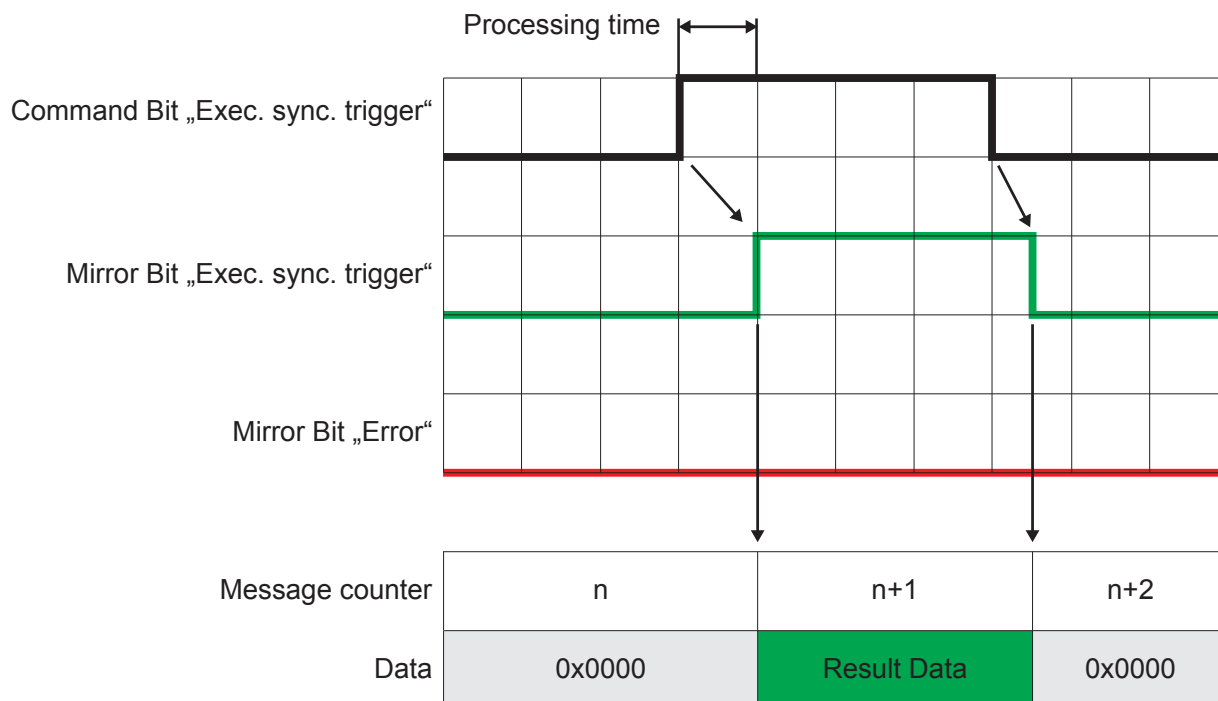
The following command executes an extended command. The ID of the extended command is stored as 16 bit integer in bytes 2-3. The remaining data depends on the extended command.

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.

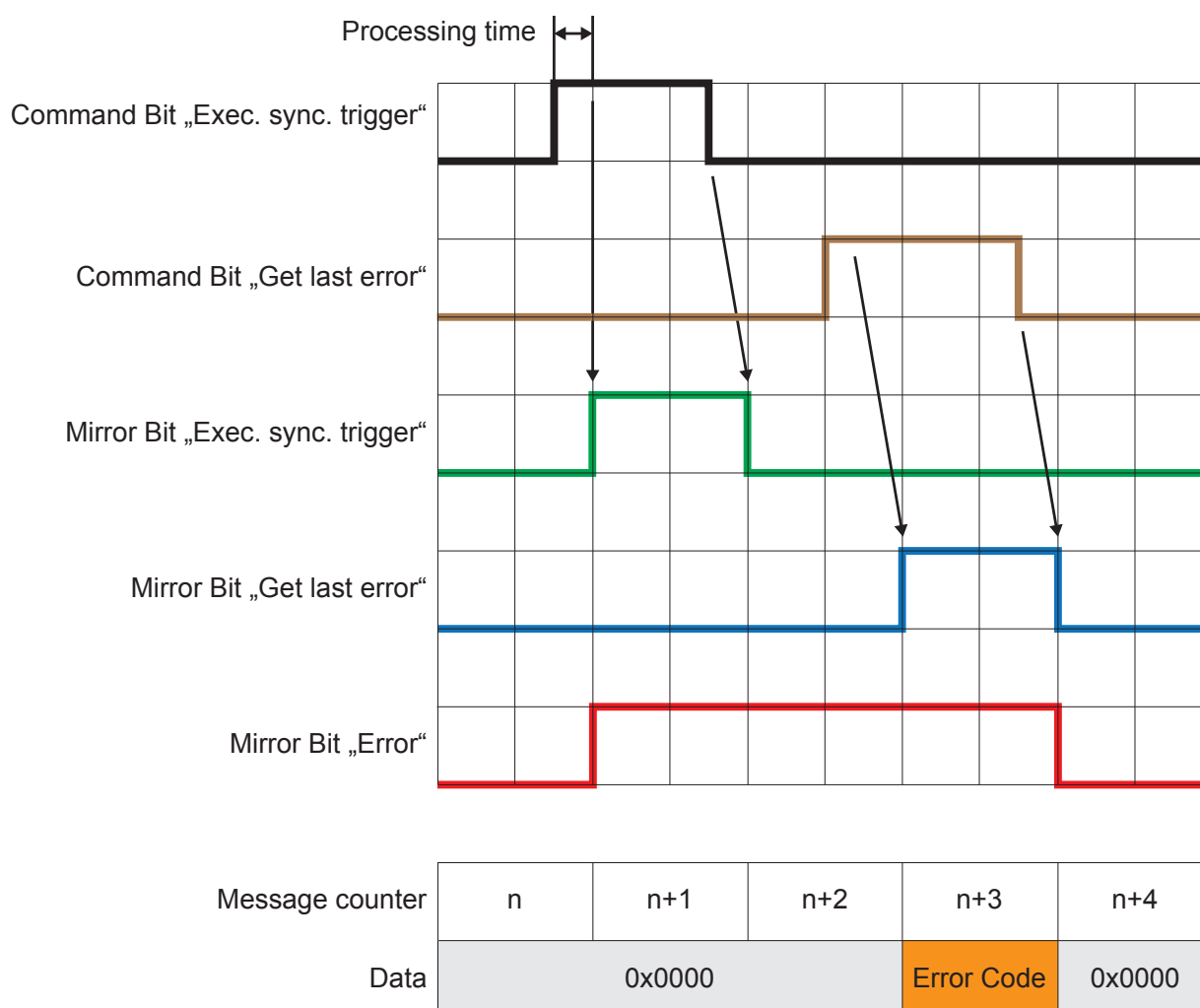
Use of extended command with the depalletising application

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 occurred 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 asynchronous 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 reset 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.

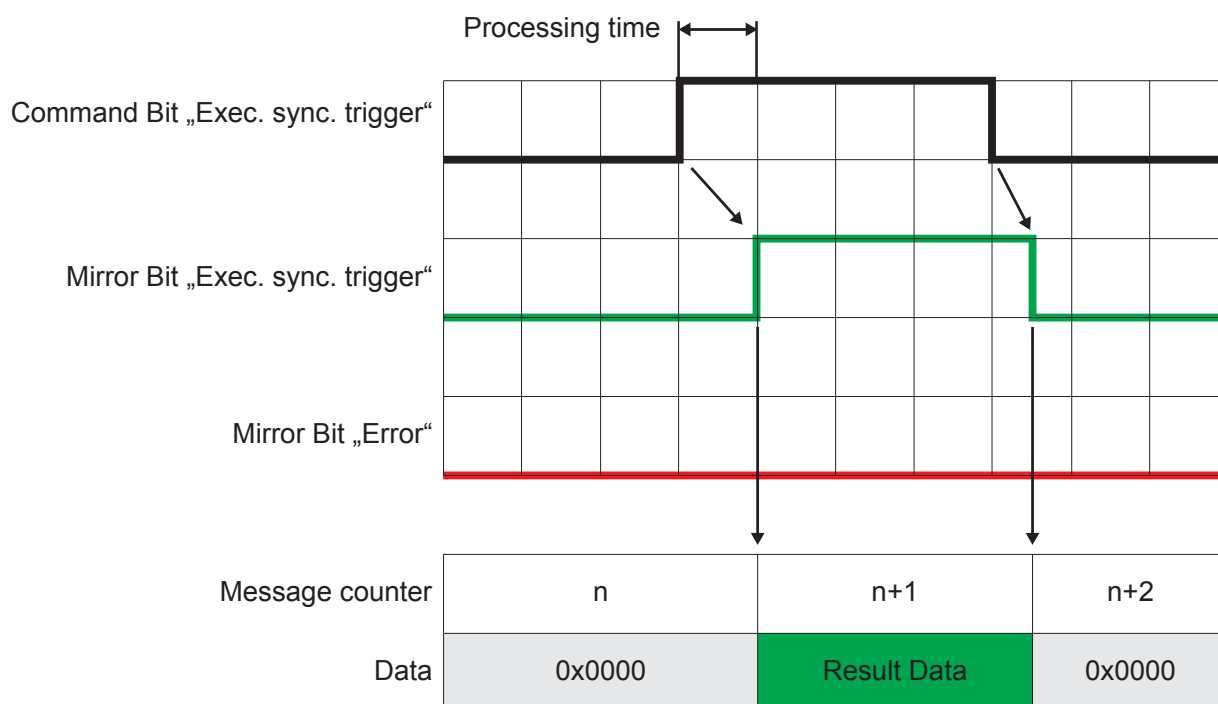
ID	Description
1	Set temporary application parameter The ID of the parameter to be changed is stored as unsigned 16 bit integer in bytes 4-5. The value of the parameter is stored as signed 16 bit integer in bytes 6-7.

DE

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

