



Notice d'utilisation  
Capteur 3D

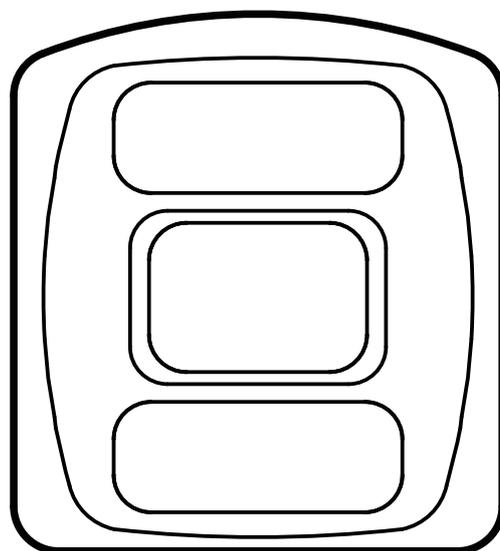
**O3D300**

**O3D302**

**O3D310**

**O3D312**

**FR**



## Contenu

|   |    |
|---|----|
| 1. Remarques préliminaires . . . . .  | 4  |
| 1.1 Symboles utilisés . . . . .   | 4  |
| 1.2 Avertissements utilisés . . . . .   | 4  |
| 2. Consignes de sécurité . . . . .  | 4  |
| 2.1 Remarques générales . . . . .   | 4  |
| 2.2 Cible 4 . . . . .   |    |
| 2.3 Raccordement électrique . . . . .   | 4  |
| 2.4 Interventions sur l'appareil . . . . .  | 4  |
| 3. Fonctionnement et caractéristiques . . . . .   | 5  |
| 4. Fourniture . . . . .   | 5  |
| 5. Accessoires . . . . .  | 5  |
| 6. Montage . . . . .  | 6  |
| 6.1 Sélectionner le lieu de montage . . . . .   | 6  |
| 6.2 Préparer l'appareil pour la mise en service. . . . .                                    | 7  |
| 6.2.1 Limites d'avertissement typiques pour O3D300 / O3D302. . . . .                        | 7  |
| 6.2.2 Limites d'avertissement typiques pour O3D310 / O3D312. . . . .                        | 8  |
| 6.2.3 Réduire la température de surface . . . . .   | 8  |
| 6.3 Installer l'appareil . . . . .  | 9  |
| 6.4 Accessoires de montage . . . . .  | 9  |
| 7. Raccordement électrique . . . . .  | 10 |
| 7.1 Schéma de branchement . . . . .   | 10 |
| 7.1.1 Broches 1 / 3 (24 V / GND). . . . .   | 11 |
| 7.1.2 Broche 2 (entrée trigger). . . . .  | 11 |
| 7.1.3 Broches 4 / 5 / 6 (sorties de commutation). . . . .                                   | 11 |
| 7.1.4 Broche 4 (sortie analogique). . . . .   | 12 |
| 7.1.5 Broches 7 / 8 (entrées de commutation). . . . .                                       | 12 |
| 7.2 Exemples de câblage . . . . .   | 13 |
| 7.2.1 Déclencher la capture d'images avec un détecteur de proximité . . . . .               | 13 |
| 7.2.2 Monter plusieurs appareils l'un à côté de l'autre. . . . .                            | 14 |
| 7.3 Sélection statique de l'application . . . . .   | 15 |
| 7.4 Sélection de l'application activée par impulsion . . . . .                              | 16 |
| 8. Eléments de visualisation . . . . .  | 17 |
| 9. Mise en service . . . . .  | 18 |
| 9.1 Paramétrer l'appareil. . . . .  | 18 |
| 9.2 Détecter un objet . . . . .   | 18 |
| 9.3 Transmettre les valeurs process. . . . .  | 19 |
| 9.3.1 Transmettre les valeurs process du contrôle de complétude via EtherNet/IP . . . . .   | 19 |
| 9.3.2 Transmettre les valeurs process du contrôle de complétude via PROFINET. . . . .       | 21 |
| 9.3.3 Transmettre les valeurs process du contrôle de complétude via TCP/IP . . . . .        | 23 |
| 9.3.4 Transmettre les valeurs process du dimensionnement d'objets via EtherNet/IP . . . . . | 24 |
| 9.3.5 Transmettre les valeurs process du dimensionnement d'objets via PROFINET . . . . .    | 26 |
| 9.3.6 Transmettre les valeurs process du dimensionnement d'objets via TCP/IP. . . . .       | 28 |
| 9.3.7 Transmettre les valeurs process de la mesure de niveau via EtherNet/IP . . . . .      | 29 |
| 9.3.8 Transmettre les valeurs process de la mesure de niveau via PROFINET . . . . .         | 31 |

|         |   |    |
|---------|---|----|
| 9.3.9   | Transmettre les valeurs process de la mesure de niveau via TCP/IP ..... | 32 |
| 10.     | Maintenance, réparation et élimination .....                            | 33 |
| 10.1    | Nettoyage .....   | 33 |
| 10.2    | Mettre à jour le firmware .....   | 33 |
| 10.3    | Remplacer l'appareil .....  | 33 |
| 11.     | Homologations/normes .....  | 33 |
| 12.     | Schémas d'encombrement .....  | 34 |
| 12.1    | O3D302 / O3D312 .....   | 34 |
| 12.2    | O3D300 / O3D310 .....   | 34 |
| 13.     | Appendix 35 .....   |    |
| 13.1    | Process Interface .....   | 35 |
| 13.1.1  | Sending Commands .....  | 35 |
| 13.1.2  | Receiving Images .....  | 37 |
| 13.1.3  | Image data .....  | 37 |
| 13.1.4  | Additional Information for CONFIDENCE_IMAGE .....                       | 40 |
| 13.1.5  | Configuration of PCIC Output .....                                      | 41 |
| 13.2    | Process Interface Command Reference .....                               | 47 |
| 13.2.1  | t Command (Asynchronous Trigger) .....                                  | 47 |
| 13.2.2  | T? Command (Synchronous Trigger) .....                                  | 47 |
| 13.2.3  | l? Command .....  | 48 |
| 13.2.4  | p Command .....   | 48 |
| 13.2.5  | a Command .....   | 49 |
| 13.2.6  | A? Command .....  | 49 |
| 13.2.7  | v Command .....   | 50 |
| 13.2.8  | V? Command .....  | 50 |
| 13.2.9  | c Command .....   | 50 |
| 13.2.10 | C? Command .....  | 51 |
| 13.2.11 | S? Command .....  | 51 |
| 13.2.12 | G? Command .....  | 52 |
| 13.2.13 | H? Command .....  | 53 |
| 13.2.14 | o Command .....   | 53 |
| 13.2.15 | O? Command .....  | 54 |
| 13.2.16 | E? Command .....  | 54 |
| 13.3    | Error codes .....   | 55 |
| 13.4    | EtherNet/IP .....   | 56 |
| 13.4.1  | Data structures for consuming and producing assemblies .....            | 56 |
| 13.4.2  | Functionality of the Ethernet/IP application .....                      | 57 |
| 13.4.3  | Signal sequence with synchronous trigger .....                          | 61 |
| 13.4.4  | Signal sequence with failed trigger .....                               | 61 |
| 13.5    | PROFINET IO .....   | 62 |
| 13.5.1  | Data structures for output and input frame .....                        | 62 |
| 13.5.2  | Functionality of PROFINET IO application .....                          | 62 |
| 13.5.3  | Signal sequence with synchronous trigger .....                          | 67 |
| 13.5.4  | Signal sequence with failed trigger .....                               | 67 |

#### Licences et marques

Microsoft®, Windows®, Windows Vista®, Windows 7®, Windows 8®, Windows 8.1® et Windows 10® sont des marques enregistrées de Microsoft Corporation.

Adobe® et Acrobat® sont des marques enregistrées d'Adobe Systems Inc.

Toutes les marques et raisons sociales utilisées sont soumises au copyright des sociétés respectives.

Cet appareil contient du logiciel Open Source (modifié si nécessaire) qui est sujet à des termes de licence spécifiques.

Remarques concernant le droit d'auteur et les termes de licence sur :

[www.ifm.com/int/GNU](http://www.ifm.com/int/GNU)

Pour du logiciel sujet à la licence publique générale GNU ou la licence publique générale limitée GNU, le texte source peut être demandé contre paiement des frais de copie et d'envoi.

## 1. Remarques préliminaires

Ce document s'adresse à des personnes compétentes. Ce sont des personnes qui sont capables – grâce à leur formation et expérience – d'envisager les risques et d'éviter des dangers potentiels qui pourraient être causés par le fonctionnement ou la maintenance de l'appareil. Le document fournit des informations sur l'utilisation correcte de l'appareil.

Lire ce document avant l'utilisation afin de vous familiariser avec les conditions d'utilisation, l'installation et le fonctionnement. Garder ce document pendant tout le temps d'utilisation de l'appareil.

### 1.1 Symboles utilisés

- ▶ Action à faire
- > Retour d'information, résultat
- [...] Désignation d'une touche, d'un bouton ou d'un affichage
- Référence croisée
-  Remarque importante  
Le non-respect peut aboutir à des dysfonctionnements ou perturbations.
-  Information  
Remarque supplémentaire

### 1.2 Avertissements utilisés

|  |
|--|
| <b>ATTENTION</b><br>Avertissement de dommages matériels. |
|--|

## 2. Consignes de sécurité

### 2.1 Remarques générales

Cette notice fait partie de l'appareil. Elle fournit des textes et des figures pour l'utilisation correcte de l'appareil et doit être lue avant installation ou emploi.

Respecter les indications de cette notice. Le non-respect de ces consignes, une utilisation en dehors des conditions définies ci-dessous, une mauvaise installation ou utilisation peuvent avoir des conséquences graves pour la sécurité des personnes et des installations.

### 2.2 Cible

Cette notice s'adresse à des personnes considérées comme compétentes selon les directives CEM et basse tension. L'appareil doit être monté, raccordé et mis en service par un électricien habilité.

### 2.3 Raccordement électrique

Mettre l'appareil hors tension en prenant des mesures externes avant toutes manipulations.

Les bornes de raccordement ne doivent être alimentées que par les signaux indiqués dans les données techniques et sur l'étiquette de l'appareil et seuls les accessoires homologués d'ifm doivent être raccordés.

### 2.4 Interventions sur l'appareil

En cas de mauvais fonctionnement de l'appareil ou en cas de doute prendre contact avec le fabricant. Des interventions sur l'appareil peuvent avoir des conséquences graves pour la sécurité des personnes et des installations. Elles ne sont pas autorisées et aboutissent à une exclusion de responsabilité et de garantie.

### 3. Fonctionnement et caractéristiques

Le capteur 3D O3D3xx est un capteur optoélectronique qui mesure, selon le principe du temps de vol, point par point la distance entre le capteur et la surface la plus proche. Le capteur 3D O3D3xx illumine la scène avec une source lumineuse infrarouge interne et calcule la distance à l'aide de la lumière réfléchie par la surface.

A l'aide d'un traitement interne des données d'images, des valeurs process sont générées à partir des données et comparées avec des valeurs de seuils. Les valeurs de comparaison et de process sont liées aux sorties TOR. Ainsi, les applications suivantes peuvent être résolues :

- Contrôle de complétude
- Mesure de niveau
- Contrôle de distance
- Dimensionnement d'objets rectangulaires
- Tri d'objets rectangulaires

Les données mesurées et les données process peuvent être fournies via Ethernet et évaluées par l'utilisateur. Le capteur 3D O3D3xx est paramétré également via Ethernet.

Le capteur 3D O3D3xx ne doit être utilisé que sous les conditions environnementales indiquées dans la fiche technique.

La sécurité de l'appareil est conçue pour l'emploi selon les conditions environnementales suivantes :

- Utilisation à l'intérieur
- Altitudes jusqu'à 2000 m
- Humidité d'air relative jusqu'à 90 % au maximum, sans condensation
- Degré de salissure 3

En raison des exigences pour les émissions de parasites électromagnétiques, l'appareil est destiné à l'utilisation dans les environnements industriels. L'appareil n'est pas approprié pour l'utilisation dans les lieux de résidence.



L'appareil ne doit être utilisé que dans les conditions environnementales indiquées dans la fiche technique.

### 4. Fourniture

- Capteur 3D O3D3xx
- Notice succincte



La fiche technique et d'autre documentation (manuel du logiciel etc.) sont disponibles sur : [www.ifm.com](http://www.ifm.com)

### 5. Accessoires

Pour le fonctionnement de l'appareil les accessoires suivants sont nécessaires :

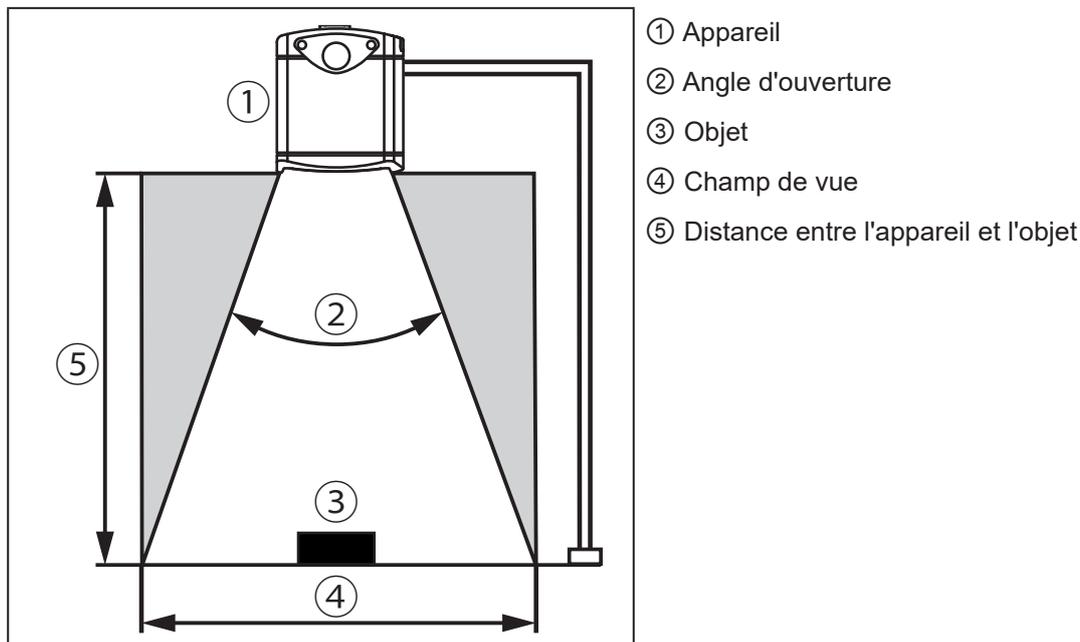
| Référence | Description  |
|-----------|--|
| E11950    | Câble d'alimentation pour caméra/capteur           |
| E11898    | Câble avec connecteur M12 pour Industrial Ethernet |



Le logiciel ifm Vision Assistant est disponible à titre gratuit sur : [www.ifm.com](http://www.ifm.com)

## 6. Montage

Le chapitre décrit ce qu'il faut connaître avant le montage et comment l'appareil doit être installé.



- ① Appareil
- ② Angle d'ouverture
- ③ Objet
- ④ Champ de vue
- ⑤ Distance entre l'appareil et l'objet

### 6.1 Sélectionner le lieu de montage

Observer les instructions suivantes pour la sélection du lieu de montage :

- ▶ L'objet ③ doit se trouver intégralement dans le champ de vue ④.
- > La taille du champ de vue dépend de la version de l'appareil et est indiquée dans la fiche technique. La taille du champ de vue dépend aussi de la distance entre l'appareil et l'objet ⑤ : Le champ de vue devient plus grand lorsque l'on augmente la distance.
- ▶ Prendre en considération les tolérances pour le positionnement de l'objet.
- ▶ Pour définir la distance entre l'appareil et l'objet ⑤, prendre en considération l'étendue de mesure de l'appareil.
- > L'étendue de mesure est indiquée dans la fiche technique de l'appareil.
- ▶ Choisir une distance entre l'appareil et l'objet ⑤ aussi petite que possible.
- > Avec une distance minimale, l'objet est détecté avec la résolution maximale.
- ▶ Eviter une forte lumière ambiante et les rayons solaires sur le lieu de montage.
- > Un niveau de lumière ambiante (ayant un spectre solaire) au-dessus de 8 klx cause des erreurs de mesure. En effet, uniquement la lumière infrarouge entre 800 et 900 nm est perturbante.
- ▶ Eviter les zones à forts encrassements comme lieu de montage.
- > Dans des zones à forts encrassements, l'objectif est souillé malgré une orientation vers le bas ①.
- ▶ Eviter des vitres transparentes entre l'appareil ① et l'objet ③.
- > Les vitres transparentes réfléchissent une partie de la lumière, même si un plateau de verre très propre est utilisé.



Si les instructions ne sont pas respectées, des erreurs de mesure sont possibles.

## 6.2 Préparer l'appareil pour la mise en service

La température de surface de l'appareil dépend du mode de fonctionnement, de la sélection des paramètres et de la connexion thermique de l'appareil avec le support de montage.



S'assurer que l'appareil satisfait aux exigences suivantes :

La température de surface, pour les surfaces facilement accessibles, peut être au maximum supérieure de 25 °C à la température ambiante (selon CEI 61010-2-201).

Les diagrammes suivants montrent les limites d'avertissement typiques auxquelles l'installateur peut se reporter.



Les diagrammes sont valables pour les modes d'exposition suivants :

- un temps d'exposition
- deux temps d'exposition
- trois temps d'exposition

Avec deux ou trois temps d'exposition, les limites d'avertissement typiques doivent être déterminées en faisant la somme des temps d'exposition. Les temps d'exposition sont indiqués dans le logiciel ifm Vision Assistant.

Respecter une des indications suivantes si les limites d'avertissement sont dépassées :

- ▶ Réduire la température de surface (→ 6.2.3).
- ▶ Monter une protection contre le contact sans diminuer la convection (circulation d'air).
- > La température de surface ne doit pas être augmentée une fois cette protection contre le contact installée.

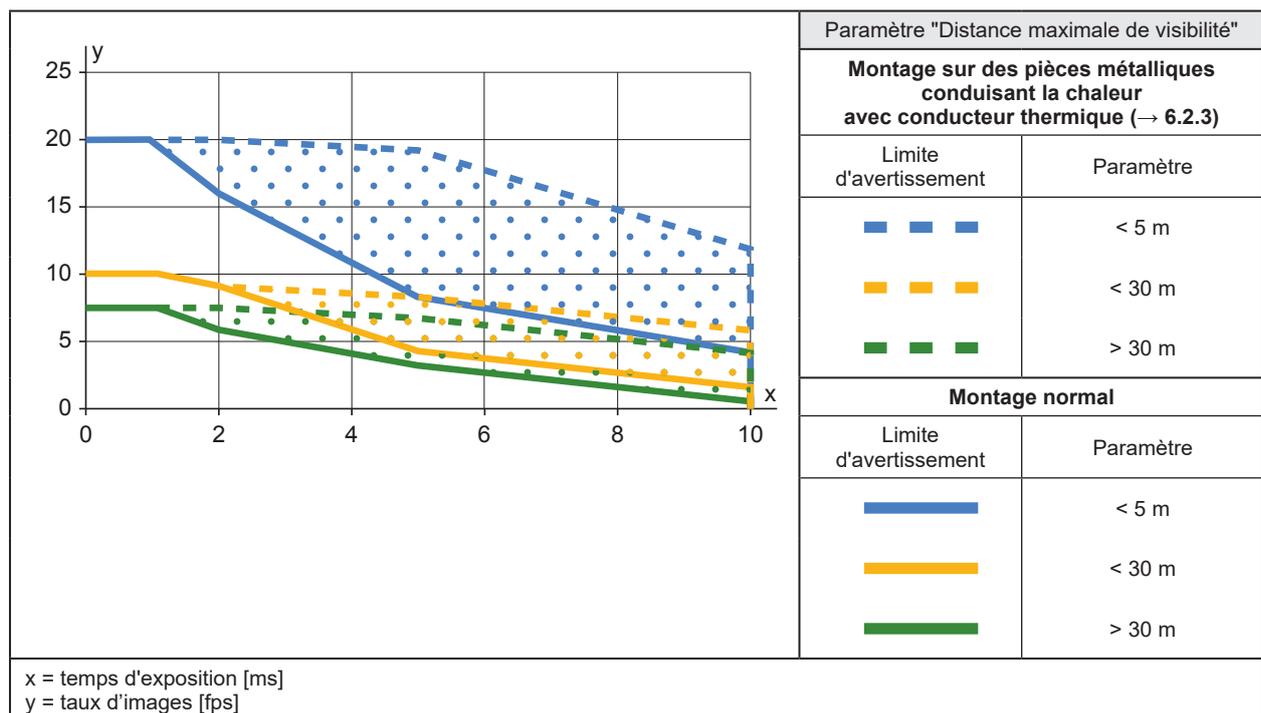


Le paramètre "Distance maximale de visibilité" est réglé dans le logiciel ifm Vision Assistant. Les limites d'avertissement sont indiquées dans les diagrammes comme des lignes pointillées et continues.

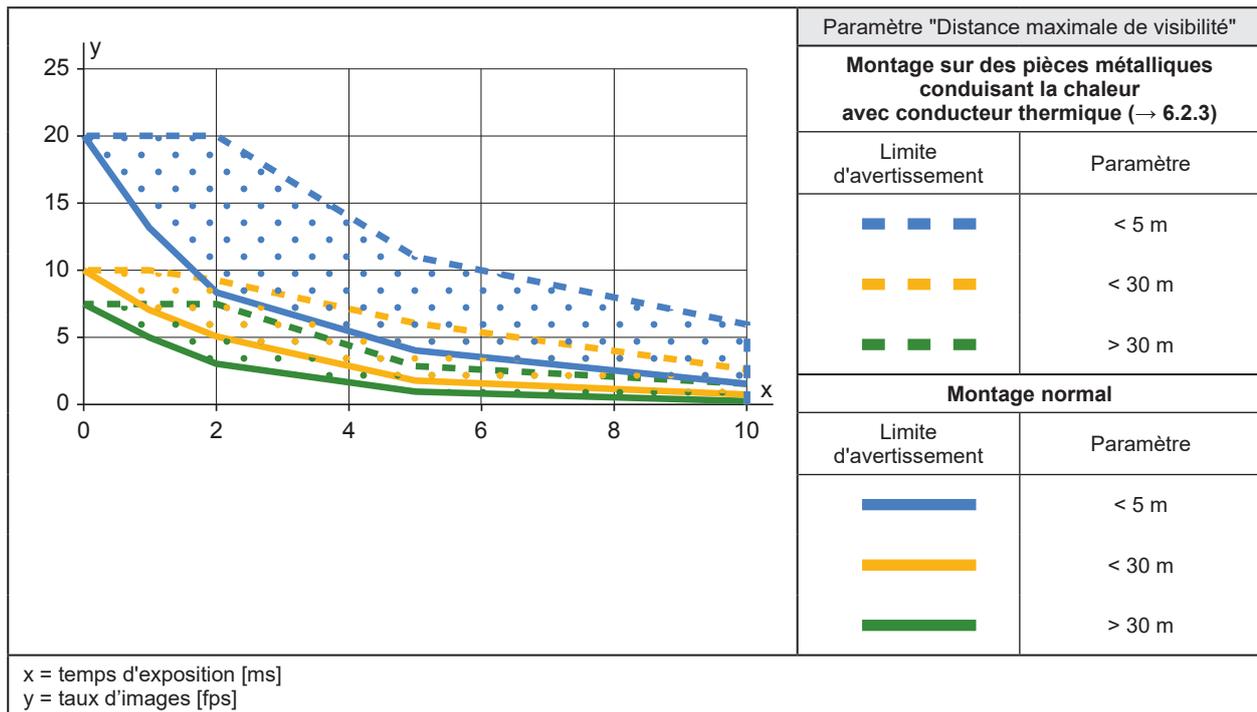
Si l'appareil se trouve dans une des zones pointillées, la température de surface doit être réduite (→ 6.2.3). Si la limite d'avertissement est dépassée malgré un montage évacuant la chaleur, une protection contre le contact peut être montée en plus.

Si les limites d'avertissement typiques ne sont pas atteintes en cas de montage normal, aucune mesure n'est nécessaire.

### 6.2.1 Limites d'avertissement typiques pour O3D300 / O3D302



## 6.2.2 Limites d'avertissement typiques pour O3D310 / O3D312



## 6.2.3 Réduire la température de surface

La température de surface peut être réduite par les mesures suivantes :

- ▶ Fixer l'appareil sur des pièces métalliques conduisant la chaleur.
  - > La fixation de l'appareil sur une grande surface métallique augmente l'évacuation de la chaleur (par ex. aluminium).
- ▶ Utiliser un conducteur thermique en cas de montage sur des pièces métalliques.
  - > L'effet conducteur de chaleur est augmenté par le conducteur thermique. Le conducteur thermique est disponible comme accessoire (→ 6.4).
- ▶ Réduire toute entrave dans l'environnement de l'appareil et la densité d'objets.
  - > Toute entrave dans l'environnement de l'appareil et une densité d'appareils augmentée peuvent avoir des effets négatifs sur la convection (mouvement de l'air).
- ▶ Monter un ou deux dissipateurs thermiques sur l'appareil.
  - > Les dissipateurs thermiques augmentent la surface de l'appareil ce qui réduit la température de surface. Les dissipateurs thermiques sont disponibles comme accessoires (→ 6.4).
- ▶ Réduire le temps d'exposition, le taux d'images ou la distance maximale de visibilité.
  - > Le mode de fonctionnement utilisé et les paramètres peuvent augmenter la température de surface.

### 6.3 Installer l'appareil

Respecter les instructions suivantes pour le montage de l'appareil :

- ▶ Monter l'appareil avec 2 vis M5 ou le kit de montage.
- > Les dimensions des perçages pour les vis M5 sont indiquées dans la fiche technique.
- > Le kit de montage est disponible comme accessoire (→ 6.4).
- ▶ Utiliser des protections contre la traction pour tous les câbles raccordés à l'appareil.

Respecter les remarques suivantes pour le montage d'un appareil O3D300 et O3D310 :

- ▶ Monter l'appareil de manière à permettre le réglage de la focale à l'aide d'un tournevis.
- > La position du réglage de la focale est indiquée dans le schéma d'encombrement (→ 12).



Lorsque l'appareil est utilisé en zone humide, l'écrou du câble avec connecteur M12 pour Industrial Ethernet (par ex. E11898) peut corroder. Pour l'utilisation permanente en zone humide utiliser un câble de raccordement avec un écrou en acier inox.

FR

### 6.4 Accessoires de montage

Suivant le lieu et le type de montage, les accessoires de montage suivants peuvent être utilisés :

| Référence | Description                           |
|-----------|---------------------------------------|
| E3D301    | Kit de montage Smart Camera           |
| E3D302    | Dissipateur thermique Smart Camera    |
| E3D303    | Conducteur thermique Smart Camera     |
| E3D304    | 2x dissipateur thermique Smart Camera |



Informations sur les accessoires disponibles sur : [www.ifm.com](http://www.ifm.com)

## 7. Raccordement électrique

Respecter les remarques suivantes avant l'installation électrique.

### ATTENTION

L'appareil doit être raccordé par un électricien qualifié. Respecter les données électriques de la fiche technique.

Appareil de la classe de protection III (CP III).

L'alimentation électrique ne doit s'effectuer que via des circuits TBTP.

L'alimentation électrique doit être conforme à UL 61010-1, chap. 9.4 - Limited Energy :

Le dispositif de limitation des surtensions doit désactiver un courant de 6,6 A en 120 s. Pour le dimensionnement du dispositif de limitation des surtensions, respecter les données techniques de l'appareil et du câblage.

L'isolation des circuits externes doit être conforme à UL 61010-2-201, fig. 102.

Pour des longueurs de câble > 30 m, utiliser une protection supplémentaire contre des tenues au choc selon CEI 6100-4-5.

Avant le raccordement électrique mettre l'installation hors tension.



Dans le champ d'application cULus :

Plage de température haute minimum du câble qui doit être raccordé au capteur : 70 °C.

### 7.1 Schéma de branchement

|   |   |   |      |   |                |   |      |   |   |   |                                 |   |                                 |   |                         |   |
|---|---|---|------|---|----------------|---|------|---|---|---|---------------------------------|---|---------------------------------|---|-------------------------|---|
|   | <b>① Ethernet</b><br>Connecteur femelle M12, codage D, 4 pôles  |   |      |   |                |   |      |   |   |   |                                 |   |                                 |   |                         |   |
|   | <table border="0"> <tr><td>1</td><td>TD +</td></tr> <tr><td>2</td><td>RD +</td></tr> <tr><td>3</td><td>TD -</td></tr> <tr><td>4</td><td>RD -</td></tr> <tr><td>S</td><td>Shield (blindage)</td></tr> </table>   | 1 | TD + | 2 | RD +           | 3 | TD - | 4 | RD -  | S | Shield (blindage)               |   |                                 |   |                         |   |
| 1 | TD +  |   |      |   |                |   |      |   |   |   |                                 |   |                                 |   |                         |   |
| 2 | RD +  |   |      |   |                |   |      |   |   |   |                                 |   |                                 |   |                         |   |
| 3 | TD -  |   |      |   |                |   |      |   |   |   |                                 |   |                                 |   |                         |   |
| 4 | RD -  |   |      |   |                |   |      |   |   |   |                                 |   |                                 |   |                         |   |
| S | Shield (blindage)   |   |      |   |                |   |      |   |   |   |                                 |   |                                 |   |                         |   |
|   | <b>② Alimentation en courant</b><br>Connecteur mâle M12, codage A, 8 pôles  |   |      |   |                |   |      |   |   |   |                                 |   |                                 |   |                         |   |
|   | <table border="0"> <tr><td>1</td><td>U+</td></tr> <tr><td>2</td><td>Entrée trigger</td></tr> <tr><td>3</td><td>GND</td></tr> <tr><td>4</td><td>Sortie de commutation 1 - (numérique ou analogique)</td></tr> <tr><td>5</td><td>Sortie de commutation 3 - Ready</td></tr> <tr><td>6</td><td>Sortie de commutation 2 - (TOR)</td></tr> <tr><td>7</td><td>Sortie de commutation 1</td></tr> <tr><td>8</td><td>Sortie de commutation 2</td></tr> </table> | 1 | U+   | 2 | Entrée trigger | 3 | GND  | 4 | Sortie de commutation 1 - (numérique ou analogique) | 5 | Sortie de commutation 3 - Ready | 6 | Sortie de commutation 2 - (TOR) | 7 | Sortie de commutation 1 | 8 |
| 1 | U+  |   |      |   |                |   |      |   |   |   |                                 |   |                                 |   |                         |   |
| 2 | Entrée trigger  |   |      |   |                |   |      |   |   |   |                                 |   |                                 |   |                         |   |
| 3 | GND   |   |      |   |                |   |      |   |   |   |                                 |   |                                 |   |                         |   |
| 4 | Sortie de commutation 1 - (numérique ou analogique)   |   |      |   |                |   |      |   |   |   |                                 |   |                                 |   |                         |   |
| 5 | Sortie de commutation 3 - Ready   |   |      |   |                |   |      |   |   |   |                                 |   |                                 |   |                         |   |
| 6 | Sortie de commutation 2 - (TOR)   |   |      |   |                |   |      |   |   |   |                                 |   |                                 |   |                         |   |
| 7 | Sortie de commutation 1   |   |      |   |                |   |      |   |   |   |                                 |   |                                 |   |                         |   |
| 8 | Sortie de commutation 2   |   |      |   |                |   |      |   |   |   |                                 |   |                                 |   |                         |   |



Protéger la prise de connexion Ethernet non utilisée avec un bouchon (E73004).  
Couple de serrage 0,6...0,8 Nm.



Le comportement des entrées et sorties de commutation peut être réglé par le logiciel ifm Vision Assistant. Le réglage commutation PNP ou NPN est toujours valable pour toutes les entrées et sorties.

S'assurer du réglage correct pour l'installation des actionneurs et capteurs (par ex. barrages optoélectroniques pour le déclenchement).

Les sorties de commutation peuvent aussi être utilisées comme sorties impulsionnelles qui remettent leur signal de commutation à 0 après un temps réglable.

La sortie analogique fournit du courant / de la tension contre GND.

### 7.1.1 Broches 1 / 3 (24 V / GND)

La plage de tension admissible est indiquée dans la fiche technique de l'appareil.

### 7.1.2 Broche 2 (entrée trigger)

La capture d'images de l'appareil peut être déclenchée avec un signal de commutation via l'entrée trigger.

Les fronts trigger suivants peuvent être utilisés :

- Le front descendant déclenche la capture d'images.
- Le front montant déclenche la capture d'images.
- Les fronts descendant et montant déclenchent la capture d'images.



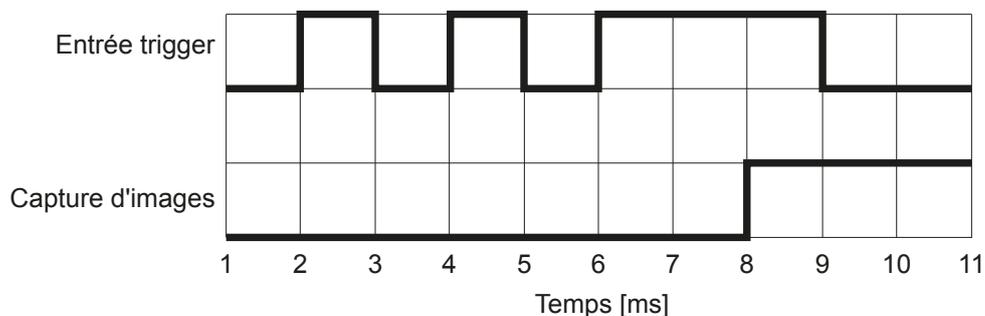
D'autres possibilités pour le déclenchement de l'appareil :

- Commande interfaces process (→ 13.2)
- Capture d'images continue avec taux d'images fixe



L'entrée trigger a une fonction anti-rebond intégrée. Selon l'installation électrique, cette fonction de l'entrée trigger n'est pas nécessaire.

La fonction anti-rebond intégrée évite le déclenchement d'un trigger par plusieurs courtes impulsions. L'impulsion doit durer au moins 2 ms pour être reconnue comme déclenchement.



### 7.1.3 Broches 4 / 5 / 6 (sorties de commutation)

Les sorties de commutation 1 à 3 fournissent les différents états de l'appareil. Outre l'état de l'appareil, les sorties de commutation peuvent aussi fournir les valeurs de référence nécessaires pour solutionner l'application.

Les spécifications électriques des sorties de commutation 1 à 3 sont indiquées dans la fiche technique.

Par défaut, la sortie de commutation 3 fournit l'état de l'appareil "Prêt pour le trigger".



"Sortie de commutation commutée" signifie que l'état de l'appareil correspondant s'est produit.

Selon le réglage, l'état de l'appareil peut avoir une des valeurs suivantes :

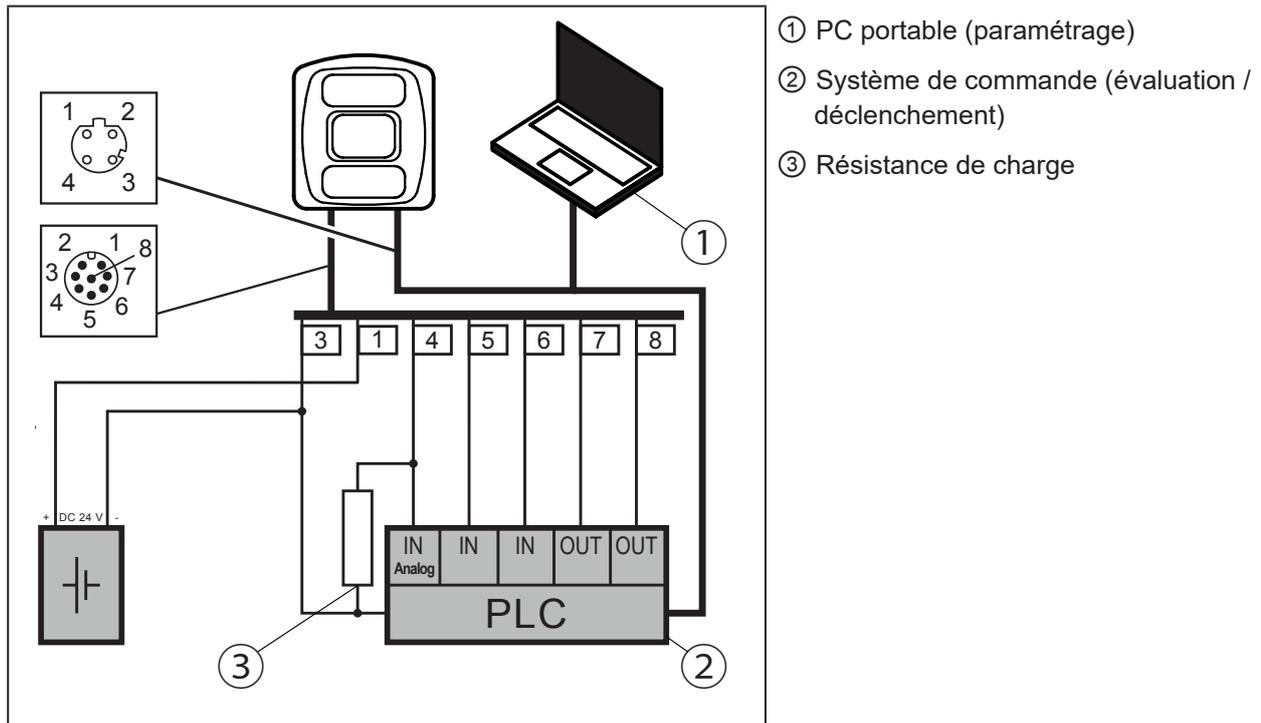
- "Prêt pour le trigger"  
L'appareil signale qu'une nouvelle image peut être capturée. Les triggers ne peuvent être traités qu'avec cet état de l'appareil. Avec capture d'images continue, l'état de l'appareil "Prêt pour le trigger" n'est pas fourni.
- "Capture d'images terminée"  
L'appareil signale que la capture des images est terminée. L'état de l'appareil peut être utilisé pour la fonction cascade d'appareils.
- "Evaluation finie"  
L'appareil signale que le traitement des images est terminé. A ce moment, les sorties de commutation sont déjà mises à jour. Les données d'image sont transmises via Ethernet.
- "Erreur"  
L'appareil signale qu'une erreur interne s'est produite. Des informations détaillées sur l'erreur peuvent être demandées via Ethernet.

### 7.1.4 Broche 4 (sortie analogique)

La sortie de commutation 1 / sortie analogique peut être utilisée comme sortie de commutation ou sortie courant analogique (4-20 mA) / sortie tension analogique (0-10 V).

La sortie courant analogique permet une transmission plus fiable que la sortie tension analogique. La sortie courant analogique est indépendante de la longueur du câble et assure une meilleure qualité de signal vers le système de commande.

Dans le système de commande, le courant analogique est converti en tension analogique via une résistance de charge contre GND. La résistance de charge est sélectionnée selon les indications dans la fiche technique. Pour un développement de chaleur plus faible dans l'appareil, les résistances de charge à haute impédance sont à préférer aux résistances de charge à faible impédance.



Avec le logiciel ifm Vision Assistant une valeur process peut être affectée à la valeur initiale (4 mA / 0 V) et à la valeur finale (20 mA / 10 V) de la sortie analogique.

### 7.1.5 Broches 7 / 8 (entrées de commutation)

Les entrées de commutation offrent les fonctions suivantes :

- Sélectionner l'application active (→ 7.3)



Les différents paramétrages des fonctions sont indiqués dans le manuel du logiciel.



Les spécifications électriques de l'entrée de commutation 1 et de l'entrée de commutation 2 sont indiquées dans la fiche technique de l'appareil.

## 7.2 Exemples de câblage

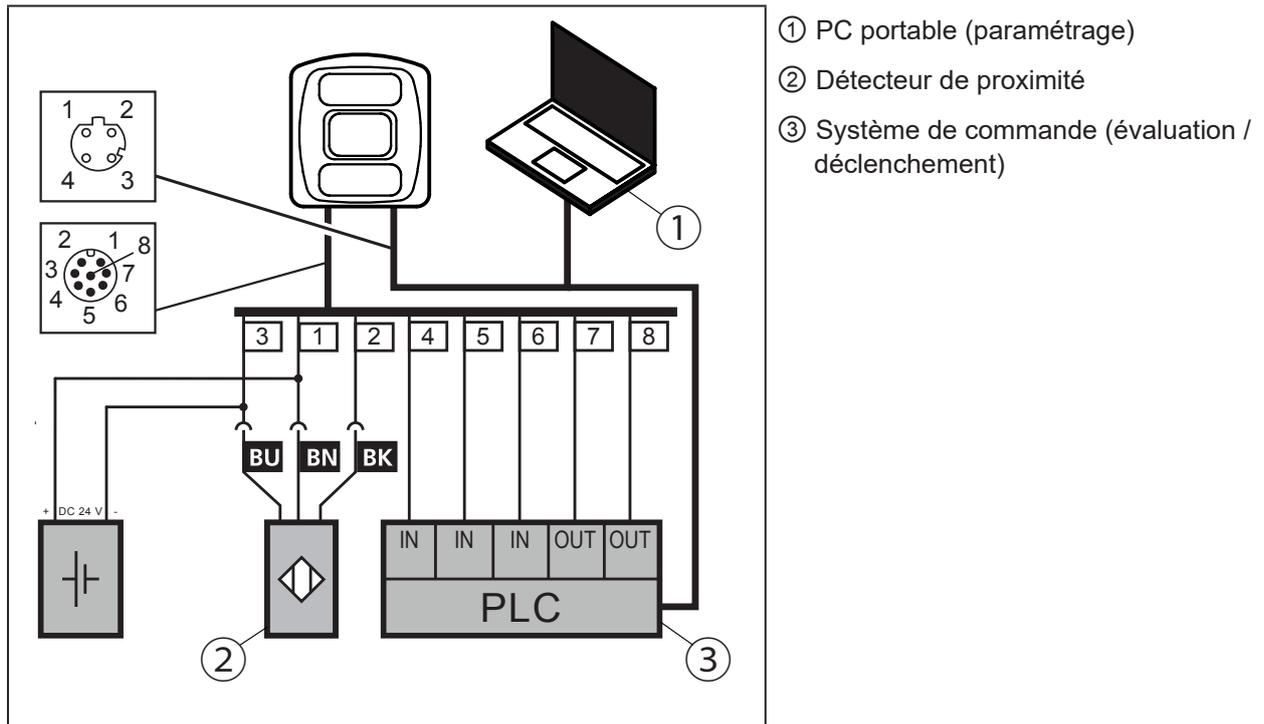
Ci-dessous, des exemples de câblage de l'appareil sont montrés.

### 7.2.1 Déclencher la capture d'images avec un détecteur de proximité

L'appareil peut être déclenché de manière externe :

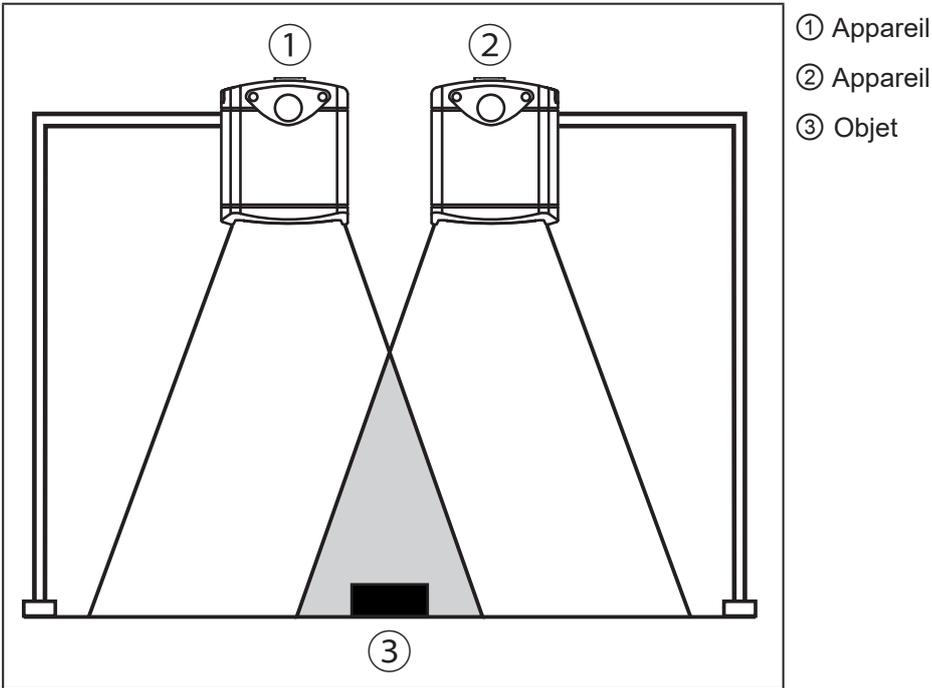
- via Ethernet
- via un détecteur de proximité, raccordé à l'entrée trigger

La figure suivante montre le câblage de l'appareil à un détecteur de proximité.



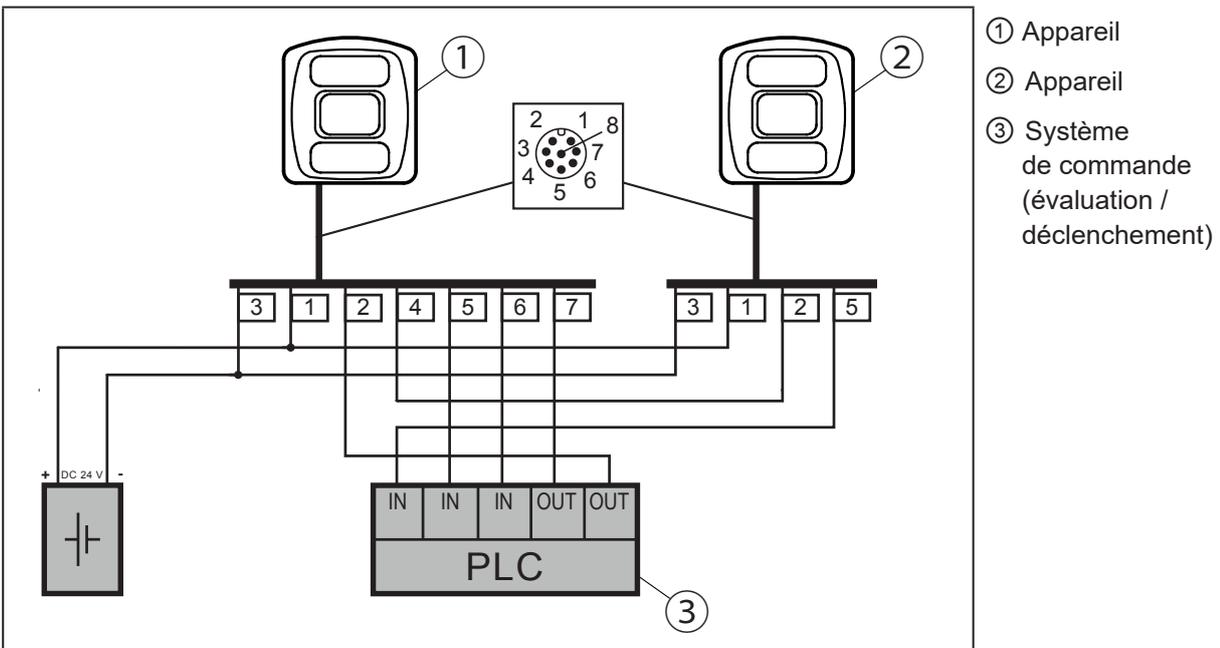
### 7.2.2 Monter plusieurs appareils l'un à côté de l'autre

Des appareils montés l'un à côté de l'autre peuvent causer des erreurs de mesure en cas d'exposition simultanée.



Il y a deux manières d'éviter les erreurs de mesure :

- Cascader les appareils via trigger matériel  
 Lors de cette fonction cascade, le système de commande déclenche la capture de l'appareil ① (voir fig. ci-dessous). Après la fin de la capture des images, l'appareil ① déclenche automatiquement l'appareil ②. En même temps, la broche 4 de l'appareil ① fournit l'état d'appareil "Capture d'images terminée". L'appareil ② signale la fin de la séquence au système de commande ③.



- Utiliser des canaux de fréquences différents.  
 Avec le logiciel ifm Vision Assistant un canal de fréquences individuel peut être affecté à chaque appareil. Les différents canaux de fréquences réduisent la présence d'erreurs de mesure.

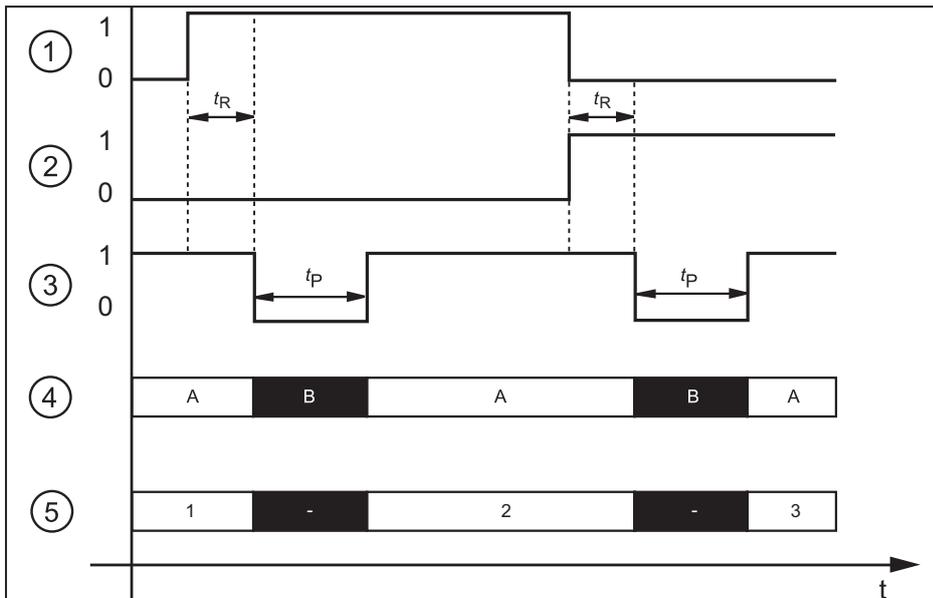


Le logiciel ifm Vision Assistant est disponible à titre gratuit sur : [www.ifm.com](http://www.ifm.com)

### 7.3 Sélection statique de l'application

L'appareil permet de mémoriser jusqu'à 32 applications d'inspection différentes. Avec la configuration correspondante, les quatre premières applications peuvent être sélectionnées via les deux entrées de commutation.

| Entrée 2 | Entrée 1 | N° de l'application |
|----------|----------|---------------------|
| 0        | 0        | 1                   |
| 0        | 1        | 2                   |
| 1        | 0        | 3                   |
| 1        | 1        | 4                   |



Exemple : Sélection application 1 → application 2 → application 3

|   |  |
|---|--|
| ① | Entrée de commutation 1 = 0 → 1 → 0        |
| ② | Entrée de commutation 2 = 0 → 0 → 1        |
| ③ | Sortie READY                               |
| ④ | Entrée trigger                             |
|   | A : trigger possible<br>B : trigger bloqué |
| ⑤ | Numéro ID de l'application active          |

Pour la sélection des applications, le temps de surveillance  $t_R$  et le temps de blocage du trigger  $t_P$  sont à considérer.

Temps de surveillance  $t_R$  : La sélection de l'application ne commence que si l'état des deux entrées de commutation reste stable pendant 20 ms après un changement d'état.

Temps de blocage du  $t_P$  : Pendant la sélection de l'application l'entrée trigger est bloquée. Le temps de blocage dépend :

- du nombre d'applications sur l'appareil
- du nombre de modèles dans l'application à activer



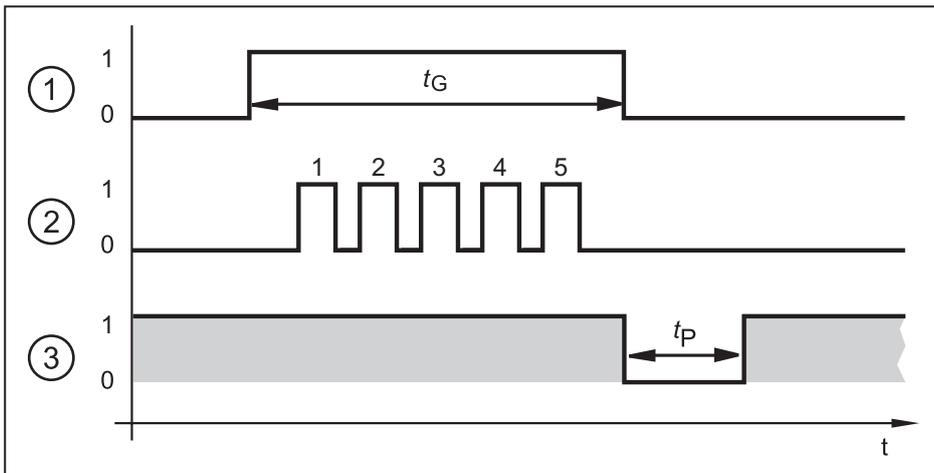
La figure ci-dessus montre la logique de sortie PNP (réglage par défaut). Le comportement de la logique de sortie PNP est à l'opposé de celui de la logique de sortie NPN :

- Logique de sortie PNP : En cas d'un signal haut (1), la tension est appliquée.
- Logique de sortie NPN : En cas d'un signal bas (0), la tension est appliquée.

Pour des informations plus détaillées sur la configuration de la sélection de l'application, voir le manuel du logiciel sur : [www.ifm.com](http://www.ifm.com)

## 7.4 Sélection de l'application activée par impulsion

Une sélection de l'application activée par impulsion est également possible.



|   |   |
|---|---|
| ① | Signal "fenêtre", entrée de commutation 1 = 0 → 1 → 0 ( $t_G$ = signal actif)     |
| ② | Signal impulsion, entrée de commutation 2 ou entrée trigger = 0 → 5 impulsion → 0 |
| ③ | Sortie READY  |

Si un signal actif est appliqué à l'entrée de commutation 1 (signal "fenêtre"), l'appareil compte les impulsions d'entrée et active l'application correspondante.

Nombre des impulsions = numéro ID de l'application.

L'entrée de commutation 2 ou l'entrée trigger de l'appareil peuvent être utilisées comme entrée impulsion.



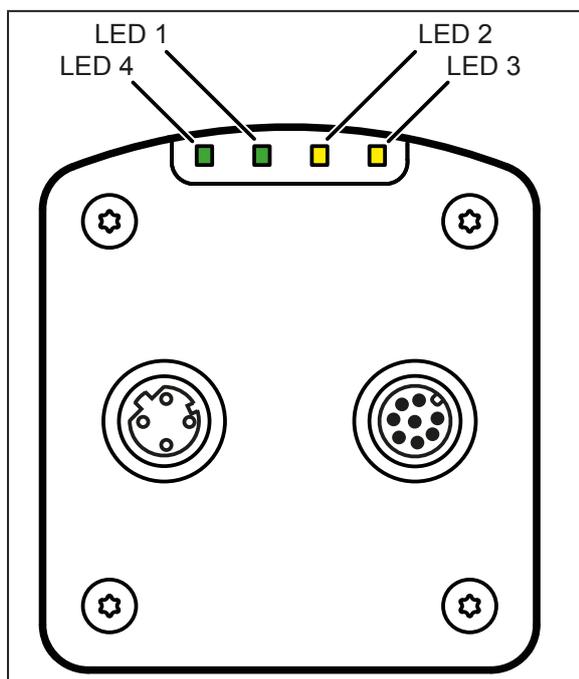
La figure ci-dessus montre la logique de sortie PNP (réglage par défaut). Le comportement de la logique de sortie PNP est à l'opposé de celui de la logique de sortie NPN :

- Logique de sortie PNP : En cas d'un signal haut (1), la tension est appliquée.
- Logique de sortie NPN : En cas d'un signal bas (0), la tension est appliquée.

Pour des informations plus détaillées sur la configuration de la sélection de l'application, voir le manuel du logiciel sur : [www.ifm.com](http://www.ifm.com)

## 8. Éléments de visualisation

L'appareil signale l'état de fonctionnement actuel via les éléments de visualisation LED 1 - 4.



FR

| LED 4<br>(Ethernet) | LED 1<br>(Power)      | LED 2<br>(Out 1) | LED 3<br>(Out 2) | Description  |
|---------------------|-----------------------|------------------|------------------|--|
|                     | Allumée               |                  |                  | L'appareil est opérationnel, tension d'alimentation appliquée  |
|                     | Clignote à 0,5 Hz     |                  |                  | L'appareil n'est pas paramétré ou le paramétrage n'a pas été chargé sur l'appareil<br>On <br>Off |
|                     | Clignote 2 x à 0,5 Hz |                  |                  | L'appareil est en mode de paramétrage<br>On <br>Off  |
|                     |                       | Allumée          |                  | La sortie de commutation 1 est commutée  |
|                     |                       | Clignote à 8 Hz  |                  | La sortie de commutation 1 a un court-circuit  |
|                     |                       |                  | Allumée          | La sortie de commutation 2 est commutée  |
|                     |                       |                  | Clignote à 8 Hz  | La sortie de commutation 2 a un court-circuit  |
| Allumée             |                       |                  |                  | Ethernet est raccordé  |
| Clignote            |                       |                  |                  | Ethernet transmet des données  |
| Eteinte             |                       |                  |                  | Ethernet n'est pas raccordé  |
|                     |                       | Clignote à 8 Hz  | Clignote à 8 Hz  | L'appareil signale une erreur interne  |
|                     |                       | Clignote à 2 Hz  | Clignote à 2 Hz  | L'appareil signale une erreur remédiable. Le message d'erreur peut être lu via Ethernet  |
|                     |                       | Chenillard ⇒     |                  | Appareil démarre   |
|                     |                       | Chenillard ⇐     |                  | L'appareil exécute une mise à jour du firmware   |

## 9. Mise en service

À la mise sous tension l'appareil est mis en service. Après 15 secondes, l'appareil se trouve en mode d'évaluation dans lequel les applications sauvegardées sont exécutées. Les éléments de visualisation signalent l'état de fonctionnement actuel (→ 8).



Jusqu'à 32 applications peuvent être sauvegardées sur l'appareil. Il y a différentes possibilités pour activer une application :

- Logiciel ifm Vision Assistant
- Commande interfaces process
- Entrées de commutation 1 et 2
- Entrée de commutation 1 et entrée trigger

### 9.1 Paramétrer l'appareil

L'appareil est paramétré avec le logiciel ifm Vision Assistant (→ voir Manuel du logiciel).



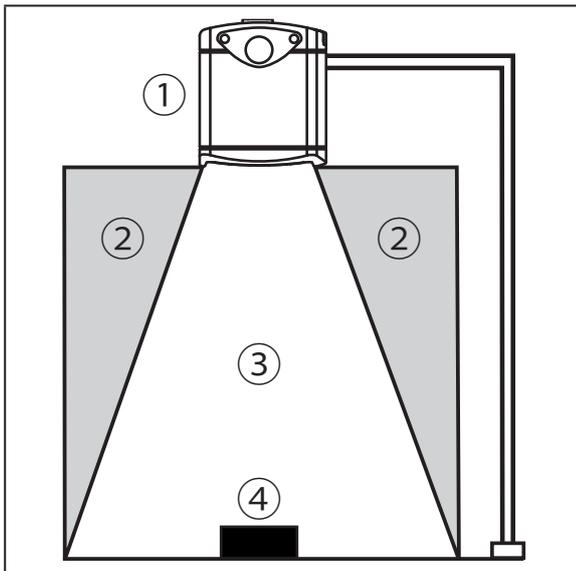
L'utilisation du logiciel ifm Vision Assistant et des informations détaillées sur le principe de mesure de l'appareil sont décrits dans le manuel du logiciel.

Le logiciel ifm Vision Assistant est disponible à titre gratuit sur : [www.ifm.com](http://www.ifm.com)

Le manuel du logiciel est disponible sur : [www.ifm.com](http://www.ifm.com)

### 9.2 Détecter un objet

Ci-dessous, les conditions qui mènent à un haut taux de détection d'objets sont décrites.



- ① Appareil
- ② Zone d'influence
- ③ Champ de vue
- ④ Objet

Un objet ④ est détecté de façon optimale, si les conditions suivantes sont satisfaites :

- L'objet est positionné dans le champ de vue ③.
- L'objet à détecter est l'objet le plus proche visible par l'appareil ①.
- La zone d'influence ② est libre d'objets (entrave etc.).
- La vitre avant de l'appareil est libre de salissures.



Si ces conditions ne sont pas satisfaites, des erreurs de mesure sont possibles.

### 9.3 Transmettre les valeurs process

#### 9.3.1 Transmettre les valeurs process du contrôle de complétude via EtherNet/IP

L'appareil peut transmettre les valeurs process à un API via le bus de terrain EtherNet/IP. Les valeurs process sont affichées dans le logiciel ifm Vision Assistant en tant qu'une chaîne de caractères comme suit :

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



Un seul bus de terrain ne peut être actif. Le bus de terrain peut être configuré (→ Manuel du logiciel).

Dans la chaîne de caractères les valeurs process sont séparées par un point-virgule. La chaîne de caractères est transmise à un API dans l'ordre affiché.



Veillez noter les remarques suivantes concernant la transmission de la chaîne de caractères à un API :

- Les octets 0 à 7 font partie de la chaîne de caractères. Ils ne sont pas affichés dans le logiciel ifm Vision Assistant (voir la copie d'écran ci-dessus).
- Les points-virgules ";" contenus dans la chaîne de caractères ne sont pas transmis.
- Les valeurs flottantes sont converties en entiers de 16 bits.
- Toutes les valeurs numériques sont converties en entiers de 16 bits avant la transmission.

La chaîne de caractères se décompose comme suit :

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

| N° octet | Données     | Codage  | Valeur process | Unité | Description                                   | Commentaire   |
|----------|-------------|---------|----------------|-------|---|---|
| 0        | 2#0000_0000 | TOR     | 1.5            |       | Mot de commande miroité                       | • Bit 1.5 indique une commande trigger réussie  |
| 1        | 2#0010_0000 | TOR     |                |       |   |   |
| 2        | 2#0000_0000 | Décimal |                |       | Identifiant de message synchrone / asynchrone |   |
| 3        | 2#0000_0000 | Décimal |                |       |   |   |
| 4        | 30          | Décimal | 30             |       | Compteur de messages                          | • L'appareil a reçu 30 messages<br>• Incrémente de 1 pour chaque action (trigger, message envoyé, etc.) |
| 5        | 0           | Décimal |                |       |   |   |
| 6        | 0           | Décimal |                |       | Réservé                                       |   |
| 7        | 0           | Décimal |                |       |   |   |
| 8        | s           | ASCII   | star           |       | Chaîne de départ                              |   |
| 9        | t           | ASCII   |                |       |   |   |
| 10       | a           | ASCII   |                |       |   |   |
| 11       | r           | ASCII   |                |       |   |   |
| 12       | 0           | Décimal | 0              |       | États de tous les ROI (0 = mauvais, 1 = bon)  | Affiche l'état du contrôle de complétude  |
| 13       | 0           | Décimal |                |       |   |   |

| N° octet | Données | Codage  | Valeur process | Unité | Description   | Commentaire   |
|----------|---------|---------|----------------|-------|---------------|---|
| 14       | 0       | Décimal |                |       |               | Les octets 14 et 15 sont utilisés par le suivi automatique de position si celui-ci est activé.<br>0 = la position n'est pas suivie<br>1 = la position est suivie<br>Toutes les données suivantes sont déplacées de 2 octets ; c'est-à-dire, le 1er ID ROI commence par les octets 16 et 17. |
| 15       | 0       | Décimal | 0              |       | ID ROI        |   |
| 16       | 0       | Décimal | 0              |       | Etat ROI      | Etat ROI :<br>0 = bon<br>1 = plan de référence non appris<br>2 = apprentissage échoué<br>3 = plan de référence non valable<br>4 = pas de pixels valables<br>5 = plan de référence ne contient pas de pixels valables<br>6 = état de dépassement<br>7 = état de manque                       |
| 17       | 0       | Décimal |                |       |               |   |
| 18       | 0       | Décimal | 0              | mm    | Valeur ROI    |   |
| 19       | 0       | Décimal |                |       |               |   |
| 20       | 1       | Décimal | 1              |       | ID ROI        |   |
| 21       | 0       | Décimal |                |       |               |   |
| 22       | 7       | Décimal | 7              |       | Etat ROI      |   |
| 23       | 0       | Décimal |                |       |               |   |
| 24       | -67     | Décimal | -67            | mm    | Valeur ROI    |   |
| 25       | -1      | Décimal |                |       |               |   |
| 26       | 2       | Décimal | 2              |       | ID ROI        |   |
| 27       | 0       | Décimal |                |       |               |   |
| 28       | 6       | Décimal | 6              |       | Etat ROI      |   |
| 29       | 0       | Décimal |                |       |               |   |
| 30       | 14      | Décimal | 14             | mm    | Valeur ROI    |   |
| 31       | 0       | Décimal |                |       |               |   |
| 32       | 3       | Décimal | 3              |       | ID ROI        |   |
| 33       | 0       | Décimal |                |       |               |   |
| 34       | 0       | Décimal | 0              |       | Etat ROI      |   |
| 35       | 0       | Décimal |                |       |               |   |
| 36       | 0       | Décimal | 0              | mm    | Valeur ROI    |   |
| 37       | 0       | Décimal |                |       |               |   |
| 38       | s       | ASCII   | stop           |       | Chaîne de fin |   |
| 39       | t       | ASCII   |                |       |               |   |
| 40       | o       | ASCII   |                |       |               |   |
| 41       | p       | ASCII   |                |       |               |   |



L'exécution erronée d'une commande aboutit à l'état suivant :

- Bit d'erreur = 1
- Le mot de commande miroité est affiché
- Bit de message asynchrone = 0
- Identifiant de message asynchrone = 0
- Le compteur de messages incrémente de 1

### 9.3.2 Transmettre les valeurs process du contrôle de complétude via PROFINET.

L'appareil peut transmettre les valeurs process à un API via le bus de terrain PROFINET. Les valeurs process sont affichées dans le logiciel ifm Vision Assistant en tant qu'une chaîne de caractères comme suit :

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

 Un seul bus de terrain ne peut être actif. Le bus de terrain peut être configuré (→ Manuel du logiciel).

Dans la chaîne de caractères les valeurs process sont séparées par un point-virgule. La chaîne de caractères est transmise à un API dans l'ordre affiché.

 Veuillez noter les remarques suivantes concernant la transmission de la chaîne de caractères à un API :

- Les octets 0 à 7 font partie de la chaîne de caractères. Ils ne sont pas affichés dans le logiciel ifm Vision Assistant (voir la copie d'écran ci-dessus).
- Les points-virgules ";" contenus dans la chaîne de caractères ne sont pas transmis.
- Les valeurs flottantes sont converties en entiers de 16 bits.
- Toutes les valeurs numériques sont converties en entiers de 16 bits avant la transmission.

La chaîne de caractères se décompose comme suit :

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

| N° octet | Données     | Codage  | Valeur process | Unité | Description                                   | Commentaire  |
|----------|-------------|---------|----------------|-------|---|--|
| 0        | 2#0010_0000 | TOR     | 0.5            |       | Mot de commande miroité                       | • Bit 0.5 indique une commande trigger réussie   |
| 1        | 2#0000_0000 | TOR     |                |       |   |  |
| 2        | 2#0000_0000 | Décimal |                |       | Identifiant de message synchrone / asynchrone |  |
| 3        | 2#0000_0000 | Décimal |                |       |   |  |
| 4        | 0           | Décimal | 30             |       | Compteur de messages                          | • L'appareil a reçu 30 messages<br>• Incréméte de 1 pour chaque action (trigger, message envoyé, etc.) |
| 5        | 30          | Décimal |                |       |   |  |
| 6        | 0           | Décimal |                |       | Réservé                                       |  |
| 7        | 0           | Décimal |                |       |   |  |
| 8        | s           | ASCII   | <b>star</b>    |       | Chaîne de départ                              |  |
| 9        | t           | ASCII   |                |       |   |  |
| 10       | a           | ASCII   |                |       |   |  |
| 11       | r           | ASCII   |                |       |   |  |
| 12       | 0           | Décimal | 0              |       | Etats de tous les ROI (0 = mauvais, 1 = bon)  | Affiche l'état du contrôle de complétude   |
| 13       | 0           | Décimal |                |       |   |  |

| N° octet | Données | Codage  | Valeur process | Unité | Description   | Commentaire   |
|----------|---------|---------|----------------|-------|---------------|---|
| 14       | 0       | Décimal |                |       |               | Les octets 14 et 15 sont utilisés par le suivi automatique de position si celui-ci est activé.<br>0 = la position n'est pas suivie<br>1 = la position est suivie<br>Toutes les données suivantes sont déplacées de 2 octets ; c'est-à-dire, le 1er ID ROI commence par les octets 16 et 17. |
| 15       | 0       | Décimal | 0              |       | ID ROI        |   |
| 16       | 0       | Décimal | 0              |       | Etat ROI      | Etat ROI :<br>0 = bon<br>1 = plan de référence non appris<br>2 = apprentissage échoué<br>3 = plan de référence non valable<br>4 = pas de pixels valables<br>5 = plan de référence ne contient pas de pixels valables<br>6 = état de dépassement<br>7 = état de manque                       |
| 17       | 0       | Décimal |                |       |               |   |
| 18       | 0       | Décimal | 0              | mm    | Valeur ROI    |   |
| 19       | 0       | Décimal |                |       |               |   |
| 20       | 0       | Décimal | 1              |       | ID ROI        |   |
| 21       | 1       | Décimal |                |       |               |   |
| 22       | 0       | Décimal | 7              |       | Etat ROI      |   |
| 23       | 7       | Décimal |                |       |               |   |
| 24       | -1      | Décimal | -67            | mm    | Valeur ROI    |   |
| 25       | -67     | Décimal |                |       |               |   |
| 26       | 0       | Décimal | 2              |       | ID ROI        |   |
| 27       | 2       | Décimal |                |       |               |   |
| 28       | 0       | Décimal | 6              |       | Etat ROI      |   |
| 29       | 6       | Décimal |                |       |               |   |
| 30       | 0       | Décimal | 14             | mm    | Valeur ROI    |   |
| 31       | 14      | Décimal |                |       |               |   |
| 32       | 0       | Décimal | 3              |       | ID ROI        |   |
| 33       | 3       | Décimal |                |       |               |   |
| 34       | 0       | Décimal | 0              |       | Etat ROI      |   |
| 35       | 0       | Décimal |                |       |               |   |
| 36       | 0       | Décimal | 0              | mm    | Valeur ROI    |   |
| 37       | 0       | Décimal |                |       |               |   |
| 38       | s       | ASCII   | stop           |       | Chaîne de fin |   |
| 39       | t       | ASCII   |                |       |               |   |
| 40       | o       | ASCII   |                |       |               |   |
| 41       | p       | ASCII   |                |       |               |   |



L'exécution erronée d'une commande aboutit à l'état suivant :

- Bit d'erreur = 1
- Le mot de commande miroité est affiché
- Bit de message asynchrone = 0
- Identifiant de message asynchrone = 0
- Le compteur de messages incrémente de 1

### 9.3.3 Transmettre les valeurs process du contrôle de complétude via TCP/IP

L'appareil peut transmettre les valeurs process à un API via le protocole TCP/IP. Les valeurs process sont affichées dans le logiciel ifm Vision Assistant en tant qu'une chaîne de caractères comme suit :

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

Dans la chaîne de caractères les valeurs process sont séparées par un point-virgule. La chaîne de caractères est transmise à un API dans l'ordre affiché.



Veillez noter les remarques suivantes concernant la transmission de la chaîne de caractères à un API :

- Les points-virgules ";" contenus dans la chaîne de caractères ne sont pas transmis.
- Toutes les valeurs numériques sont converties en entiers de 16 bits avant la transmission.

La chaîne de caractères se décompose comme suit (type de données : ASCII) :

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

| Valeur process | Unité | Description                                  |
|----------------|-------|--|
| star           |       | Chaîne de départ                             |
| 0              |       | Etats de tous les ROI (0 = mauvais, 1 = bon) |
| 00             |       | ID ROI                                       |
| 0              |       | Etat ROI                                     |
| +0.000         | m     | Valeur ROI                                   |
| 01             |       | ID ROI                                       |
| 7              |       | Etat ROI                                     |
| -0.068         | m     | Valeur ROI                                   |
| 02             |       | ID ROI                                       |
| 6              |       | Etat ROI                                     |
| +0.013         | m     | Valeur ROI                                   |
| 03             |       | ID ROI                                       |
| 0              |       | Etat ROI                                     |
| +0.001         | m     | Valeur ROI                                   |
| stop           |       | Chaîne de fin                                |

Etat ROI :

- 0 = bon
- 1 = plan de référence non appris
- 2 = apprentissage échoué
- 3 = plan de référence non valable
- 4 = pas de pixels valables
- 5 = plan de référence ne contient pas de pixels valables
- 6 = état de dépassement
- 7 = état de manque

### 9.3.4 Transmettre les valeurs process du dimensionnement d'objets via EtherNet/IP

L'appareil peut transmettre les valeurs process à un API via le bus de terrain EtherNet/IP. Les valeurs process sont affichées dans le logiciel ifm Vision Assistant en tant qu'une chaîne de caractères comme suit :

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

 Un seul bus de terrain ne peut être actif. Le bus de terrain peut être configuré (→ Manuel du logiciel).

Dans la chaîne de caractères les valeurs process sont séparées par un point-virgule. La chaîne de caractères est transmise à un API dans l'ordre affiché.

 Veuillez noter les remarques suivantes concernant la transmission de la chaîne de caractères à un API :

- La chaîne de caractères est configurable. Dans le logiciel ifm Vision Assistant on peut choisir quelles valeurs process doivent être transmises.
- Les octets 0 à 7 font partie de la chaîne de caractères. Ils ne sont pas affichés dans le logiciel ifm Vision Assistant (voir la copie d'écran ci-dessus).
- Les points-virgules ";" contenus dans la chaîne de caractères ne sont pas transmis.
- Les valeurs flottantes sont converties en entiers de 16 bits.
- Toutes les valeurs numériques sont converties en entiers de 16 bits avant la transmission.

La chaîne de caractères se décompose comme suit :

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

| N° octet | Données     | Codage  | Valeur process | Unité | Description                                   | Commentaire  |
|----------|-------------|---------|----------------|-------|---|--|
| 0        | 2#0000_0000 | TOR     | 1.5            |       | Mot de commande miroité                       | • Bit 1.5 indique une commande trigger réussie   |
| 1        | 2#0010_0000 | TOR     |                |       |   |  |
| 2        | 2#0000_0000 | TOR     |                |       | Identifiant de message synchrone / asynchrone |  |
| 3        | 2#0000_0000 | TOR     |                |       |   |  |
| 4        | 2#0000_0011 | TOR     | 3              |       | Compteur de messages                          | • L'appareil a reçu 3 messages<br>• Incrémenté de 1 pour chaque action (trigger, message envoyé, etc.) |
| 5        | 2#0000_0000 | TOR     |                |       |   |  |
| 6        | 2#0000_0000 | TOR     |                |       | Réservé                                       |  |
| 7        | 2#0000_0000 | TOR     |                |       |   |  |
| 8        | s           | ASCII   | star           |       | Chaîne de départ                              |  |
| 9        | t           | ASCII   |                |       |   |  |
| 10       | a           | ASCII   |                |       |   |  |
| 11       | r           | ASCII   |                |       |   |  |
| 12       | 2#0000_0001 | TOR     | 1              |       | Bit de résultat                               | 0 = aucune boîte trouvée<br>1 = boîte trouvée  |
| 13       | 2#0000_0000 | TOR     |                |       |   |  |
| 14       | 104         | Décimal | 104            | mm    | Largeur                                       |  |
| 15       | 0           | Décimal |                |       |   |  |
| 16       | 88          | Décimal | 88             | mm    | Hauteur                                       |  |
| 17       | 0           | Décimal |                |       |   |  |
| 18       | 108         | Décimal | 109            | mm    | Longueur                                      |  |
| 19       | 0           | Décimal |                |       |   |  |

| N° octet | Données | Codage  | Valeur process | Unité | Description       | Commentaire |
|----------|---------|---------|----------------|-------|-------------------|-------------|
| 20       | 21      | Décimal | 21             |       | Coordonnée x      |             |
| 21       | 0       | Décimal |                |       |                   |             |
| 22       | -11     | Décimal | -11            |       | Coordonnée y      |             |
| 23       | -1      | Décimal |                |       |                   |             |
| 24       | -124    | Décimal | 389            |       | Coordonnée z      |             |
| 25       | 1       | Décimal |                |       |                   |             |
| 26       | -98     | Décimal | 158            |       | Angle de rotation |             |
| 27       | 0       | Décimal |                |       |                   |             |
| 28       | 97      | Décimal | 97             |       | Qualité largeur   |             |
| 29       | 0       | Décimal |                |       |                   |             |
| 30       | 93      | Décimal | 94             |       | Qualité hauteur   |             |
| 31       | 0       | Décimal |                |       |                   |             |
| 32       | 97      | Décimal | 97             |       | Qualité longueur  |             |
| 33       | 0       | Décimal |                |       |                   |             |
| 34       | s       | ASCII   | stop           |       | Chaîne de fin     |             |
| 35       | t       | ASCII   |                |       |                   |             |
| 36       | o       | ASCII   |                |       |                   |             |
| 37       | p       | ASCII   |                |       |                   |             |



L'exécution erronée d'une commande aboutit à l'état suivant :

- Bit d'erreur = 1
- Le mot de commande miroité est affiché
- Bit de message asynchrone = 0
- Identifiant de message asynchrone = 0
- Le compteur de messages incrémente de 1

### 9.3.5 Transmettre les valeurs process du dimensionnement d'objets via PROFINET

L'appareil peut transmettre les valeurs process à un API via le bus de terrain PROFINET. Les valeurs process sont affichées dans le logiciel ifm Vision Assistant en tant qu'une chaîne de caractères comme suit :

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



Un seul bus de terrain ne peut être actif. Le bus de terrain peut être configuré (→ Manuel du logiciel).

Dans la chaîne de caractères les valeurs process sont séparées par un point-virgule. La chaîne de caractères est transmise à un API dans l'ordre affiché.



Veillez noter les remarques suivantes concernant la transmission de la chaîne de caractères à un API :

- La chaîne de caractères est configurable. Dans le logiciel ifm Vision Assistant on peut choisir quelles valeurs process doivent être transmises.
- Les octets 0 à 7 font partie de la chaîne de caractères. Ils ne sont pas affichés dans le logiciel ifm Vision Assistant (voir la copie d'écran ci-dessus).
- Les points-virgules ";" contenus dans la chaîne de caractères ne sont pas transmis.
- Les valeurs flottantes sont converties en entiers de 16 bits.
- Toutes les valeurs numériques sont converties en entiers de 16 bits avant la transmission.

La chaîne de caractères se décompose comme suit :

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

| N° byte | Données     | Codage  | Valeur process | Unité | Description                                   | Commentaire  |
|---------|-------------|---------|----------------|-------|---|--|
| 0       | 2#0010_0000 | TOR     | 0.5            |       | Mot de commande miroité                       | • Bit 0.5 indique une commande trigger réussie   |
| 1       | 2#0000_0000 | TOR     |                |       |   |  |
| 2       | 2#0000_0000 | TOR     |                |       | Identifiant de message synchrone / asynchrone |  |
| 3       | 2#0000_0000 | TOR     |                |       |   |  |
| 4       | 2#0000_0000 | TOR     | 3              |       | Compteur de messages                          | • L'appareil a reçu 3 messages<br>• Incrémente de 1 pour chaque action (trigger, message envoyé, etc.) |
| 5       | 2#0000_0011 | TOR     |                |       |   |  |
| 6       | 2#0000_0000 | TOR     |                |       | Réservé                                       |  |
| 7       | 2#0000_0000 | TOR     |                |       |   |  |
| 8       | s           | ASCII   | <b>star</b>    |       | Chaîne de départ                              |  |
| 9       | t           | ASCII   |                |       |   |  |
| 10      | a           | ASCII   |                |       |   |  |
| 11      | r           | ASCII   |                |       |   |  |
| 12      | 2#0000_0000 | TOR     | 1              |       | Bit de résultat                               | 0 = aucune boîte trouvée<br>1 = boîte trouvée  |
| 13      | 2#0000_0001 | TOR     |                |       |   |  |
| 14      | 0           | Décimal | 104            | mm    | Largeur                                       |  |
| 15      | 104         | Décimal |                |       |   |  |
| 16      | 0           | Décimal | 88             | mm    | Hauteur                                       |  |
| 17      | 88          | Décimal |                |       |   |  |
| 18      | 0           | Décimal | 109            | mm    | Longueur                                      |  |
| 19      | 109         | Décimal |                |       |   |  |

| N° byte | Données | Codage  | Valeur process | Unité | Description       | Commentaire |
|---------|---------|---------|----------------|-------|-------------------|-------------|
| 20      | 0       | Décimal | 21             |       | Coordonnée x      |             |
| 21      | 21      | Décimal |                |       |                   |             |
| 22      | -1      | Décimal | -11            |       | Coordonnée y      |             |
| 23      | -11     | Décimal |                |       |                   |             |
| 24      | 1       | Décimal | 389            |       | Coordonnée z      |             |
| 25      | -124    | Décimal |                |       |                   |             |
| 26      | 0       | Décimal | 158            |       | Angle de rotation |             |
| 27      | -98     | Décimal |                |       |                   |             |
| 28      | 0       | Décimal | 97             |       | Qualité largeur   |             |
| 29      | 97      | Décimal |                |       |                   |             |
| 30      | 0       | Décimal | 94             |       | Qualité hauteur   |             |
| 31      | 94      | Décimal |                |       |                   |             |
| 32      | 0       | Décimal | 97             |       | Qualité longueur  |             |
| 33      | 97      | Décimal |                |       |                   |             |
| 34      | s       | ASCII   | stop           |       | Chaîne de fin     |             |
| 35      | t       | ASCII   |                |       |                   |             |
| 36      | o       | ASCII   |                |       |                   |             |
| 37      | p       | ASCII   |                |       |                   |             |



L'exécution erronée d'une commande aboutit à l'état suivant :

- Bit d'erreur = 1
- Le mot de commande miroité est affiché
- Bit de message asynchrone = 0
- Identifiant de message asynchrone = 0
- Le compteur de messages incrémente de 1

### 9.3.6 Transmettre les valeurs process du dimensionnement d'objets via TCP/IP

L'appareil peut transmettre les valeurs process à un API via le protocole TCP/IP. On peut sélectionner dans le logiciel ifm Vision Assistant quelles valeurs process sont à transmettre. Les valeurs process sont affichées dans le logiciel ifm Vision Assistant en tant qu'une chaîne de caractères comme suit :

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

Dans la chaîne de caractères les valeurs process sont séparées par un point-virgule. La chaîne de caractères est transmise à un API dans l'ordre affiché.



Veillez noter les remarques suivantes concernant la transmission de la chaîne de caractères à un API :

- Les points-virgules ";" contenus dans la chaîne de caractères ne sont pas transmis.
- Toutes les valeurs numériques sont converties en entiers de 16 bits avant la transmission.

La chaîne de caractères se décompose comme suit (type de données : ASCII) :

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

| Valeur process | Unité | Description       |
|----------------|-------|-------------------|
| star           |       | Chaîne de départ  |
| 1              |       | Objet trouvé      |
| 0 104          | m     | Largeur           |
| 0 088          | m     | Hauteur           |
| 0 109          | m     | Longueur          |
| +0 021         |       | Coordonnée x      |
| -0 011         |       | Coordonnée y      |
| +0 389         |       | Coordonnée z      |
| 158            |       | Angle de rotation |
| 097            |       | Qualité largeur   |
| 094            |       | Qualité hauteur   |
| 097            |       | Qualité longueur  |
| stop           |       | Chaîne de fin     |

### 9.3.7 Transmettre les valeurs process de la mesure de niveau via EtherNet/IP

L'appareil peut transmettre les valeurs process à un API via le bus de terrain EtherNet/IP. Les valeurs process sont affichées dans le logiciel ifm Vision Assistant en tant qu'une chaîne de caractères comme suit :



 Un seul bus de terrain ne peut être actif. Le bus de terrain peut être configuré (→ Manuel du logiciel).

La chaîne de caractères est transmise à un API dans l'ordre affiché.

 Veuillez noter les remarques suivantes concernant la transmission de la chaîne de caractères à un API :

- Les octets 0 à 7 font partie de la chaîne de caractères. Ils ne sont pas affichés dans le logiciel ifm Vision Assistant (voir la copie d'écran ci-dessus).
- Les points-virgules ";" contenus dans la chaîne de caractères ne sont pas transmis.
- Les valeurs flottantes sont converties en entiers de 16 bits.
- Toutes les valeurs numériques sont converties en entiers de 16 bits avant la transmission.

La chaîne de caractères se décompose comme suit :

0070

| N° octet | Données     | Codage  | Valeur process | Unité | Description                                   | Commentaire   |
|----------|-------------|---------|----------------|-------|---|---|
| 0        | 2#0000_0000 | TOR     | 1.5            |       | Mot de commande miroité                       | Bit 1.5 indique une commande trigger réussie  |
| 1        | 2#0010_0000 | TOR     |                |       |   |   |
| 2        | 2#0000_0000 | Décimal | 30             |       | Identifiant de message synchrone / asynchrone |   |
| 3        | 2#0000_0000 | Décimal |                |       |   |   |
| 4        | 30          | Décimal |                |       |   |   |
| 5        | 0           | Décimal |                |       | Compteur de messages                          | <ul style="list-style-type: none"> <li>• L'appareil a reçu 30 messages</li> <li>• Incrémente de 1 pour chaque action (trigger, message envoyé, etc.)</li> </ul> |
| 6        | 0           | Décimal |                |       | Réservé                                       |   |
| 7        | 0           | Décimal |                |       |   |   |
| 8        | 0           | Décimal | 0              |       | Etats de tous les ROI (0 = mauvais, 1 = bon)  | Affiche l'état de la mesure de niveau   |
| 9        | 0           | Décimal |                |       |   |   |
| 10       | 0           | Décimal | 0              |       | ID ROI  | Etat ROI :<br>0 = bon<br>6 = état de dépassement<br>7 = état de manque  |
| 11       | 0           | Décimal |                |       |   |   |
| 12       | 7           | Décimal | 7              |       | Etat ROI                                      |   |
| 13       | 0           | Décimal |                |       |   |   |
| 14       | 0           | Décimal | 0              | mm    | Valeur ROI                                    |   |
| 15       | 0           | Décimal |                |       |   |   |

FR



L'exécution erronée d'une commande aboutit à l'état suivant :

- Bit d'erreur = 1
- Le mot de commande miroité est affiché
- Bit de message asynchrone = 0
- Identifiant de message asynchrone = 0
- Le compteur de messages incrémente de 1

### 9.3.8 Transmettre les valeurs process de la mesure de niveau via PROFINET

L'appareil peut transmettre les valeurs process à un API via le bus de terrain PROFINET. Les valeurs process sont affichées dans le logiciel ifm Vision Assistant en tant qu'une chaîne de caractères comme suit :

0070



Un seul bus de terrain ne peut être actif. Le bus de terrain peut être configuré (→ Manuel du logiciel).

La chaîne de caractères est transmise à un API dans l'ordre affiché.



Veillez noter les remarques suivantes concernant la transmission de la chaîne de caractères à un API :

- Les octets 0 à 7 font partie de la chaîne de caractères. Ils ne sont pas affichés dans le logiciel ifm Vision Assistant (voir la copie d'écran ci-dessus).
- Les points-virgules ";" contenus dans la chaîne de caractères ne sont pas transmis.
- Les valeurs flottantes sont converties en entiers de 16 bits.
- Toutes les valeurs numériques sont converties en entiers de 16 bits avant la transmission.

La chaîne de caractères se décompose comme suit :

0070

| N° byte | Données     | Codage  | Valeur process | Unité | Description                                  | Commentaire  |
|---------|-------------|---------|----------------|-------|--|--|
| 0       | 2#0010_0000 | TOR     | 0.5            |       | Mot de commande miroité                      | Bit 0.5 indique une commande trigger réussie   |
| 1       | 2#0000_0000 | TOR     |                |       |  |  |
| 2       | 2#0000_0000 | Décimal |                |       | Identifiant message synchrone / asynchrone   |  |
| 3       | 2#0000_0000 | Décimal |                |       |  |  |
| 4       | 0           | Décimal | 30             |       | Compteur de messages                         | <ul style="list-style-type: none"> <li>• L'appareil a reçu 30 messages</li> <li>• Incréméte de 1 pour chaque action (trigger, message envoyé, etc.)</li> </ul> |
| 5       | 30          | Décimal |                |       |  |  |
| 6       | 0           | Décimal |                |       | Réservé                                      |  |
| 7       | 0           | Décimal |                |       |  |  |
| 8       | 0           | Décimal | 0              |       | Etats de tous les ROI (0 = mauvais, 1 = bon) | Affiche l'état de la mesure de niveau  |
| 9       | 0           | Décimal |                |       |  |  |
| 10      | 0           | Décimal | 0              |       | ID ROI                                       | Etat ROI :<br>0 = bon<br>6 = état de dépassement<br>7 = état de manque   |
| 11      | 0           | Décimal |                |       |  |  |
| 12      | 0           | Décimal | 7              |       | Etat ROI                                     |  |
| 13      | 7           | Décimal |                |       |  |  |
| 14      | 0           | Décimal | 0              | mm    | Valeur ROI                                   |  |
| 15      | 0           | Décimal |                |       |  |  |



L'exécution erronée d'une commande aboutit à l'état suivant :

- Bit d'erreur = 1
- Le mot de commande miroité est affiché
- Bit de message asynchrone = 0
- Identifiant de message asynchrone = 0
- Le compteur de messages incrémente de 1

### 9.3.9 Transmettre les valeurs process de la mesure de niveau via TCP/IP

L'appareil peut transmettre les valeurs process à un API via le protocole TCP/IP. Les valeurs process sont affichées dans le logiciel ifm Vision Assistant en tant qu'une chaîne de caractères comme suit :

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

Dans la chaîne de caractères les valeurs process sont séparées par un point-virgule. La chaîne de caractères est transmise à un API dans l'ordre affiché.



Veillez noter les remarques suivantes concernant la transmission de la chaîne de caractères à un API :

- Les points-virgules ";" contenus dans la chaîne de caractères ne sont pas transmis.
- Toutes les valeurs numériques sont converties en entiers de 16 bits avant la transmission.

La chaîne de caractères se décompose comme suit (type de données : ASCII) :

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

| Valeur process | Unité | Description                                  |
|----------------|-------|--|
| star           |       | Chaîne de départ                             |
| 0              |       | États de tous les ROI (0 = mauvais, 1 = bon) |
| 00             |       | ID ROI                                       |
| 7              |       | État ROI                                     |
| +0 000         | m     | Valeur ROI                                   |
| stop           |       | Chaîne de fin                                |

État ROI :  
 0 = bon  
 6 = état de dépassement  
 7 = état de manque

## 10. Maintenance, réparation et élimination

Respecter les instructions suivantes :

- ▶ Ne pas ouvrir l'appareil. A l'intérieur de l'appareil il n'y a pas de composants réparables par l'utilisateur. L'appareil ne doit être réparé que par le fabricant.
- ▶ Respecter la réglementation du pays en vigueur pour la destruction écologique de l'appareil.

### 10.1 Nettoyage

Observer les instructions suivantes avant le nettoyage de l'appareil :

- ▶ Utiliser un chiffon propre et sans peluches.
- ▶ Comme produit lessiviel, utiliser un nettoyant à vitres.



Si ces instructions ne sont pas respectées, des erreurs de mesure sur la vitre avant, causées par des rayures, sont possibles.

### 10.2 Mettre à jour le firmware

Le firmware de l'appareil peut être mis à jour avec le logiciel ifm Vision Assistant.



Les paramètres sauvegardés dans l'appareil sont perdus par la mise à jour du firmware. Avant la mise à jour du firmware, faire une copie de sauvegarde des paramètres.

- ▶ Avant la mise à jour du firmware, exporter les paramètres.
- ▶ Après la mise à jour du firmware, importer les paramètres.



Les mises à jour du firmware sont disponibles sur : [www.ifm.com](http://www.ifm.com)

### 10.3 Remplacer l'appareil

Lors du remplacement d'un appareil les paramètres sont perdus. Avant le remplacement de l'appareil, faire une copie de sauvegarde des paramètres.

- ▶ Avant le remplacement, exporter les paramètres de l'appareil à remplacer.
- ▶ Après le remplacement, importer les paramètres dans le nouvel appareil.



Avec l'exportation et l'importation des paramètres il est possible d'équiper plusieurs appareils rapidement avec les mêmes paramètres.

## 11. Homologations/normes

La déclaration de conformité UE est disponible sur : [www.ifm.com](http://www.ifm.com)



## 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

|     |   |    |   |
|-----|---|----|---|
| CR  | Carriage Return   | 13 | D |
| LF  | Linefeed  | 10 | A |
| < > | Marking of a placeholder<br>(e.g. <code> is a placeholder for code) |    |   |
| [ ] | Optional argument<br>(possible but not required)                    |    |   |

|           |  |
|-----------|--|
| <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.<br> A ticket number must be > 0999.<br>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:

|    |   |                             |
|----|---|-----------------------------|
| 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:

|      |  |
|------|--|
| 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:

|                     |  |
|---------------------|--|
| <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

|           |                          |  |  |
|-----------|--------------------------|--|--|
| 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 according to the application. If there is no application on given index, the application ID will contain 0 and the name an empty string "". |

### 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:

|   |                         |
|---|-------------------------|
| Ticket and length information   | (→ 13.2.2)              |
| Ticket  | "0000"                  |
| Start sequence  | String "star" (4 bytes) |
| Normalised amplitude image<br>Output format: 16-bit unsigned integer  | 1 image                 |
| Distance image<br>Output format: 16-bit unsigned integer.<br>Unit: mm | 1 image                 |
| X image<br>Output format: 16-bit signed integer.<br>Unit: mm          | 1 image                 |
| Y image<br>Output format: 16-bit signed integer.<br>Unit: mm          | 1 image                 |
| Z image<br>Output format: 16-bit signed integer.<br>Unit: mm          | 1 image                 |
| Confidence image<br>Output format: 8-bit unsigned integer             | 1 image                 |
| Diagnostic data   |                         |
| Stop sequence   | String "stop" (4 bytes) |
| Ticket signature  | <CR><LF>                |

FR

### 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.

Chunk type

|        |                |  |   |
|--------|----------------|--|---|
| 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 |
| 0x0018 | PIXEL_FORMAT   | Pixel format   | 4 |

| Offset | Name        | Description   | Size [byte] |
|--------|-------------|---|-------------|
| 0x001C | TIME_STAMP  | Time stamp in microseconds  | 4           |
| 0x0020 | FRAME_COUNT | Frame counter   | 4           |
| 0x0024 | PIXEL_DATA  | The pixel data in the given type and dimension of the image. Padded to 4-byte boundary. | 4           |

Available chunk types:

|                       |     |  |
|-----------------------|-----|--|
| USERDATA              | 0   | Undefined user data with arbitrary content   |
| RADIAL_DISTANCE_IMAGE | 100 | <p>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 device calibration, excluding effects caused by multipath and multiple objects contributions (e.g. "flying pixels"). Reference point is the optical centre of the device inside the device housing.</p> <p>Invalid PMD pixels (e.g. due to saturation) have a value of zero.</p> <p>Data type: 16-bit unsigned integer (little endian)</p> <p>Unit: millimetres</p>   |
| NORM_AMPLITUDE_IMAGE  | 101 | <p>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.</p> <p>Invalid PMD pixels (e.g. due to saturation) have an amplitude value of 0.</p> <p>Data type: 16-bit unsigned integer</p>  |
| AMPLITUDE_IMAGE       | 103 | <p>Each pixel of the amplitude matrix denotes the amount of modulated light (i.e. the light from the device 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.</p> <p>Invalid PMD pixels (e.g. due to saturation) have an amplitude value of 0.</p> <p>Data type: 16-bit unsigned integer</p> |
| CARTESIAN_X_COMPONENT | 200 | <p>The X matrix denotes the X component of the Cartesian coordinate of a PMD 3D measurement. The origin of the device coordinate system is in the middle of the lens' front glass, if the extrinsic parameters are all set to 0.</p> <p>Data type: 16-bit signed integer</p> <p>Unit: millimetres</p>  |

| Constant              | Value | Description  |
|-----------------------|-------|--|
| CARTESIAN_Y_COMPONENT | 201   | The Y matrix denotes the Y component of the Cartesian coordinate of a PMD 3D measurement. The origin of the device coordinate system is in the middle of the lens' front glass, if the extrinsic parameters are all set to 0.<br><br>Data type: 16-bit signed integer<br>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 device coordinate system is in the middle of the lens' front glass, if the extrinsic parameters are all set to 0.<br><br>Data type: 16-bit signed integer<br>Unit: millimetres |
| CARTESIAN_ALL         | 203   | CARTESIAN_X_COMPONENT,<br>CARTESIAN_Y_COMPONENT,<br>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.<br><br>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)  |

Pixel format:

|              |    |  |
|--------------|----|--|
| 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:

|     |  |   |
|-----|--|---|
| 0   | <b>1</b> = pixel invalid   | <p>Pixel invalid</p> <p>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.</p>   |
| 1   | <b>1</b> = pixel saturated   | <p>Pixel is saturated</p> <p>Contributes to pixel validity: yes</p>   |
| 2   | <b>1</b> = bad A-B symmetry  | <p>A-B pixel symmetry</p> <p>The A-B symmetry value of the four phase measurements is above threshold.</p> <p>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.</p> <p>Contributes to pixel validity: yes</p> |
| 3   | <b>1</b> = amplitude below minimum amplitude threshold   | <p>Amplitude limits</p> <p>The amplitude value is below minimum amplitude threshold.</p> <p>Contributes to pixel validity: yes</p>  |
| 4+5 | <p><b>Bit 5, bit 4</b></p> <p><b>0 0</b> = unused</p> <p><b>0 1</b> = shortest exposure time (only used in 3 exposure mode)</p> <p><b>1 0</b> = middle exposure time in 3 exposure mode, short exposure in double exposure mode</p> <p><b>1 1</b> = longest exposure time (always 1 in single exposure mode)</p> | <p>Exposure time indicator</p> <p>The two bits indicate which exposure time was used in a multiple exposure measurement.</p> <p>Contributes to pixel validity: no</p>   |
| 6   | <b>1</b> = pixel is clipped  | <p>Clipping box on 3D data</p> <p>If clipping is active this bit indicates that the pixel coordinates are outside the defined volume.</p> <p>Contributes to pixel validity: yes</p>   |
| 7   | <b>1</b> = suspect/defective pixel   | <p>Suspect pixel</p> <p>This pixel has been marked as "suspect" or "defective" and values have been replaced by interpolated values from the surroundings.</p> <p>Contributes to pixel validity: no</p>   |

### 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:

|          |  |                        |
|----------|--|------------------------|
| layouter | Defines the basic data output format.<br>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:

|        |   |                      |
|--------|---|----------------------|
| type   | Defines the type of data which must be written.<br>The data might be stored in a different type (e.g. stored as integer but should be output as Float32)<br>The type "records" will need some special handling. | Type: string         |
| id     | Defines an identifier for this data element.<br>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.<br>E.g. cut an integer to less than 4 bytes.   | Type: object         |

Available values for the type property:

|         |   |
|---------|---|
| records | <p>Defines that this element represents a list of records.</p> <p>If type is set to "records", there must be an "elements" property.</p> <p>The "elements" property defines which data should be written per record.</p>                    |
| string  | <p>Data is written as string.</p> <p>Most of the time this will be used with "value" property to write fixed start, end or delimiter text.</p> <p>Text encoding should be UTF8 if there is nothing else specified in format properties.</p> |
| float32 | <p>Data is written as floating point number.</p> <p>This has a lot of formatting options (at least with "flexible" layout software)</p> <p>See following section about format properties.</p>   |
| uint32  | <p>Data is written as integer.</p> <p>This has a lot of formatting options (at least with "flexible" layout software)</p> <p>See following section about format properties.</p>   |
| int32   | <p>Data is written as integer.</p> <p>This has a lot of formatting options (at least with "flexible" layout software)</p> <p>See following section about format properties.</p>   |
| 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    | <p>Data is written as a BLOB (byte by byte as if it came from the data provider).</p> <p>(Binary Large Object)</p>  |

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

Common format properties:

|              |  |         |
|--------------|--|---------|
| 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:

|       |                         |        |
|-------|-------------------------|--------|
| order | Little, big and network | Little |
|-------|-------------------------|--------|

ASCII format properties:

|                  |  |       |
|------------------|--|-------|
| 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: <ul style="list-style-type: none"> <li>● binary (2)</li> <li>● octal (8)</li> <li>● decimal (10)</li> <li>● hexadecimal (16)</li> </ul> | 10    |

FR

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:

|                               |   |  |
|-------------------------------|---|--|
| 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<br>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<br>Measured on the illumination board  | Float32, unit: °C                            |
| x_image<br>y_image<br>z_image | Cartesian coordinates for each pixel<br>Each dimension is a separate image  | 16-bit signed integer                        |

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

|   |  |  |
|---|--|--|
| 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<br>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<br>length<br>width<br>height<br>qualityLength<br>qualityWidth<br>qualityHeight<br>xMidTop<br>yMidTop<br>zMidTop<br>yawAngle<br>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<br>float<br>float<br>float<br>float<br>float<br>float<br>float<br>float<br>float<br>float |
| numGood<br>numUnderSP1<br>numOverSP2<br>numInvalid<br>allROIsGood<br>anchorFound<br>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<br>int<br>int<br>int<br>bool<br>bool<br>bool   |

FR

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

|         |   |             |
|---------|---|-------------|
| 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)<br><ul style="list-style-type: none"> <li>• ROI_PROCESS_VALUE_VALID = 0</li> <li>• ROI_PROCESS_VALUE_REFIMAGE_SET_NOT_TEACHED = 1</li> <li>• ROI_PROCESS_VALUE_TEACHING_FAILED = 2</li> <li>• ROI_PROCESS_VALUE_REFIMAGE_INVALID = 3</li> <li>• ROI_PROCESS_VALUE_NO_VALID_PIXEL = 4</li> <li>• ROI_PROCESS_VALUE_REFIMAGE_NO_VALID_PIXEL = 5</li> <li>• ROI_PROCESS_VALUE_OVERFLOW = 6</li> <li>• ROI_PROCESS_VALUE_UNDERFILL = 7</li> </ul> | uint32      |
| quality | 0..1  | float32     |

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

|                                  |   |         |
|----------------------------------|---|---------|
| statistics_overall_count         | Allows the user to output the statistics value with the result of the frame, maps to ModelResults:<br>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:<br>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:<br>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:<br>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:<br>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:<br>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:<br>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:<br>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:<br>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:<br>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:<br>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:<br>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:<br>adv_statistics.frame_duration.max        | float32 |

## 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 t Command (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.   |
|             | !  | <ul style="list-style-type: none"> <li>• Device is busy with an evaluation</li> <li>• Device is in an invalid state for this command, e.g. configuration mode</li> <li>• Device is set to a different trigger source</li> <li>• No active application</li> </ul> |

FR

### 13.2.2 T? Command (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.   |
|             | !  | <ul style="list-style-type: none"> <li>• Device is busy with an evaluation</li> <li>• Device is in an invalid state for this command, e.g. configuration mode</li> <li>• Device is set to a different trigger source</li> <li>• No active application</li> </ul> |
| Note        | Result data can be sent via EtherNet/IP, PROFINET or TCP/IP (→ 9.3). |  |

### 13.2.3 I? Command

|             |   |  |
|-------------|---|--|
| Command     | I<image-ID>?  |  |
| Description | Request last image taken  |  |
| Type        | Request   |  |
| Reply       | <length><image data>  |  |
|             | !   | <ul style="list-style-type: none"> <li>No image available</li> <li>Wrong ID</li> </ul>   |
|             | ?   | <ul style="list-style-type: none"> <li>Invalid command length</li> </ul>   |
| Note        | <image-ID><br>2 digits for the image type<br><length><br>char string with exactly 9 digits as decimal number for the image data size in bytes<br><image data><br>image data | Valid image ID:<br>01 - amplitude image<br>02 - normalised amplitude image<br>03 - distance image<br>04 - X image (distance information)<br>05 - Y image (distance information)<br>06 - Z image (distance information)<br>07 - confidence image (status information)<br>08 - extrinsic calibration<br>09 - unit_vector_matrix_ex, ey,ez<br>10 - last result output as formatted for this connection<br>11 - all distance images: X, Y, and Z |

### 13.2.4 p Command

|             |  |   |
|-------------|--|---|
| 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<br>0: deactivates all asynchronous output<br>1: activates asynchronous result output<br>2: activates asynchronous error output<br>3: activates asynchronous error and data output<br>4: activates asynchronous notifications<br>5: activates asynchronous notifications and asynchronous result<br>6: activates asynchronous notifications and asynchronous error output<br>7: activates all outputs | On device restart the value configured within the application is essential for the output of data.<br>This command can be executed in any device state.<br>By default the error codes will not be provided by the device. |

### 13.2.5 a Command

|             |  |   |
|-------------|--|---|
| Command     | a<application number>  |   |
| Description | Activates the selected application   |   |
| Type        | Action   |   |
| Reply       | *  |   |
|             | !  | <ul style="list-style-type: none"> <li>• Application not available</li> <li>• &lt;application number&gt; contains wrong value</li> <li>• External application switching activated</li> <li>• Device is in an invalid state for this command, e.g. configuration mode</li> </ul> |
|             | ?  | Invalid command length  |
| Note        | <application number><br>2 digits for the application number as decimal value |   |

FR

### 13.2.6 A? Command

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

### 13.2.7 v Command

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



The default protocol version is „V3“.

### 13.2.8 V? Command

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

### 13.2.9 c Command

|             |  |   |
|-------------|--|---|
| Command     | c<length><configuration>   |   |
| Description | Uploads a PCIC output configuration lasting this session   |   |
| Type        | Action   |   |
| Reply       | *  |   |
|             | !  | <ul style="list-style-type: none"> <li>• Error in configuration</li> <li>• Wrong data length</li> </ul> |
|             | ?  | Invalid command length  |
| Note        | <length><br>9 digits as decimal value for the data length<br><br><configuration><br>configuration data |   |

### 13.2.10 C? Command

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

FR

### 13.2.11 S? Command

|             |  |                       |
|-------------|--|-----------------------|
| 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><br>tabulator (0x09)<br><number of results><br>Images taken since application start. 10 digits decimal value with leading 0s<br><number of positive decodings><br>Number of decodings leading to a positive result. 10 digits decimal value with leading 0s<br><number of false decodings><br>Number of decodings leading to a negative result. 10 digits decimal value with leading 0s |                       |

### 13.2.12 G? Command

|             |   |  |
|-------------|---|--|
| Command     | G?  |  |
| Description | Requests device information   |  |
| Type        | Request   |  |
| Reply       | <p>&lt;vendor&gt;&lt;t&gt;&lt;article number&gt;&lt;t&gt;<br/>                 &lt;name&gt;&lt;t&gt;&lt;location&gt;&lt;t&gt;&lt;description&gt;&lt;t&gt;&lt;ip&gt;<br/>                 &lt;subnet mask&gt;&lt;t&gt;&lt;gateway&gt;&lt;t&gt;&lt;MAC&gt;&lt;t&gt;&lt;DHCP&gt;&lt;t&gt;&lt;port number&gt;</p>   |  |
| Note        | <ul style="list-style-type: none"> <li>• &lt;vendor&gt;<br/>IFM ELECTRONIC</li> <li>• &lt;t&gt;<br/>Tabulator (0x09)</li> <li>• &lt;article number&gt;<br/>e.g. O3D300</li> <li>• &lt;name&gt;<br/>UTF8 Unicode string</li> <li>• &lt;location&gt;<br/>UTF8 Unicode string</li> <li>• &lt;description&gt;<br/>UTF8 Unicode string</li> <li>• &lt;ip&gt;<br/>IP address of the device as ASCII character sting<br/>e.g. 192.168.0.96</li> <li>• &lt;port number&gt;<br/>port number of the XML-RPC</li> <li>• &lt;subnet mask&gt;<br/>subnet mask of the device as ASCII<br/>e.g. 192.168.0.96</li> <li>• &lt;gateway&gt;<br/>gateway of the device as ASCII<br/>e.g 192.168.0.96</li> <li>• &lt;MAC&gt;<br/>MAC adress of the device as ASCII<br/>e.g. AA:AA:AA:AA:AA:AA</li> <li>• &lt;DHCP&gt;<br/>ASCII string "0" for off and "1" for on</li> </ul> |  |

### 13.2.13 H? Command

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

### 13.2.14 o Command

|             |   |   |
|-------------|---|---|
| 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        | <ul style="list-style-type: none"> <li>• &lt;IO-ID&gt;<br/>2 digits for digital output:<br/>"01" for IO1<br/>"02" for IO2<br/>"03" for IO3</li> <li>• &lt;IO-state&gt;<br/>1 digit for the state:<br/>"0" for logic state low<br/>"1" for logic state high</li> </ul> |   |

### 13.2.15 O? Command

|             |   |   |
|-------------|---|---|
| Command     | O<IO-ID>?   |   |
| Description | Requests the state of a specific ID   |   |
| Type        | Request   |   |
| Reply       | <IO-ID><IO-state>   |   |
|             | !   | <ul style="list-style-type: none"> <li>Invalid state (e.g. configuration mode)</li> <li>Wrong ID</li> </ul> |
|             | ?   | Invalid command length  |
| Note        | <ul style="list-style-type: none"> <li>&lt;IO-ID&gt;<br/>2 digits for digital output:<br/>"01" for IO1<br/>"02" for IO2<br/>"03" for IO3</li> <li>&lt;IO-state&gt;<br/>1 digit for the state:<br/>"0" for logic state low<br/>"1" for logic state high</li> </ul> | <p>The camera supports ID 1 and ID 2.<br/>The sensor supports ID 1, ID 2 and ID 3.</p>                      |

### 13.2.16 E? Command

|             |  |   |
|-------------|--|---|
| Command     | E?   |   |
| Description | Requests the current error state   |   |
| Type        | Request  |   |
| Reply       | <code>   |   |
|             | !  | Invalid state (e.g. configuration mode) |
|             | ?  | Invalid command length                  |
| Note        | <ul style="list-style-type: none"> <li>&lt;code&gt;<br/>Error code with 8 digits as a decimal value. It contains leading zeros.</li> </ul> |   |

### 13.3 Error codes

By default the error codes will not be provided by the device. The p command can activate their provision (→ 13.2.4).

|           |  |
|-----------|--|
| 10000001  | 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

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

Consuming assembly data layout

|             |              |              |
|-------------|--------------|--------------|
| Description | Command word | Command data |
|-------------|--------------|--------------|

Layout of producing assembly

|             |                            |   |                 |          |  |                           |
|-------------|----------------------------|---|-----------------|----------|--|---------------------------|
| 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

|             |   |  |
|-------------|---|--|
| Description | Error bit<br>This bit has no meaning in the consuming assembly. It is used for signaling an occurred error to the PLC | Command bits<br>Each bit represents a specific command |
|-------------|---|--|

Command word

|             |           |      |      |      |      |      |                |                   |                |                      |                      |              |              |                             |                                   |      |
|-------------|-----------|------|------|------|------|------|----------------|-------------------|----------------|----------------------|----------------------|--------------|--------------|-----------------------------|-----------------------------------|------|
| 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 | N.a. |
|-------------|-----------|------|------|------|------|------|----------------|-------------------|----------------|----------------------|----------------------|--------------|--------------|-----------------------------|-----------------------------------|------|

Synchronous / asynchronous message identifier

|             |                          |  |
|-------------|--------------------------|--|
| 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.

FR

### 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.

### 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.

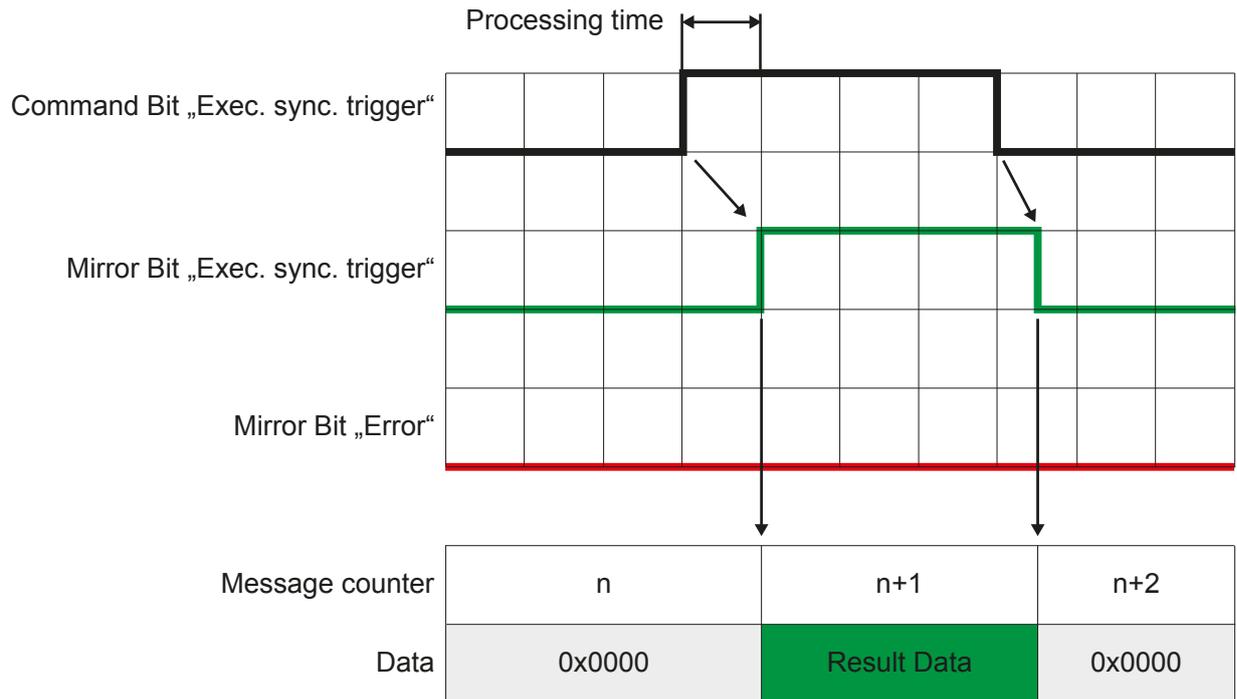
The data content of the processing assembly is set to 0.

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

### **Default endianness**

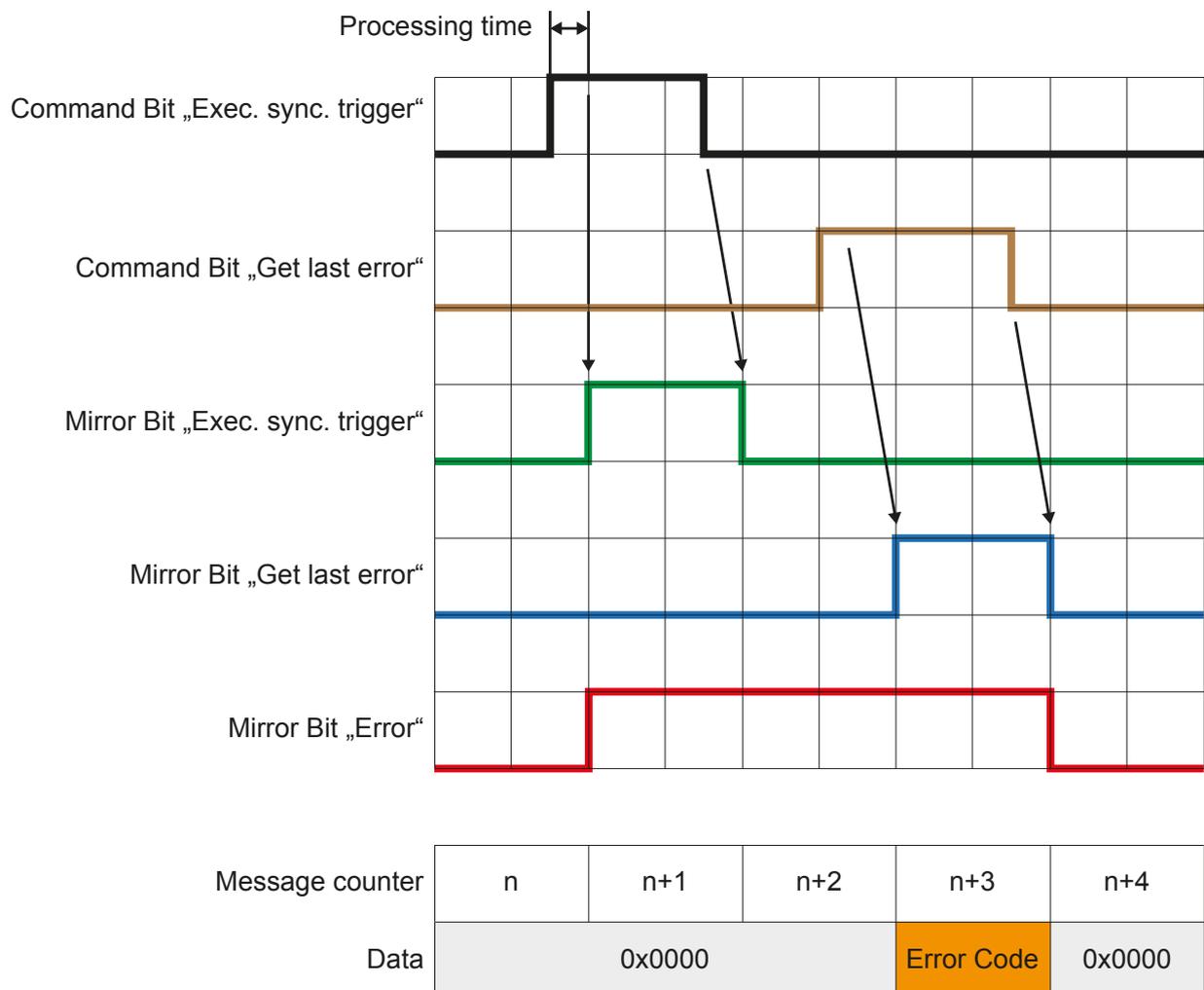
The default endianness is in little-endian format.

### 13.4.3 Signal sequence with synchronous trigger



FR

### 13.4.4 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.

|             |                            |   |                 |          |                |                    |
|-------------|----------------------------|---|-----------------|----------|----------------|--------------------|
| Description | Command word for mirroring | Synchronous / asynchronous message identifier | Message counter | Reserved | Mandatory data | Non mandatory data |
|-------------|----------------------------|---|-----------------|----------|----------------|--------------------|

#### Layout of command word

|             |   |  |
|-------------|---|--|
| Description | Error bit<br>This bit has no meaning in the consuming assembly. It is used for signaling an occurred error to the PLC | Command bits<br>Each bit represents a specific command |
|-------------|---|--|

#### Command word

|             |           |      |      |      |      |      |                |                   |                |                      |                      |              |              |                             |                                   |      |
|-------------|-----------|------|------|------|------|------|----------------|-------------------|----------------|----------------------|----------------------|--------------|--------------|-----------------------------|-----------------------------------|------|
| 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 | N.a. |
|-------------|-----------|------|------|------|------|------|----------------|-------------------|----------------|----------------------|----------------------|--------------|--------------|-----------------------------|-----------------------------------|------|

#### Synchronous / asynchronous identifier

|             |                          |  |
|-------------|--------------------------|--|
| 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.

### 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

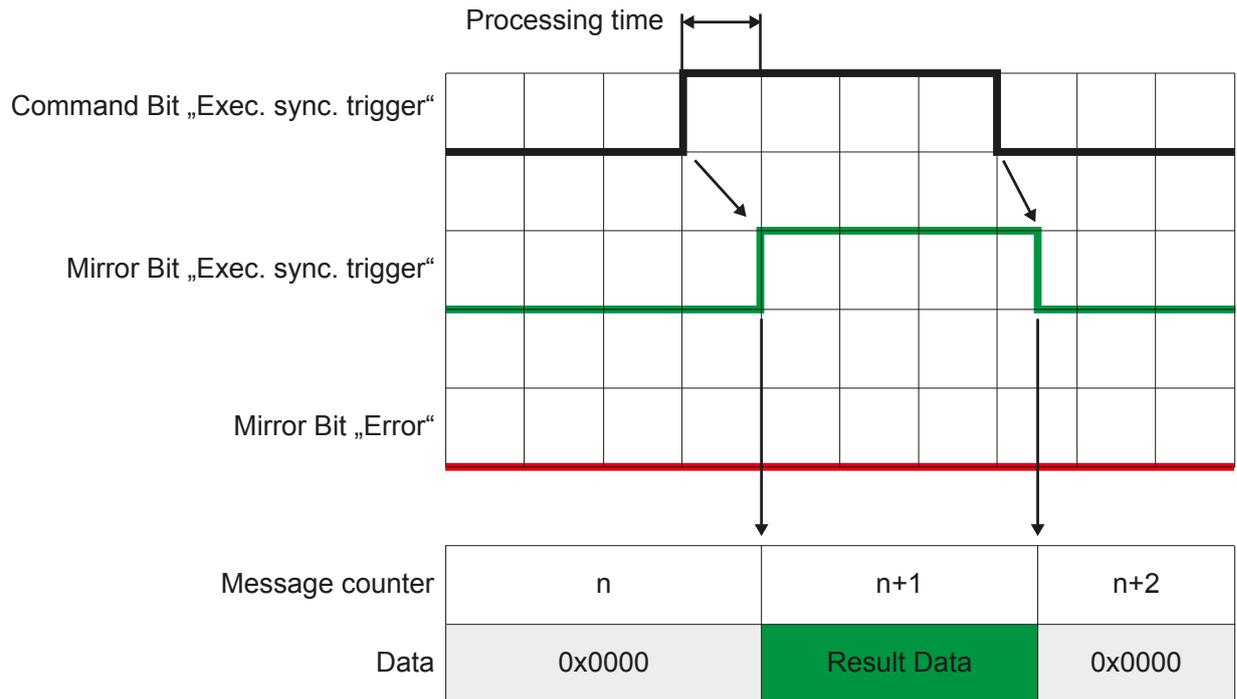
The bytes 2-5 have to be set to 0. An error is risen if bytes 2-5 are not set to 0.

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

### **Default endianness**

The default endianness is in little-endian format.

### 13.5.3 Signal sequence with synchronous trigger



FR

### 13.5.4 Signal sequence with failed trigger

