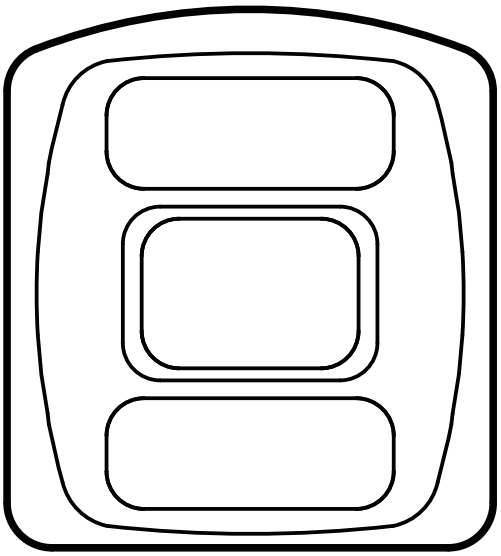
操作说明
三维传感器

O3D300
O3D302
O3D310
O3D312



CN

706398 / 06 12 / 2017

目录

CN

许可证和商标

Microsoft®、Windows®、Windows XP®、Windows Vista®、Windows 7®、Windows 8® 和 Windows 8.1® 为 Microsoft Corporation 的注册商标。

Adobe® 和 Acrobat® 为 Adobe Systems Inc. 的注册商标。

所有使用的商标和公司名称均受相应公司的版权保护。

此装置包含受特殊授权条款约束的（可能已修改的）开源软件。

如需了解版权信息和授权条款，请参阅：

www.ifm.com/int/GNU

对于受 GNU 通用公共许可证或 GNU 次要通用公共许可证约束的软件，支付复制和传送费用即可请求源代码。

1. 初步说明

此文档供专业人士使用。 此类专业人士是指经过适当培训有丰富的实践经验，能够预见和避免在操作和维护设备期间的风险及危险的人士。 本文档包含正确操作设备的相关信息。

使用产品前请阅读本文档，以了解操作条件、安装和操作。 使用设备期间，请始终妥善保管本文档。

1.1 使用的符号

► 说明

> 反应，结果

[…] 按键、按钮或指示标记

→ 交叉引用

重要说明
如不遵守，可能导致故障或干扰。

信息
补充说明

1.2 使用的警告

注意
财产损失警告。

2. 安全说明

2.1 概要

此类说明是设备不可或缺的一部分。 其中包含文字与图解，用于描述装置的正确操作方法，务请在安装或使用前阅读本说明。

请遵守操作说明。 未遵守说明、未按以下规定的使用方法操作，安装不当或操作不正确可能会严重影响操作者和机器的安全。

2.2 目标群体

此类说明适用于根据 EMC 和低电压指令的已获授权人员。 必须由具备资质的电工来安装和连接装置，以及将其投入使用。

2.3 电气连接

操作设备前，请断开设备的外部连接。

仅可为连接插脚提供技术资料中以及装置标签上所示的信号，且仅可连接经认可的 IFM 附件。

2.4 擅自改装设备

若发生故障或有相关疑问，请与制造商联系。 任何擅自改装设备的操作均可能严重影响操作员和机械的安全。 请勿擅自改装设备，我们拒绝因此引发的任何责任和保修索赔。

3. 功能和特性

O3D3xx 3D 传感器是利用光飞时间测量原理，逐点测量传感器与最近表面之间距离的光电传感器。 O3D3xx 三维传感器通过红外光源来照亮场景，并通过从表面反射的光来计算距离。

利用图像数据，通过内部图像处理生成过程值并与阈值对比。 对比值和过程值与数字输出关联。 这有助于解决以下应用：

- 完整性监控
- 液位测量
- 距离监控
- 矩形物体尺寸确认
- 矩形物体分拣

测量数据和过程值可通过以太网提供并由用户评估。 O3D3xx 三维传感器的参数设置还可通过以太网来执行。

O3D3xx 三维传感器可用于技术资料中指定的操作条件下。

就设备安全性而言，额定可用于以下环境条件：

- 室内使用
- 高达 2000 m 的高度
- 最高达 90% 的相对空气湿度，无冷凝
- 3 级污染程度

鉴于电磁干扰排放的要求，该设备适合用于工业环境。 设备并非为用于居住区内而设计的。

设备仅可用于技术资料中指定的操作条件下。

4. 所供配件

- O3D3xx 三维传感器
- 简要说明

我们的网站上提供技术资料和其他 文档（软件手册等）：
www.ifm.com

5. 附件

装置的操作需要以下附件：

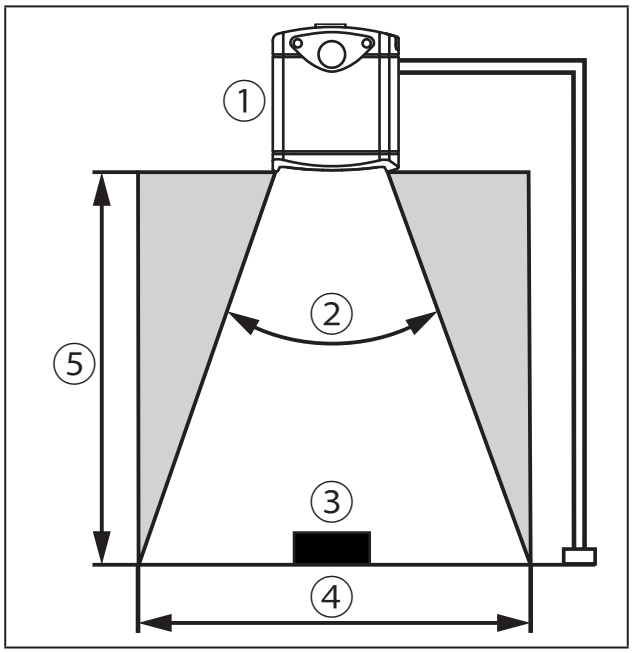
货号	说明
E11950	摄像头/传感器的供电电缆
E11898	M12 工业以太网连接电缆

我们的网站上免费提供 ifm Vision Assistant 软件： www.ifm.com

CN

6. 安装

该章节描述安装前必须遵守的要求以及传感器的安装方式。



① 设备
② 孔径角
③ 物体
④ 视场
⑤ 设备和物体之间的距离

6.1 选择安装位置

遵守关于安装位置选择的以下说明：

► 物体 ③ 必须完全处于视场 ④ 内。

> 视场的大小取决于传感器类型且会在技术资料中指示。 视场的大小还取决于传感器与物体 ⑤ 的距离：距离增大时，视场将变得更大。

► 定位物体时，请考虑公差。

► 确定传感器与物体 ⑤ 之间的距离时，请考虑传感器的测量范围。

> 测量范围将在传感器技术资料中指示。

► 选择尽可能小的传感器与物体 ⑤ 之间的距离。

> 如果距离尽可能小，则将以最大分辨率检测物体。

► 避免安装位置的任何强烈环境光和日光。

> 超过 8 klx（对应太阳光谱）的外来光强度将导致测量错误。 实际上，仅关系到 800 与 900 nm 之间的红外组件。

► 避免安装于污染严重的环境中。

> 在污染严重的环境中，尽管方向朝下 ①，传感器透镜仍会变脏。

► 避免在传感器 ① 与物体 ③ 之间有透明面板。

> 即使使用十分洁净的玻璃面板，透明面板也会反射部分光。

如果未遵守说明，则可能会发生测量错误。

6.2 其他传感器安装指南

传感器的表面温度取决于工作模式、参数选择和传感器受到的环境热暴露情况。

确保传感器符合以下要求：

对于易于触及的表面，其温度可以为比环境温度高最多 25 °C（按照 IEC 61010-2-201）。

下图包含可作为安装人员参考的一般警告限制。

在以下工作模式下，该图表有效：

- 低 [1 曝光]
- 中等 [2 曝光]
- 高 [3 曝光]

在中等和高曝光情况下，必须通过曝光时间总和来确定一般警告限制。 曝光时间将在 ifm Vision Assistant 软件中指示。

如果超过警告限制，请遵循其中一项说明：

► 降低表面温度 (→ 6.2.3)。

► 将传感器安装于既可提供防热源保护，而又能在传感器周围保持空气流通的位置或外壳中。

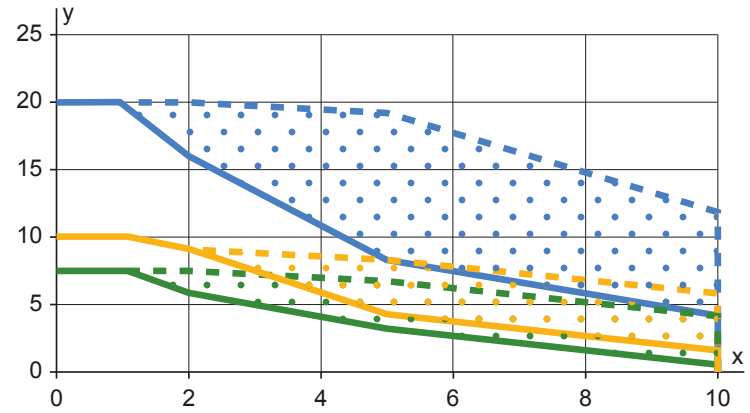
> 应避免传感器表面温度的上升。

参数“最大背景距离”在 ifm Vision Assistant 中设置。 在图表中，参数的警告限制以虚线和实线显示。

如果传感器位于其中一个带点区域，则必须降低表面温度 (→ 6.2.3)。 尽管有散热装置，如果超过警告限制，仍可额外安装接触防护装置。

如果您在正常安装时，保持在一般警告限制以下，则无需采取措施。

6.2.1 O3D300 / O3D302 的一般警告限制

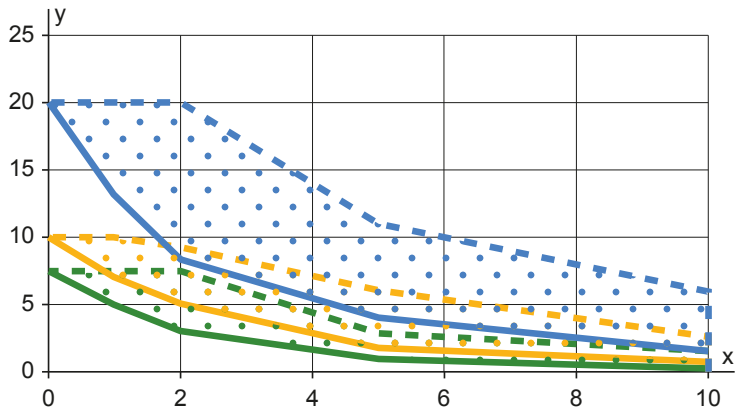


参数“最大背景距离”	
安装于带有热导体的导热金属部件上 (→ 6.2.3)	
警告限制	参数
	< 5 m
	< 30 m
	> 30 m
正常安装	
警告限制	参数
	< 5 m
	< 30 m
	> 30 m

x = 曝光时间 [ms]
y = 帧速率 [fps]

CN

6.2.2 O3D310/O3D312 的一般警告限制



参数“最大背景距离”	
安装于带有热导体的导热金属部件上 (→ 6.2.3)	
警告限制	参数
	< 5 m
	< 30 m
	> 30 m
正常安装	
警告限制	参数
	< 5 m
	< 30 m
	> 30 m

x = 曝光时间 [ms]
y = 帧速率 [fps]

6.2.3 降低表面温度

使用以下措施，可降低表面温度：

► 将传感器安装于导热金属部件上。

> 传感器与金属部件的较大表面接触可增加散热（例如，铝）。

► 将传感器安装于金属部件上时，使用热导体。

> 导热效果通过热导体来增加。 热导体可作为附件提供 (→ 6.4)。

► 减少设备周围的障碍物。 降低靠近设备的物体密度。

> 传感器周围的障碍物以及较高的安装密度可能会对对流（空气流动）产生不良影响。

► 在传感器上安装一个或两个散热器。

> 散热器可增大传感器的表面，降低表面温度。 散热器可作为附件提供 (→ 6.4)。

► 减少曝光时间、帧速率或最大背景距离。

> 使用的工作模式和参数可能会提高表面温度。

6.3 安装传感器

安装传感器时，遵守以下说明：

► 使用 2 个 M5 螺丝或安装组件来安装传感器。

> M5 螺丝的孔眼尺寸将在技术资料中指示。

> 安装组件可作为附件提供 (→ 6.4)。

► 为连接至设备的所有电缆使用应变释放装置。

安装 O3D300 和 O3D310 时，遵守以下说明：

► 以有助于利用螺丝刀操作焦点设定器的方式安装传感器。

> 对焦调整螺丝的位置将在比例图中指示 (→ 12)。

如果装置将永久用于潮湿区域，则 M12 工业以太网电缆（例如 E11898）的螺母可能会腐蚀。 使用带有高级不锈钢螺母的电缆以永久用于潮湿区域。

6.4 安装附件

视安装位置和类型而定，您可使用以下安装附件：

货号	说明
E3D301	智能摄像头安装组件
E3D302	智能摄像头冷却元件
E3D303	智能摄像头热导体
E3D304	2x 智能摄像头冷却元件

您可在以下位置找到有关附件的更多信息： www.ifm.com

CN

7. 电气连接

执行电气安装前，遵守以下说明。

注意

务必由具备资质的电工连接设备。 遵守技术资料中的电气数据要求。

防护等级为 III (PC III) 的设备。

电力供应仅可通过 PELV 电路来实现。

电力供应必须符合 UL61010-1，第 9.4 章 - 限制能源的要求。

过电流保护装置必须能在 120 秒内关闭 6.6 A 的电流。 如需了解过电流保护装置的正确额定值，请考虑传感器和配线的技术数据。

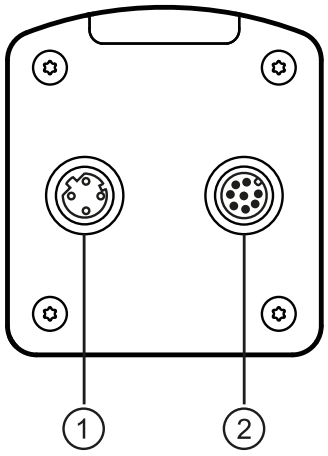
外部电路的分离必须符合 UL61010-2-201，图 102 的 要求。

对于电缆长度 > 30 m 的情况，请按照 IEC 6100-4-5 要求，使用额外的保护装置来防止浪涌电压。

连接设备前，请切断电源。

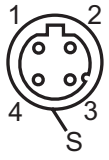
有关 cULus 的有效范围：连接至现场接线端子的缆线的最低额定温度： 70 °C。

7.1 接线



① 以太网

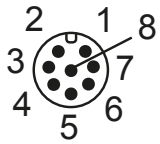
M12 插座，编码为 D，4 针



1	TD +
2	RD +
3	TD -
4	RD -
S	屏蔽

② 电源供应器

M12 连接器，编码为 A，8 针



1	U+
2	触发输入
3	GND
4	开关输出 1 - (数字或模拟)
5	开关输出 3 - 准备就绪
6	开关输出 2 - (数字)
7	开关输入 1
8	开关输入 2

用保护盖 (E73004) 覆盖未使用的以太网连接。拧紧扭矩 0.6...0.8 Nm。

开关输入和输出的运行状况可通过 ifm Vision Assistant 软件来设置。 设置 PNP 或 NPN 始终适用于所有开关输入和输出。

安装执行器和传感器时，确保设置是正确的
（例如，用于触发的光电传感器）。

开关输出也可作为脉冲输出操作，它们会在可调时间后复位其开关信号。

模拟输出提供针对 GND 的电流/电压。

7.1.1 插脚 1/3 (24 V/GND)

允许的电压范围将在传感器的技术资料中指示。

7.1.2 插脚 2（触发输入）

传感器的图像拍摄可经由触发输入通过开关信号来触发。

可使用以下触发边沿：

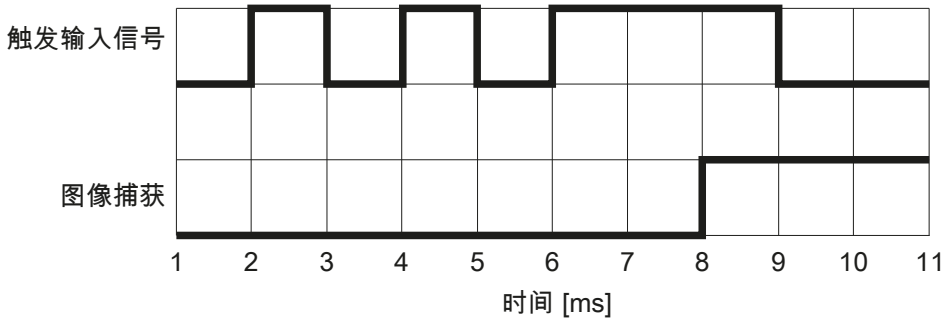
- 下降沿触发图像拍摄
- 上升沿触发图像拍摄
- 下降和上升沿好触发图像拍摄

触发传感器的其他可能性：

- 处理接口命令 (→ 13.2)
- 使用固定帧速率的连续图像拍摄

在内部为触发输入去抖动。 视电气装置而定，不一定需要触发线的去抖动功能。

内部去抖动可防止多个短脉冲触发。 脉冲必须至少长达 2 ms，方可被视为触发。



7.1.3 插脚 4 / 5 / 6（开关输出）

开关输出 1 至 3 提供不同的传感器状态。 除传感器状态之外，开关输出还可提供解决应用所需的参考值。

开关输出 1 至 3的电气规格将在技术资料中指示。

开关输出 3 将传感器状态“触发准备就绪”作为默认设置。

“已开启开关输出”意味着已出现相应的传感器状态。

视设置而定，传感器状态可能拥有任一以下值：

- “触发准备就绪”
  传感器发出信号指示可拍摄新图像。 仅在此传感器状态下，可处理触发操作。 对于连续图像拍摄，传感器状态“触发准备就绪”未输出。
- “已完成图像拍摄”
  传感器发出信号指示已完成图像拍摄。 传感器状态可用于传感器级联。
- “已完成评估”
  传感器发出信号指示已完成图像处理。 此时，已更新开关输出。 图像数据通过以太网传送。
- “错误”
  传感器发出信号指示内部错误。 可通过以太网请求提供有关错误的详细信息。

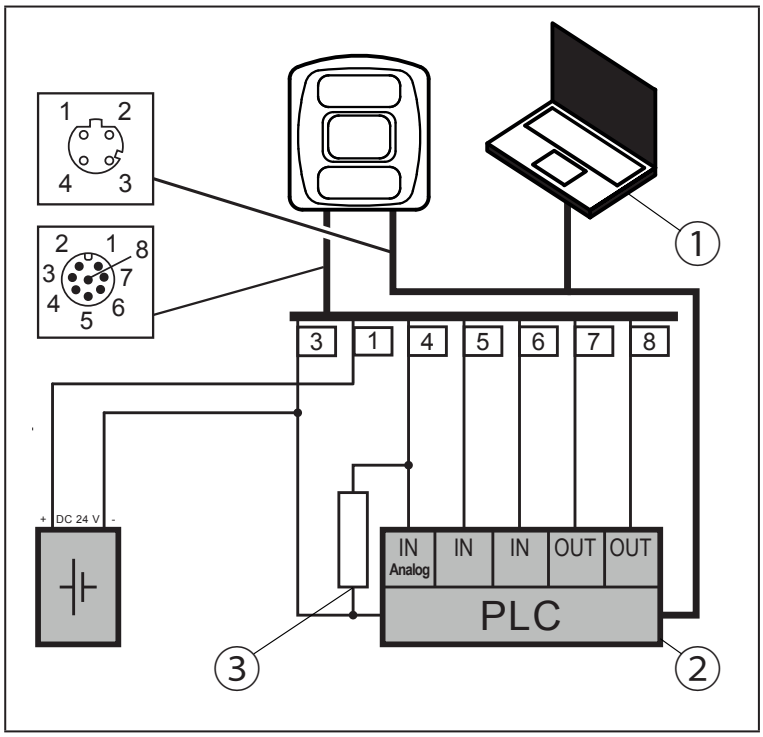
CN

7.1.4 插脚 4（模拟输出）

开关输出 1/模拟输出可用作开关输出或模拟电流输出 (4-20 mA) /模拟电压输出 (0-10 V)。

模拟电流输出的传输可靠性高于模拟电压输出。 模拟电流输出与电缆长度无关，可确保针对工业控制器有更好的信号质量。

在工业控制器中，模拟电流通过针对 GND 的负载电阻器转化为模拟电压。 负载电阻器根据技术资料中的指示加以选择。 高阻力负载电阻器比低阻力负载电阻器更加，因为设备形成的热量更低。



① 笔记本电脑（参数设置）

② 工业控制器（评估/触发）

③ 负载电阻器

使用 ifm Vision Assistant 软件，则可将过程值指定给模拟输出的初始值 (4 mA / 0 V) 和终值 (20 mA / 10 V)。

7.1.5 插脚 7 / 8（开关输入）

开关输入提供以下功能：

- 选择活动应用 (→ 7.3)

功能的不同参数设定将在软件手册中指示。

开关输出 1 和 2 的电气数据将在传感器的技术资料中指示。

7.2 配线示例

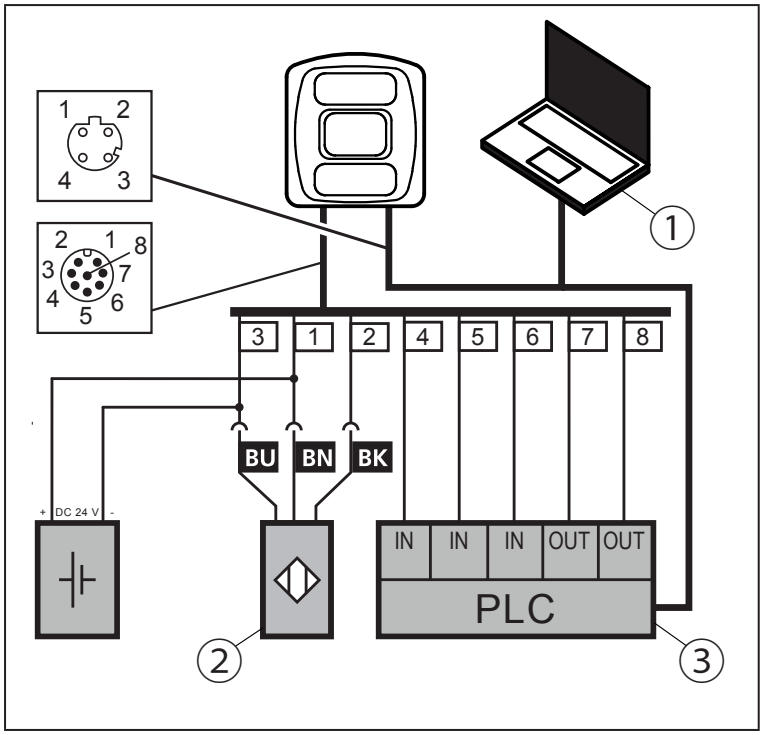
以下提供传感器的配线示例。

7.2.1 使用接近传感器触发图像拍摄

可在外部触发传感器：

- 通过以太网
- 通过连接至触发输入的接近传感器

下图显示接近传感器配线方式。



① 笔记本电脑（参数设置）

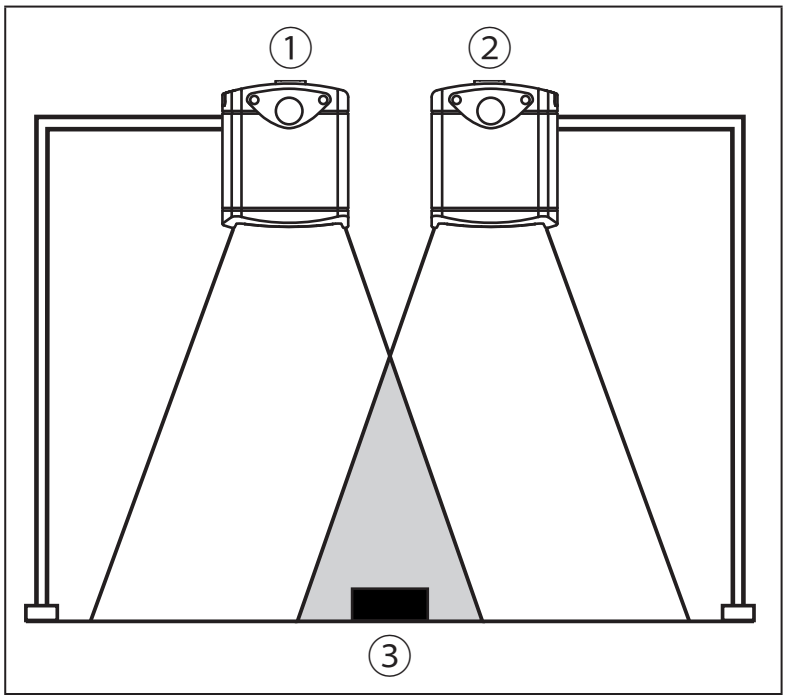
② 接近传感器

③ 工业控制器（评估/触发）

CN

7.2.2 彼此相邻地安装多个传感器

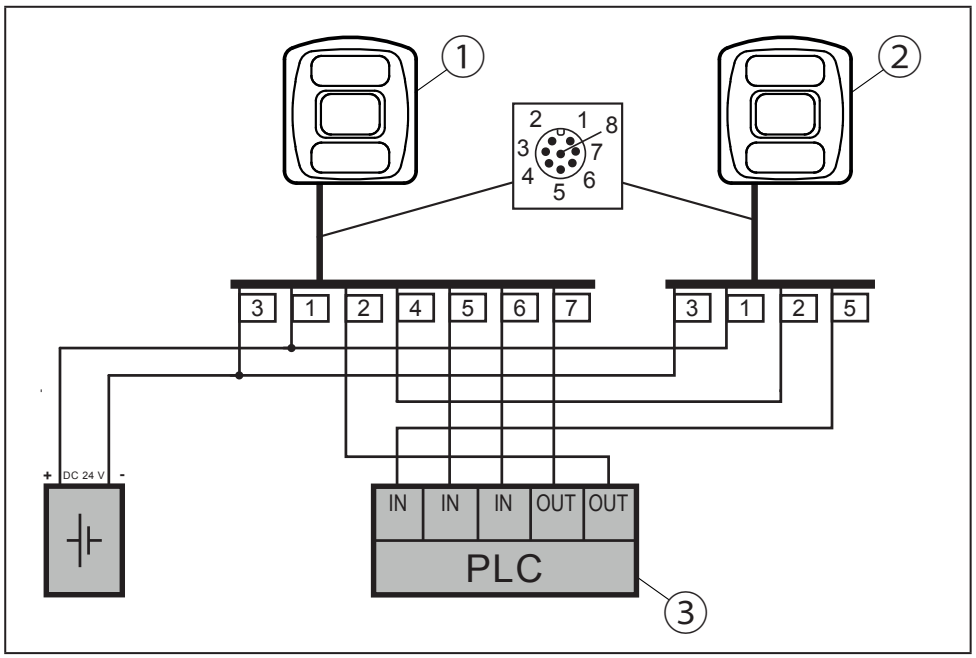
彼此相邻地安装的传感器可能会因同时曝光而导致测量错误。



① 设备
② 设备
③ 物体

可通过两种方式来避免测量错误：

- 级联传感器通过硬件触发
  级联期间，控制器触发传感器 ① 的图像拍摄（见下图）。 完成图像拍摄后，传感器 ① 自动触发传感器 ②。 同时，传感器 ① 的插脚 4 提供传感器状态“已完成图像拍摄”。 传感器 ② 向工业控制器 ③ 发送序列结束的信号。



① 设备
② 设备
③ 工业控制器（评估/触发）

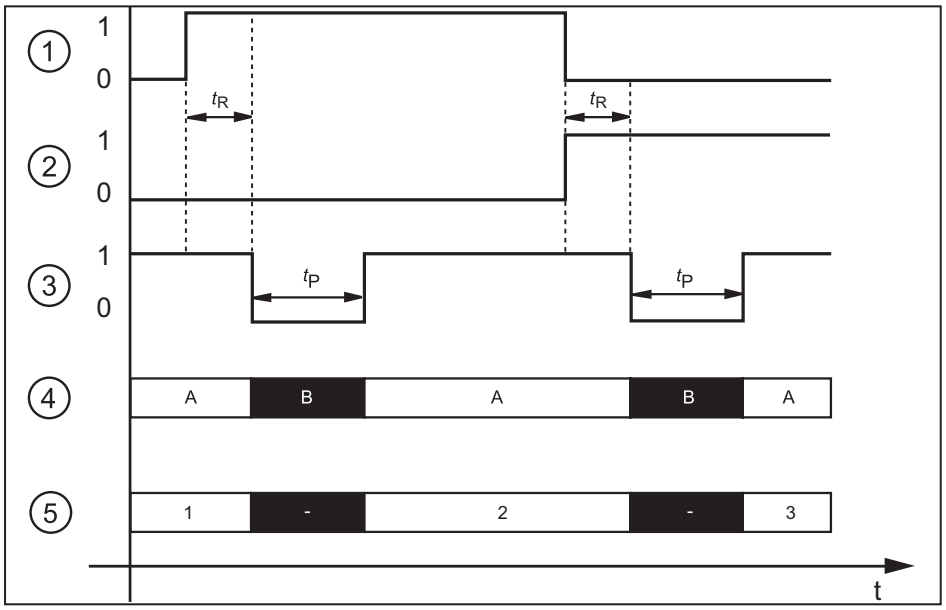
- 使用不同的频率通道
  使用 ifm Vision Assistant 软件，可为每个传感器指定其自己的频率通道。 不同的频率通道可减少测量错误的出现几率。

我们的网站上免费提供 ifm Vision Assistant 软件： www.ifm.com

7.3 应用的静态选择

传感器可储存最多 32 个不同的应用。 鉴于相应的配置，可通过两个开关输入选择前四个应用。

输入 2	输入 1	应用编号
0	0	1
0	1	2
1	0	3
1	1	4



例如： 选择应用 1 → 应用 2 → 应用 3

①	开关输入 1 = 0 → 1 → 0
②	开关输入 2 = 0 → 0 → 1
③	输出“就绪”
④	触发输入
	A： 触发已启用
	B： 触发已禁用
⑤	活动应用的 ID 号

如需选择应用，需要考虑检测时间 t_R 和触发禁用时间 t_P。

监控时间 t_R： 边缘发生变化后，在两个开关输入状态保持稳定 20 ms 之前，应用的外部选择不会开始。

触发禁用时间 t_P： 应用选择期间，触发输入禁用。 禁用时间取决于：

- 设备上的应用数量
- 应用中待启用模块的数量

上图显示 PNP 输出逻辑（出厂设置）。 NPN 输出逻辑的状况与 PNP 输出逻辑的状况相反：

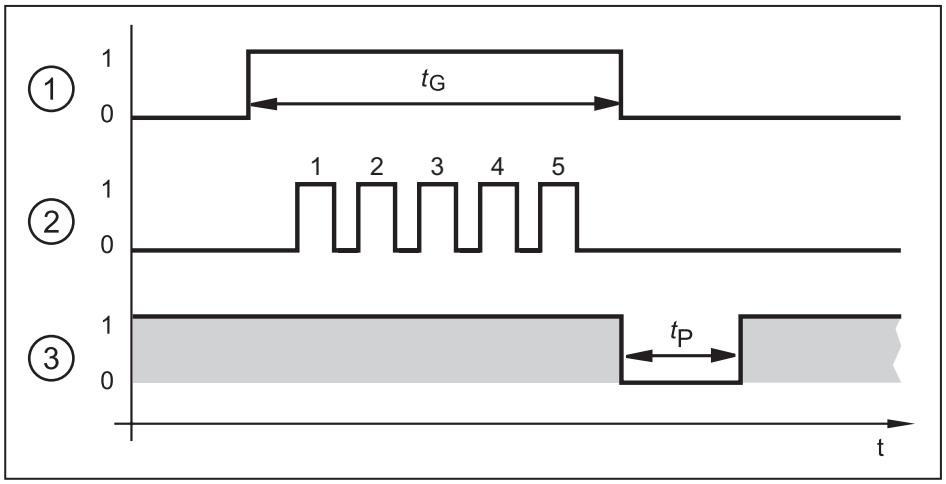
- PNP 输出逻辑： 如果是高信号 (1)，则应用电压。
- NPN 输出逻辑： 如果是低信号 (0)，则应用电压。

如需了解关于应用选择配置的更详细信息，请参阅设备的软件手册。 www.ifm.com

CN

7.4 应用的脉冲控制选择

作为静态选择的替代选择，应用选择可为脉冲控制式的。



①	门信号，开关输入 1 = 0 → 1 → 0 (t_G = 信号有效)
②	脉冲信号，开关输入 2 或触发输入 = 0 → 5 个脉冲 → 0
③	输出“就绪”

通过开关输入 1 上存在有效信号 (门信号) ，但设备会计算输入脉冲并激活相应的应用。

脉冲数量 = 应用的 ID 号

设备的开关输入 2 或触发输入可作为脉冲输入。

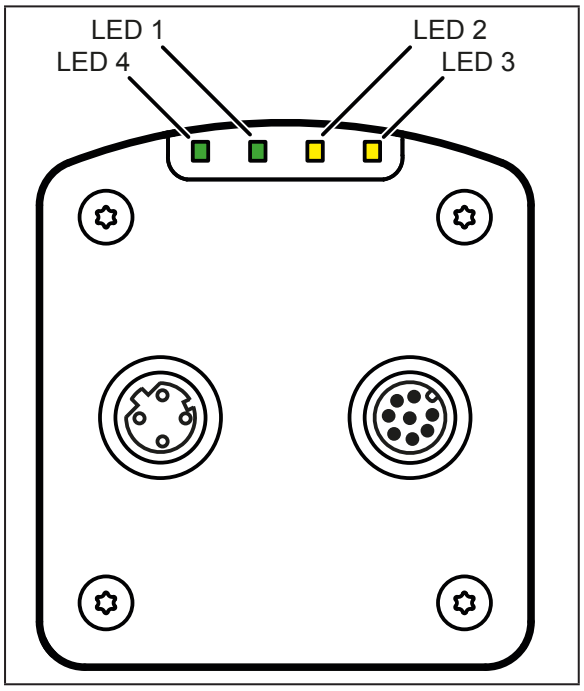
上图显示 PNP 输出逻辑 (出厂设置) 。 NPN 输出逻辑的状况与 PNP 输出逻辑的状况相反：

- PNP 输出逻辑： 如果是高信号 (1) ，则应用电压。
- NPN 输出逻辑： 如果是低信号 (0) ，则应用电压。

如需了解关于应用选择配置的更详细信息，请参阅设备的软件手册。 www.ifm.com

8. 指示器

通过 LED 指示器 1 - 4，传感器可发出信号指示当前工作状态。



LED 4 （以太网）	LED 1 （电源）	LED 2 (Out 1)	LED 3 (Out 2)	说明
	打开			传感器工作准备就绪，已提供电源电压
	以 0.5 Hz 的频率闪烁			未设置任何参数或参数设置未加载至传感器 On Off
	以 0.5 Hz 的频率闪烁 2 次			传感器处于参数设置模式 On Off
		打开		开关输出 1 已开启
		以 8 Hz 的频率闪烁		开关输出 1 短路
			打开	开关输出 2 已开启
			以 8 Hz 的频率闪烁	开关输出 2 短路
打开				以太网已连接
闪烁				以太网正在传送数据
关闭				以太网未连接
		以 8 Hz 的频率闪烁	以 8 Hz 的频率闪烁	传感器发出信号指示内部错误
		以 2 Hz 的频率闪烁	以 2 Hz 的频率闪烁	传感器发出信号指示可修正错误。 错误信息可通过以太网读取
	空载运行 ⇒			设备启动中
	空载运行 ⇐			传感器正在执行固件更新

CN

9. 设置

通电后设备进入运行状态。 15 秒后，传感器将处于评估模式，在此模式下将执行保存的应用。 指示器发出信号指示当前工作状态 (→ 8)。

传感器上可保持多达 32 个应用。 可通过各种方式激活应用：

- ifm Vision Assistant 软件
- 处理接口命令
- 开关输入 1 和 2
- 开关输入 1 和触发输入

9.1 设置设备的参数

传感器使用 ifm Vision Assistant 软件来设置（→ 请参阅软件手册）。

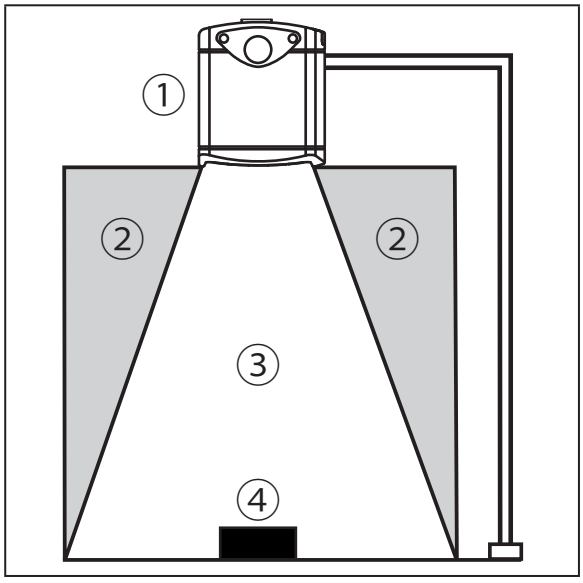
Ifm Vision Assistant 软件和有关设备测量原理的详细信息将在软件手册中说明。

我们的网站上免费提供 ifm Vision Assistant 软件：www.ifm.com

我们的网站上提供软件手册：www.ifm.com

9.2 检测物体

以下说明可促成较高物体检测率的条件。



① 设备
② 影响区域
③ 视场
④ 物体

如果符合以下条件，则可达到物体 ④ 的最佳检测效果：

- 物体位于视场 ③ 中
- 物体是最靠近传感器 ① 的可见物体
- 影响区域 ② 中没有物体（障碍物等）
- 传感器的透镜窗口无污染。

如果未符合条件，则可能会发生测量错误。

9.3 传送过程值

9.3.1 通过 EtherNet/IP 传送完整性监控的过程值

装置可通过 EtherNet/IP 现场总线将过程值传送至 PLC。 过程值在 ifm Vision Assistant 中显示为如下所示的输出字符串：

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

一次仅可有一条现场总线处于活动状态。 现场总线可调整（→ 软件手册）。

在输出字符串中，过程值由分号隔开。 输出字符串按照显示的序列传输至 PLC。

注意下方附注，以将输出字符串传送至 PLC：

- 字节 0 到 7 是输出字符串的一部分。 它们不会在 ifm Vision Assistant 中显示（请参阅上方截图）。
- 输出字符串中的分号 ";" 不会传输。
- 浮动值将先转化为 16 位整数再传送。
- 所有数值将先转化为 16 位整数再传送。

输出字符串由以下部分组成：

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

字节编号	数据	编码	过程值	单位	说明	注释
0 1	2#0000_0000 2#0010_0000	二进制 二进制	1.5		重复 命令字	• 位 1.5 表示成功的触发命令
2 3	2#0000_0000 2#0000_0000	十进制 十进制			同步/ 异步 消息识别	
4 5	30 0	十进制 十进制	30		消息计数器	• 装置收到了 30 条消息 • 每次操作（触发、消息发送等）的增量为 1。
6 7	0 0	十进制 十进制			保留	
8 9 10 11	s t a r	ASCII ASCII ASCII ASCII	star		开始字符串	
12 13	0 0	十进制 十进制	0		所有 ROI 的状态 （0 = 差，1 = 好）	显示完整性监控的状态
14 15	0 0	十进制 十进制	0		ROI ID	位置调整激活后会利用字节 14 和 15。 0 = 位置未调整 1 = 位置已调整 所有以下数据按 2 个字节变化；即第一个 ROI ID 以字节 16 和 17 开始。

CN

字节编号	数据	编码	过程值	单位	说明	注释
16 17	0 0	十进制 十进制	0		ROI 状态	ROI 状态： 0 = 好 1 = 未示教参考液位 2 = 示教失败 3 = 参考液位无效 4 = 无有效像素 5 = 参考液位不包含任何有效像素 6 = 溢出 7 = 不满
18 19	0 0	十进制 十进制	0	mm	ROI 值	
20 21	1 0	十进制 十进制	1		ROI ID	
22 23	7 0	十进制 十进制	7		ROI 状态	
24 25	-67 -1	十进制 十进制	-67	mm	ROI 值	
26 27	2 0	十进制 十进制	2		ROI ID	
28 29	6 0	十进制 十进制	6		ROI 状态	
30 31	14 0	十进制 十进制	14	mm	ROI 值	
32 33	3 0	十进制 十进制	3		ROI ID	
34 35	0 0	十进制 十进制	0		ROI 状态	
36 37	0 0	十进制 十进制	0	mm	ROI 值	
38 39 40 41	s t O p	ASCII ASCII ASCII ASCII	stop		停止字符串	

错误执行命令会导致以下状态：

- 错误位 = 1
- 显示重复命令字
- 异步消息位 = 0
- 异步消息识别 = 0
- 消息计数器增量为 1

9.3.2 通过 PROFINET 传送完整性监控的过程值

装置可通过 PROFINET 现场总线将过程值传送至 PLC。 过程值在 ifm Vision Assistant 中显示为如下所示的输出字符串：

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

一次仅可有一条现场总线处于活动状态。 现场总线可调整（→ 软件手册）。

在输出字符串中，过程值由分号隔开。 输出字符串按照显示的序列传输至 PLC。

注意下方附注，以将输出字符串传送至 PLC：

- 字节 0 到 7 是输出字符串的一部分。 它们不会在 ifm Vision Assistant 中显示（请参阅上方截图）。
- 输出字符串中的分号 ";" 不会传输。
- 浮动值将先转化为 16 位整数再传送。
- 所有数值将先转化为二进制 16 位整数再传送。

输出字符串由以下部分组成：

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

字节编号	数据	编码	过程值	单位	说明	注释
0 1	2#0010_0000 2#0000_0000	二进制 二进制	0.5		重复 命令字	• 位 0.5 表示成功的触发命令
2 3	2#0000_0000 2#0000_0000	十进制 十进制			同步/ 异步 消息识别	
4 5	0 30	十进制 十进制	30		消息计数器	• 装置收到了 30 条消息。 • 每次操作（触发、消息发送等）的增量为 1。
6 7	0 0	十进制 十进制			保留	
8 9 10 11	s t a r	ASCII ASCII ASCII ASCII	star		开始字符串	
12 13	0 0	十进制 十进制	0		所有 ROI 的状态 （0 = 差，1 = 好）	显示完整性 监控的状态
14 15	0 0	十进制 十进制	0		ROI ID	位置调整激活后会利用字节 14 和 15。 0 = 位置未调整 1 = 位置已调整 所有以下数据按 2 个字节变化；即第一个 ROI ID 以字节 16 和 17 开始。

CN

字节编号	数据	编码	过程值	单位	说明	注释
16 17	0 0	十进制 十进制	0		ROI 状态	ROI 状态： 0 = 好 1 = 未示教参考液位 2 = 示教失败 3 = 参考液位无效 4 = 无有效像素 5 = 参考液位不包含任何有效像素 6 = 溢出 7 = 不满
18 19	0 0	十进制 十进制	0	mm	ROI 值	
20 21	0 1	十进制 十进制	1		ROI ID	
22 23	0 7	十进制 十进制	7		ROI 状态	
24 25	-1 -67	十进制 十进制	-67	mm	ROI 值	
26 27	0 2	十进制 十进制	2		ROI ID	
28 29	0 6	十进制 十进制	6		ROI 状态	
30 31	0 14	十进制 十进制	14	mm	ROI 值	
32 33	0 3	十进制 十进制	3		ROI ID	
34 35	0 0	十进制 十进制	0		ROI 状态	
36 37	0 0	十进制 十进制	0	mm	ROI 值	
38 39 40 41	s t O p	ASCII ASCII ASCII ASCII	stop		停止字符串	

错误执行命令会导致以下状态：

- 错误位 = 1
- 显示重复命令字
- 异步消息位 = 0
- 异步消息识别 = 0
- 消息计数器增量为 1

9.3.3 通过 TCP/IP 传送完整性监控的过程值

装置可通过 TCP/IP 协议将过程值传送至 PLC。 过程值在 ifm Vision Assistant 中显示为如下所示的输出字符串：

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

在输出字符串中，过程值由分号隔开。 输出字符串按照显示的序列传输至 PLC。

注意下方附注，以将输出字符串传送至 PLC：

- 输出字符串中的分号 ";" 不会传输。
- 所有数值将先转化为二进制 16 位整数再传送。

输出字符串由以下部分（数据类型： ASCII）组成：

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

过程值	单位	说明
star		开始字符串
0		所有 ROI 的状态（0 = 差，1 = 好）
00		ROI ID
0		ROI 状态
+0.000	m	ROI 值
01		ROI ID
7		ROI 状态
-0.068	m	ROI 值
02		ROI ID
6		ROI 状态
+0.013	m	ROI 值
03		ROI ID
0		ROI 状态
+0.001	m	ROI 值
stop		停止字符串

ROI 状态：
0 = 好
1 = 未示教参考液位
2 = 示教失败
3 = 参考液位无效
4 = 无有效像素
5 = 参考液位不包含任何有效像素
6 = 溢出
7 = 不满

CN

9.3.4 通过 EtherNet/IP 传送物体尺寸确认的过程值

装置可通过 EtherNet/IP 现场总线将过程值传送至 PLC。 过程值在 ifm Vision Assistant 中显示为如下所示的输出字符串：

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

一次仅可有一条现场总线处于活动状态。 现场总线可调整（→ 软件手册）。

在输出字符串中，过程值由分号隔开。 输出字符串按照显示的序列传输至 PLC。

注意下方附注，以将输出字符串传送至 PLC：

- 可调整输出字符串。 可在 ifm Vision Assistant 中设置待传输过程值。
- 字节 0 到 7 是输出字符串的一部分。 它们不会在 ifm Vision Assistant 中显示（请参阅上方截图）。
- 输出字符串中的分号 ";" 不会传输。
- 浮动值将先转化为 16 位整数再传送。
- 所有数值将先转化为二进制 16 位整数再传送。

输出字符串由以下部分组成：

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

字节编号	数据	编码	过程值	单位	说明	注释
0 1	2#0000_0000 2#0010_0000	二进制 二进制	1.5		重复 命令字	• 位 1.5 表示成功的触发命令
2 3	2#0000_0000 2#0000_0000	二进制 二进制			同步/ 异步 消息识别	
4 5	2#0000_0011 2#0000_0000	二进制 二进制	3		消息计数器	• 装置收到了 3 条消息。 • 每次操作（触发、消息发送等）的增量为 1。
6 7	2#0000_0000 2#0000_0000	二进制 二进制			保留	
8 9 10 11	s t a r	ASCII ASCII ASCII ASCII	star		开始字符串	
12 13	2#0000_0001 2#0000_0000	二进制 二进制	1		结果位	0 = 未找到箱体 1 = 已找到箱体
14 15	104 0	十进制 十进制	104	mm	宽度	
16 17	88 0	十进制 十进制	88	mm	高度	
18 19	108 0	十进制 十进制	109	mm	长度	
20 21	21 0	十进制 十进制	21		x 坐标	
22 23	-11 -1	十进制 十进制	-11		y 坐标	
24 25	-124 1	十进制 十进制	389		z 坐标	

字节编号	数据	编码	过程值	单位	说明	注释
26 27	-98 0	十进制 十进制	158		旋转度数	
28 29	97 0	十进制 十进制	97		质量宽度	
30 31	93 0	十进制 十进制	94		质量高度	
32 33	97 0	十进制 十进制	97		质量长度	
34 35 36 37	s t O p	ASCII ASCII ASCII ASCII	stop		停止字符串	

错误执行命令会导致以下状态：

- 错误位 = 1
- 显示重复命令字
- 异步消息位 = 0
- 异步消息识别 = 0
- 消息计数器增量为 1

CN

9.3.5 通过 PROFINET 传送物体尺寸确认的过程值

装置可通过 PROFINET 现场总线将过程值传送至 PLC。 过程值在 ifm Vision Assistant 中显示为如下所示的输出字符串：

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

一次仅可有一条现场总线处于活动状态。 现场总线可调整（→ 软件手册）。

在输出字符串中，过程值由分号隔开。 输出字符串按照显示的序列传输至 PLC。

注意下方附注，以将输出字符串传送至 PLC：

- 可调整输出字符串。 可在 ifm Vision Assistant 中设置待传输过程值。
- 字节 0 到 7 是输出字符串的一部分。 它们不会在 ifm Vision Assistant 中显示（请参阅上方截图）。
- 输出字符串中的分号 ";" 不会传输。
- 浮动值将先转化为 16 位整数再传送。
- 所有数值将先转化为二进制 16 位整数再传送。

输出字符串由以下部分组成：

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

字节编号	数据	编码	过程值	单位	说明	注释
0 1	2#0010_0000 2#0000_0000	二进制 二进制	0.5		重复 命令字	• 位 0.5 表示成功的触发命令
2 3	2#0000_0000 2#0000_0000	二进制 二进制			同步/ 异步 消息识别	
4 5	2#0000_0000 2#0000_0011	二进制 二进制	3		消息计数器	• 装置收到了 3 条消息。 • 每次操作（触发、消息发送等）的增量为 1。
6 7	2#0000_0000 2#0000_0000	二进制 二进制			保留	
8 9 10 11	s t a r	ASCII ASCII ASCII ASCII	star		开始字符串	
12 13	2#0000_0000 2#0000_0001	二进制 二进制	1		结果位	0 = 未找到箱体 1 = 已找到箱体
14 15	0 104	十进制 十进制	104	mm	宽度	
16 17	0 88	十进制 十进制	88	mm	高度	
18 19	0 109	十进制 十进制	109	mm	长度	
20 21	0 21	十进制 十进制	21		x 坐标	
22 23	-1 -11	十进制 十进制	-11		y 坐标	
24 25	1 -124	十进制 十进制	389		z 坐标	

字节编号	数据	编码	过程值	单位	说明	注释
26 27	0 -98	十进制 十进制	158		旋转度数	
28 29	0 97	十进制 十进制	97		质量宽度	
30 31	0 94	十进制 十进制	94		质量高度	
32 33	0 97	十进制 十进制	97		质量长度	
34 35 36 37	s t O p	ASCII ASCII ASCII ASCII	stop		停止字符串	

错误执行命令会导致以下状态：

- 错误位 = 1
- 显示重复命令字
- 异步消息位 = 0
- 异步消息识别 = 0
- 消息计数器增量为 1

CN

9.3.6 通过 TCP/IP 传送物体尺寸确认的过程值

装置可通过 TCP/IP 协议将过程值传送至 PLC。 可在 ifm Vision Assistant 中选择待发送过程值。 过程值在 ifm Vision Assistant 中显示为如下所示的输出字符串：

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

在输出字符串中，过程值由分号隔开。 输出字符串按照显示的序列传输至 PLC。

注意下方附注，以将输出字符串传送至 PLC：

- 输出字符串中的分号 ";" 不会传输。
- 所有数值将先转化为二进制 16 位整数再传送。

输出字符串由以下部分（数据类型： ASCII）组成：

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

过程值	单位	说明
star		开始字符串
1		找到了物体
0.104	m	宽度
0.088	m	高度
0.109	m	长度
+0.021		x 坐标
-0.011		y 坐标
+0.389		z 坐标
158		旋转度数
097		质量宽度
094		质量高度
097		质量长度
stop		停止字符串

9.3.7 通过 EtherNet/IP 传送液位测量的过程值

装置可通过 EtherNet/IP 现场总线将过程值传送至 PLC。 过程值在 ifm Vision Assistant 中显示为如下所示的输出字符串：

一次仅可有一条现场总线处于活动状态。 现场总线可调整（→ 软件手册）。

输出字符串按照显示的序列传输至 PLC。

注意下方附注，以将输出字符串传送至 PLC：

- 字节 0 到 7 是输出字符串的一部分。 它们不会在 ifm Vision Assistant 中显示（请参阅上方截图）。
- 输出字符串中的分号 ";" 不会传输。
- 浮动值将先转化为 16 位整数再传送。
- 所有数值将先转化为二进制 16 位整数再传送。

输出字符串由以下部分组成：

0070

字节编号	数据	编码	过程值	单位	说明	注释
0 1	2#0000_0000 2#0010_0000	二进制 二进制	1.5		重复 命令字	位 1.5 表示成功的触发命令
2 3	2#0000_0000 2#0000_0000	十进制 十进制			同步/ 异步 消息识别	
4 5	30 0	十进制 十进制	30		消息计数器	• 装置收到了 30 条消息。 • 每次操作（触发、消息发送等）的增量为 1。
6 7	0 0	十进制 十进制			保留	
8 9	0 0	十进制 十进制	0		所有 ROI 的状态 （0 = 差，1 = 好）	显示液位 测量的状态
10 11	0 0	十进制 十进制	0		ROI ID	ROI 状态： 0 = 好 6 = 溢出 7 = 不满
12 13	7 0	十进制 十进制	7		ROI 状态	
14 15	0 0	十进制 十进制	0	mm	ROI 值	

错误执行命令会导致以下状态：

- 错误位 = 1
- 显示重复命令字
- 异步消息位 = 0
- 异步消息识别 = 0
- 消息计数器增量为 1

CN

9.3.8 通过 PROFINET 传送液位测量的过程值

装置可通过 PROFINET 现场总线将过程值传送至 PLC。 过程值在 ifm Vision Assistant 中显示为如下所示的输出字符串：

一次仅可有一条现场总线处于活动状态。 现场总线可调整（→ 软件手册）。

输出字符串按照显示的序列传输至 PLC。

注意下方附注，以将输出字符串传送至 PLC：

- 字节 0 到 7 是输出字符串的一部分。 它们不会在 ifm Vision Assistant 中显示（请参阅上方截图）。
- 输出字符串中的分号 ";" 不会传输。
- 浮动值将先转化为 16 位整数再传送。
- 所有数值将先转化为二进制 16 位整数再传送。

输出字符串由以下部分组成：

0070

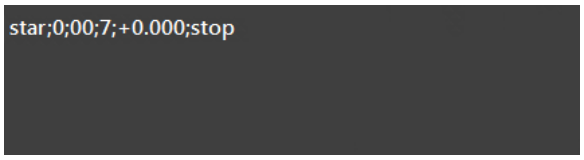
字节编号	数据	编码	过程值	单位	说明	注释
0 1	2#0010_0000 2#0000_0000	二进制 二进制	0.5		重复 命令字	位 0.5 表示成功的触发命令
2 3	2#0000_0000 2#0000_0000	十进制 十进制			同步/ 异步 消息识别	
4 5	0 30	十进制 十进制	30		消息计数器	• 装置收到了 30 条消息。 • 每次操作（触发、消息发送等）的增量为 1。
6 7	0 0	十进制 十进制			保留	
8 9	0 0	十进制 十进制	0		所有 ROI 的状态 （0 = 差，1 = 好）	显示液位 测量的状态
10 11	0 0	十进制 十进制	0		ROI ID	ROI 状态： 0 = 好 6 = 溢出 7 = 不满
12 13	0 7	十进制 十进制	7		ROI 状态	
14 15	0 0	十进制 十进制	0	mm	ROI 值	

错误执行命令会导致以下状态：

- 错误位 = 1
- 显示重复命令字
- 异步消息位 = 0
- 异步消息识别 = 0
- 消息计数器增量为 1

9.3.9 通过 TCP/IP 传送液位测量的过程值

装置可通过 TCP/IP 协议将过程值传送至 PLC。 过程值在 ifm Vision Assistant 中显示为如下所示的输出字符串：



在输出字符串中，过程值由分号隔开。 输出字符串按照显示的序列传输至 PLC。

注意下方附注，以将输出字符串传送至 PLC：

- 输出字符串中的分号 ";" 不会传输。
- 所有数值将先转化为二进制 16 位整数再传送。

输出字符串由以下部分（数据类型： ASCII）组成：

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

过程值	单位	说明
star		开始字符串
0		所有 ROI 的状态（0 = 差，1 = 好）
00		ROI ID
7		ROI 状态
+0.000	m	ROI 值
stop		停止字符串

ROI 状态：
0 = 好
6 = 溢出
7 = 不满

CN

10. 维护、修理及处理

请遵守以下说明：

- 请勿打开设备，因为它不含可由用户维护的任何组件。 仅可由制造商修理设备。
- 按照国家环保法规处理传感器。

10.1 清洁

清洁传感器前，遵守以下说明：

- 使用洁净的无绒布。
- 使用玻璃清净剂作为清洁剂。

如果未遵守说明，则透镜上的刮痕可能会导致测量错误。

10.2 更新固件

通过 ifm Vision Assistant 软件可更新传感器的固件。

固件更新会使传感器中保存的参数丢失。 更新固件前，创建参数的备份副本：

- 更新固件前，导出参数。
- 更新固件后，导入参数。

我们的网站上提供固件更新： www.ifm.com

10.3 更换设备

更换设备后，参数会丢失。 更换设备时创建参数的备份副本：

- 更换之前先导出旧设备的参数。
- 更换之后将参数导入新的设备。

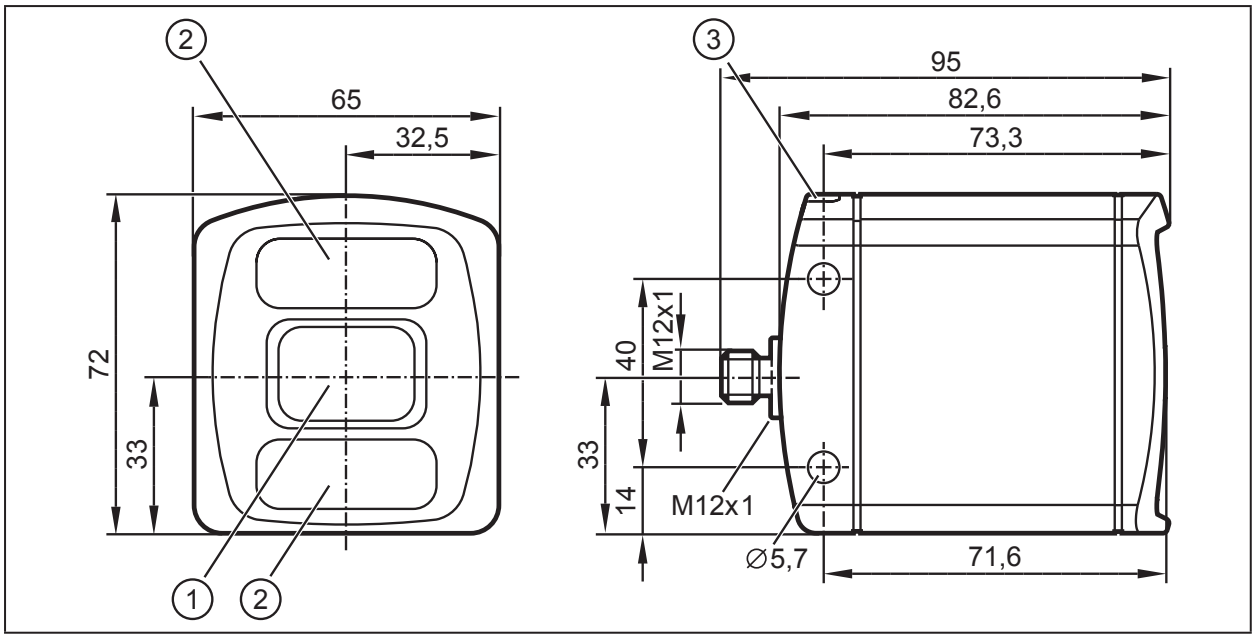
通过参数的导出和导入，可为多个设备快速提供相同的参数。

11. 认证/标准

EC 符合性声明可在以下位置找到： www.ifm.com

12. 比例图

12.1 O3D302/O3D312

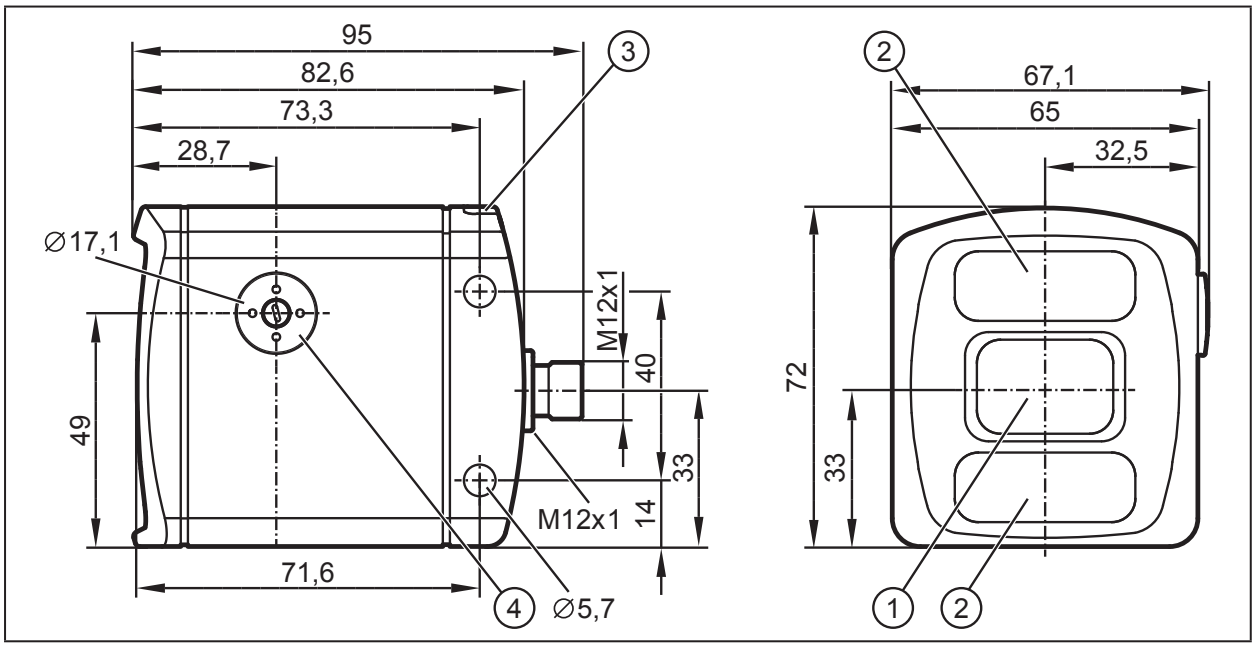


① 透镜

② 照明设备

③ 双色 LED（黄色/绿色）

12.2 O3D300/O3D310



① 透镜

② 照明设备

③ 双色 LED（黄色/绿色）

④ 焦点设定器

CN

13. Appendix

13.1 Process Interface

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

13.1.1 Sending Commands

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

Structure of the protocol:

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

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

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

They are different protocol versions available:

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

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

Ticket numbers for asynchronous messages:

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

Format of asynchronous notifications

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

Example for protocol version 3:

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

Result:

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

Explanation of the result:

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

Asynchronous message IDs

Asynchronous message ID	Description	Example	Description
000500000	Application changed	{"ID": 1034160761,"Index":1,"Name": "Pos 1","valid":true}	
000500001	Application is not valid	{"ID": 1034160761,"Index":1,"Name": "Pos 1","valid":false}	If a application exists on given index but it is invalid, the ID and Name are filled accoring to the application. If there is no application on given index, the application ID will contain 0 and the name an empty string "".

CN

13.1.2 Receiving Images

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

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

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

13.1.3 Image data

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

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

Chunk type

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

Offset	Name	Description	Size [byte]
0x0018	PIXEL_FORMAT	Pixel format	4
0x001C	TIME_STAMP	Time stamp in microseconds	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:

Constant	Value	Description
USERDATA	0	Undefined user data with arbitrary content
RADIAL_DISTANCE_IMAGE	100	Each pixel of the distance matrix denotes the ToF distance measured by the corresponding pixel or group of pixels of the imager. The distance value is corrected by the 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. Invalid PMD pixels (e.g. due to saturation) have a value of zero. Data type: 16-bit unsigned integer (little endian) Unit: millimetres
NORM_AMPLITUDE_IMAGE	101	Each pixel of the normalized amplitude image denotes the raw amplitude (see amplitude image below for further explanation), normalized to exposure time. Furthermore, vignetting effects are compensated, ie the darkening of pixels at the image border is corrected. The visual impression of this grayscale image is comparable to that of a common 2D camera. Invalid PMD pixels (e.g. due to saturation) have an amplitude value of 0. Data type: 16-bit unsigned integer
AMPLITUDE_IMAGE	103	Each pixel of the amplitude matrix denotes the amount of modulated light (i.e. the light from the 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. Invalid PMD pixels (e.g. due to saturation) have an amplitude value of 0. Data type: 16-bit unsigned integer
CARTESIAN_X_COMPONENT	200	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. Data type: 16-bit signed integer Unit: millimetres

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

Pixel format:

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

13.1.4 Additional Information for CONFIDENCE_IMAGE

Further information for the confidence image:

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

CN

13.1.5 Configuration of PCIC Output

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

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

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

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

The layout software has the following main object properties:

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

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

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

Available values for the type property:

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

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

Common format properties:

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

Binary format properties:

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

ASCII format properties:

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

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

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

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

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

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

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

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

The following element IDs are available:

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

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

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

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

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

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

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

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.
	!	• Device is busy with an evaluation • Device is in an invalid state for this command, e.g. configuration mode • Device is set to a different trigger source • No active application

13.2.2 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.
	!	• Device is busy with an evaluation • Device is in an invalid state for this command, e.g. configuration mode • Device is set to a different trigger source • No active application
Note	Result data can be sent via EtherNet/IP, PROFINET or TCP/IP (→ 9.3).	

13.2.3 I? Command

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

13.2.4 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 0: deactivates all asynchronous output 1: activates asynchronous result output 2: activates asynchronous error output 3: activates asynchronous error and data output 4: activates asynchronous notifications 5: activates asynchronous notifications and asynchronous result 6: activates asynchronous notifications and asynchronous error output 7: activates all outputs	On device restart the value configured within the application is essential for the output of data. This command can be executed in any device state. By default the error codes will not be provided by the device.

CN

13.2.5 a Command

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

13.2.6 A? Command

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

13.2.7 v Command

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

The default protocol version is „V3“.

13.2.8 V? Command

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

13.2.9 c Command

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

CN

13.2.10 C? Command

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

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

13.2.12 G? Command

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

CN

13.2.13 H? Command

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

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	• <IO-ID> 2 digits for digital output: "01" for IO1 "02" for IO2 "03" for IO3 • <IO-state> 1 digit for the state: "0" for logic state low "1" for logic state high	

13.2.15 O? Command

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

13.2.16 E? Command

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

CN

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).

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

13.4 EtherNet/IP

13.4.1 Data structures for consuming and producing assemblies

Assemblies

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

Consuming assembly data layout

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

Layout of producing assembly

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

Layout of command word

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

Command word

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

Synchronous / asynchronous message identifier

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

Data to send exceeds processing assembly data section size

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

CN

13.4.2 Functionality of the Ethernet/IP application

The chapter describes the initialization of assembly buffers.

On initialization all buffers are set to 0.

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

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

Multiple state changes

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

Reset of command bit state by PLC

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

Blocking of asynchronous messages

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

Client disconnect

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

General reply to an implemented command

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

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

Reply to an implemented command - reply contains specific data

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

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

Reply to an implemented command with error in data section

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

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

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

Reply to an implemented command that leads to an error

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

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

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

Reply to a not implemented command

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

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

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

Reset of error bit

The error bit will be resetted to 0, if

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

Functionality of asynchronous message bit

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

Bits for asynchronous message identifier

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

The ticket number for asynchronous results is 0.

The ticket number for asynchronous error codes is 1.

Message counter

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

Get last error

This command is used to reset the error bit.

Get connection ID

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

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

Get statistics

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

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

All values are 32 bit unsigned integers.

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.

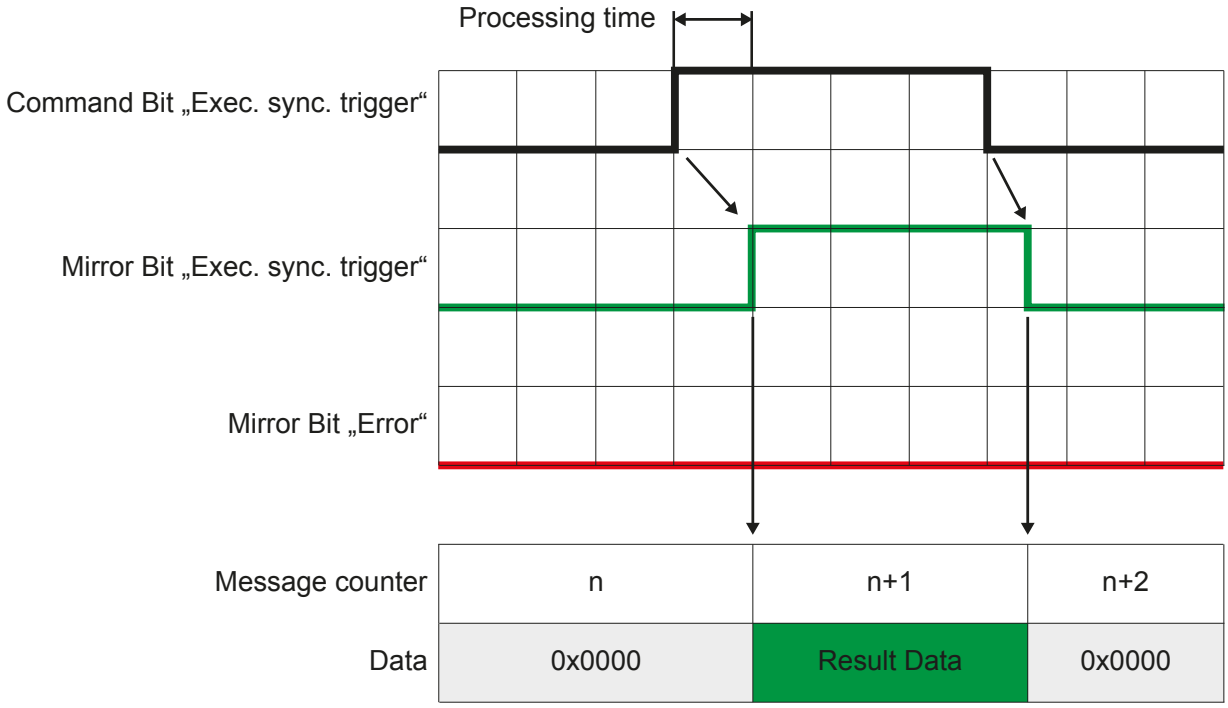
The data content of the processing assembly is set to 0.

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

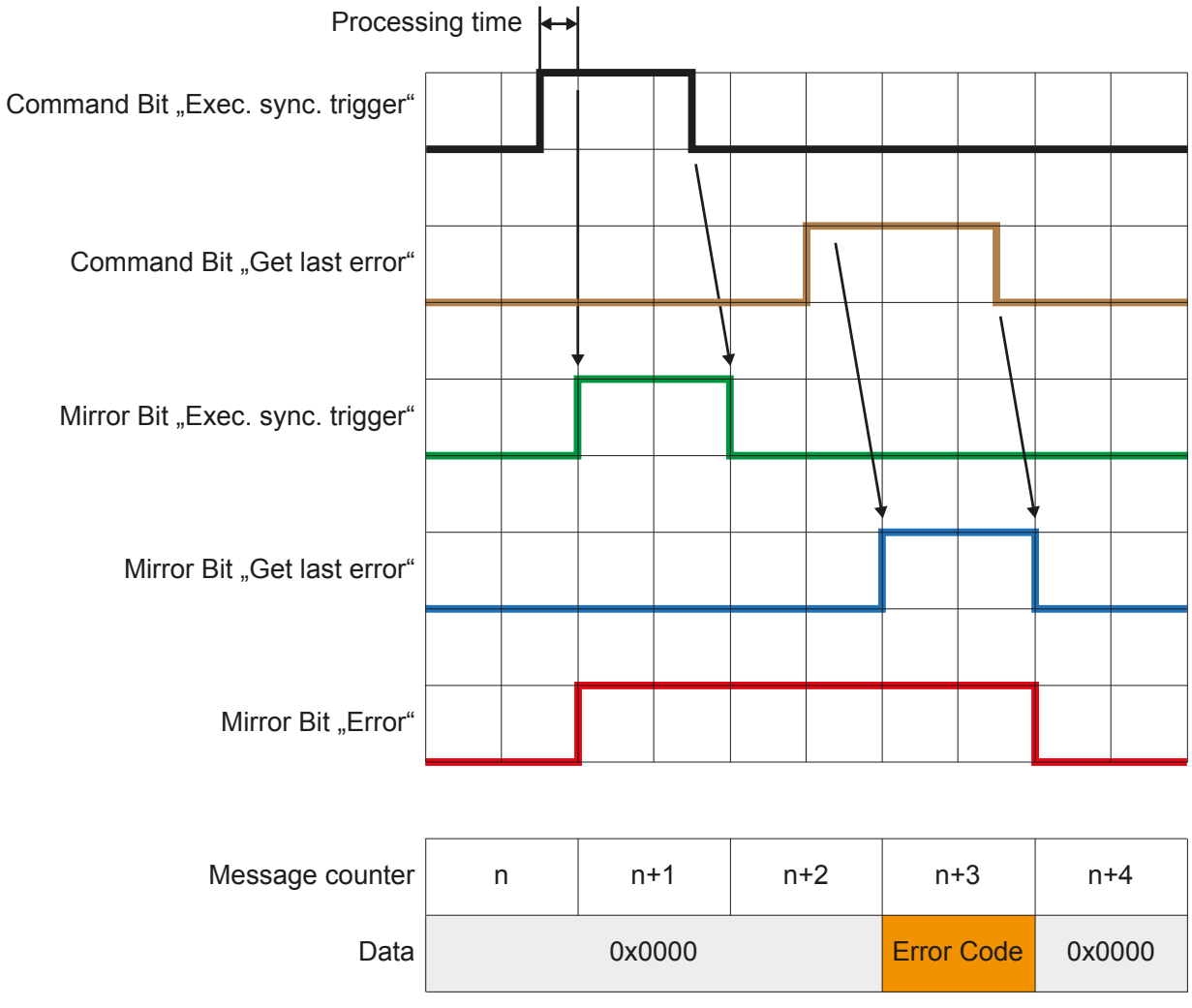
Default endianness

The default endianness is in little-endian format.

13.4.3 Signal sequence with synchronous trigger



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.

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

Layout of command word

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

Command word

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

Synchronous / asynchronous identifier

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

13.5.2 Functionality of PROFINET IO application

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

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

Number of supported PROFINET connections

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

CN

Initialisation of input and output buffers

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

Command execution triggering

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

Handling of multiple command bits

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

Command execution completion

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

Blocking of asynchronous messages

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

Client disconnect

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

General reply to an implemented command

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

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

Reply to an implemented command - reply contains specific data

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

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

Reply to an implemented command with error in data section

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

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

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

Reply to an implemented command that leads to an error

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

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

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

Reply to a not implemented command

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

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

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

Reset of error bit

The error bit will be resetted to 0, if

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

Queuing of error codes

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

Functionality of asynchronous message bit

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

Bits for asynchronous message identifier

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

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

Message counter

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

Get last error

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

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

Get connection ID

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

Get statistics

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

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

All values are 32 bit unsigned integers.

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

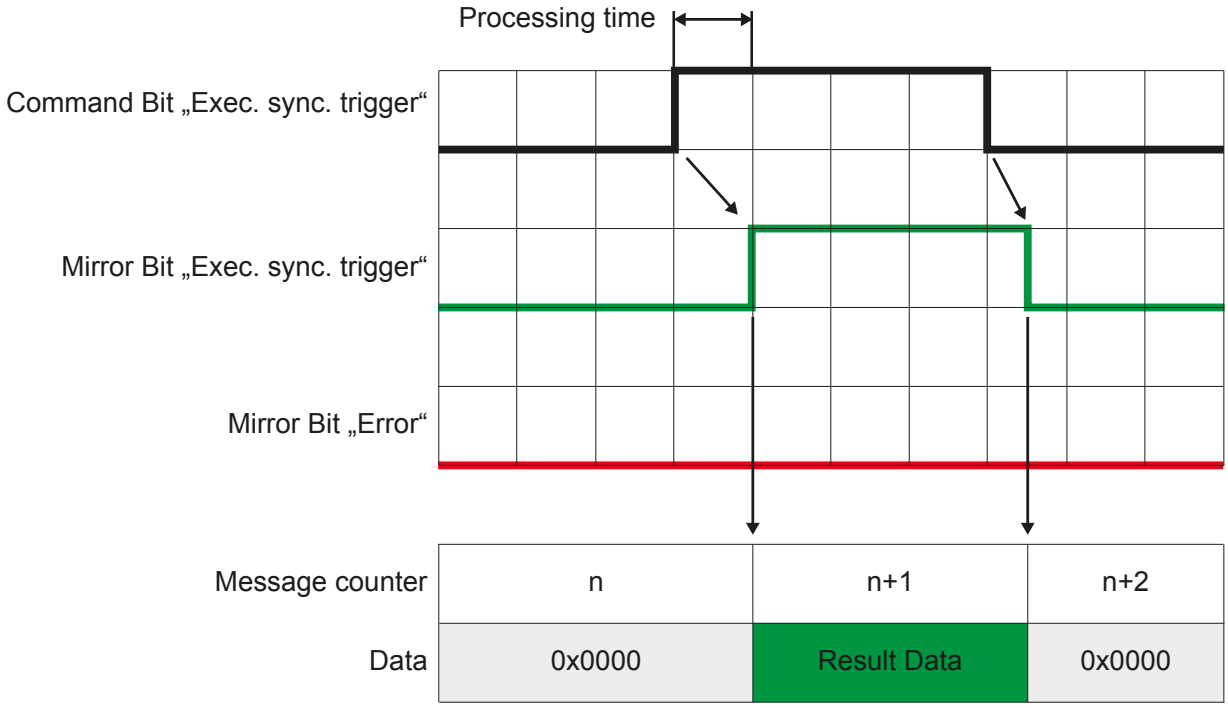
The bytes 2-5 have to be set to 0. An error is risen if bytes 2-5 are not set to 0.

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



13.5.4 Signal sequence with failed trigger