



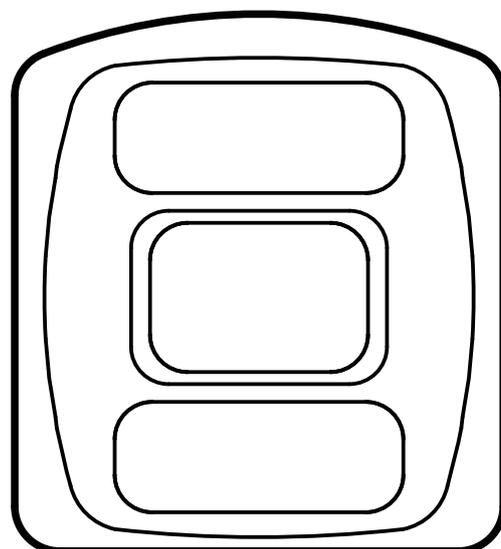
Инструкция по эксплуатации 3D-датчик

O3D300

O3D302

O3D310

O3D312



RU

Содержание

1. Введение	4
1.1 Используемые символы	4
1.2 Используемые предупреждения	4
2. Инструкции по безопасной эксплуатации	4
2.1 Основное	4
2.2 Целевая группа	4
2.3 Электрическое подключение	4
2.4 Вмешательство в устройство прибора	5
3. Функции и ключевые характеристики	5
4. Комплект поставки	5
5. Принадлежности	5
6. Установка	6
6.1 Выбор места установки	6
6.2 Дополнительные указания по установке датчика	7
6.2.1 Стандартные пределы предупреждения O3D300 / O3D302	7
6.2.2 Стандартные пределы предупреждения O3D310 / O3D312	8
6.2.3 Снижение температуры поверхности	8
6.3 Установка датчика	9
6.4 Монтажные принадлежности	9
7. Электрическое подключение	10
7.1 Схема подключения	10
7.1.1 Контакт 1 / 3 (24 В / GND)	11
7.1.2 Контакт 2 (триггерный вход)	11
7.1.3 Контакт 4 / 5 / 6 (коммутационные выходы)	11
7.1.4 Контакт 4 (аналоговый выход)	12
7.1.5 Контакт 7 / 8 (коммутационные входы)	12
7.2 Примеры подключения	13
7.2.1 Запуск захвата изображения с помощью датчика приближения	13
7.2.2 Установка датчиков рядом друг с другом	14
7.3 Статический выбор программы	15
7.4 Выбор программы, управляемый импульсом	16
8. Индикаторы	17
9. Ввод в эксплуатацию	18
9.1 Настройки параметров датчика	18
9.2 Обнаружение объекта	18
9.3 Передача рабочих значений	19
9.3.1 Передача рабочих значений параметров мониторинга комплектности с помощью EtherNet/IP	19
9.3.2 Передача рабочих значений параметров мониторинга комплектности с помощью PROFINET	21
9.3.3 Передача рабочих значений мониторинга комплектности через TCP/IP	23
9.3.4 Передача рабочих значений, определяющих размеры объекта через EtherNet/IP	24
9.3.5 Передача рабочих значений, определяющих размеры объекта через PROFINET	26
9.3.6 Передача рабочих значений, определяющих размеры объекта через TCP/IP	28
9.3.7 Передача рабочих значений мониторинга уровня через EtherNet/IP	29
9.3.8 Передача рабочих значений измерения уровня через PROFINET	30

9.3.9	Передача рабочих значений мониторинга уровня через TCP/IP	31
10.	Техническое обслуживание, ремонт и утилизация	32
10.1	Очистка	32
10.2	Обновление прошивки	32
10.3	Замена прибора	32
11.	Разрешения/стандарты	32
12.	Типовые размеры	33
12.1	O3D302 / O3D312	33
12.2	O3D300 / O3D310	33
13.	Appendix	34
13.1	Process Interface	34
13.1.1	Sending Commands	34
13.1.2	Receiving Images	36
13.1.3	Image data	36
13.1.4	Additional Information for CONFIDENCE_IMAGE	39
13.1.5	Configuration of PCIC Output	40
13.2	Process Interface Command Reference	46
13.2.1	t Command (Asynchronous Trigger)	46
13.2.2	T? Command (Synchronous Trigger)	46
13.2.3	l? Command	47
13.2.4	p Command	47
13.2.5	a Command	48
13.2.6	A? Command	48
13.2.7	v Command	49
13.2.8	V? Command	49
13.2.9	c Command	49
13.2.10	C? Command	50
13.2.11	S? Command	50
13.2.12	G? Command	51
13.2.13	H? Command	52
13.2.14	o Command	52
13.2.15	O? Command	53
13.2.16	E? Command	53
13.3	Error codes	54
13.4	EtherNet/IP	55
13.4.1	Data structures for consuming and producing assemblies	55
13.4.2	Functionality of the Ethernet/IP application	56
13.4.3	Signal sequence with synchronous trigger	60
13.4.4	Signal sequence with failed trigger	60
13.5	PROFINET IO	61
13.5.1	Data structures for output and input frame	61
13.5.2	Functionality of PROFINET IO application	61
13.5.3	Signal sequence with synchronous trigger	66
13.5.4	Signal sequence with failed trigger	66

Лицензии и торговые марки

Microsoft®, Windows®, Windows XP®, Windows Vista®, Windows 7®, Windows 8® и Windows 8.1® являются зарегистрированными торговыми марками корпорации Microsoft.

Adobe® и Acrobat® являются зарегистрированными торговыми марками корпорации Adobe Systems Inc.

Все торговые марки и названия компании охраняются авторским правом.

Прибор имеет программное обеспечение с открытым исходным кодом, распространяемое на основе лицензии.

Всю необходимую информацию относительно авторских прав и лицензий смотрите на:

www.ifm.com/int/GNU

Для программного обеспечения Вам может понадобиться исходный код в соответствии со стандартной общественной лицензией ограниченного применения на свободное программное обеспечение (GNU Lesser General Public License или GNU Library General Public License).

1. Введение

Инструкция предназначена для специалистов. Специалистами считаются квалифицированные работники, которые прошли специальное обучение, и их опыт позволяет им оценивать риски и предотвращать возможные опасности, которые могут возникнуть во время эксплуатации или технического обслуживания прибора. Инструкция содержит информацию о правильной эксплуатации прибора.

Перед эксплуатацией прибора внимательно прочтите инструкцию по установке, ознакомьтесь с правилами и условиями по эксплуатации прибора, а также его функционированием. Храните данную инструкцию на протяжении всего срока эксплуатации прибора, чтобы при необходимости обращаться к ней впоследствии.

1.1 Используемые символы

► Инструкции по применению

> Реакция, результат

[...] Маркировка органов управления, кнопок или обозначение индикации

→ Ссылка на соответствующий раздел



Важное примечание

Несоблюдение этих рекомендаций может привести к неправильному функционированию устройства или созданию помех.



Информация

Дополнительное разъяснение

1.2 Используемые предупреждения

ПРИМЕЧАНИЕ

Предупреждение о нанесении материального ущерба.

2. Инструкции по безопасной эксплуатации

2.1 Основное

Данные инструкции по применению являются неотъемлемой частью прибора. Перед установкой прибора необходимо внимательно ознакомиться с инструкциями, соблюдение которых обеспечивает правильное функционирование прибора.

Строго соблюдайте инструкции по эксплуатации. Несоблюдение инструкций по установке и эксплуатации прибора или его использование не по назначению может привести к неисправности оборудования или серьезным травмам персонала.

2.2 Целевая группа

Инструкция по эксплуатации предназначена для использования только уполномоченными лицами по стандарту EMC и директивам в отношении приборов низкого напряжения. Прибор должен устанавливать, подключать и вводить в эксплуатацию квалифицированный электрик.

2.3 Электрическое подключение

Перед выполнением любых работ по установке или обслуживанию отключите прибор от внешнего источника питания.

Соединительные клеммы могут питаться только от сигналов, указанных в технической спецификации и на табличке прибора и подключаться к ним могут только рекомендованные принадлежности от ifm.

2.4 Вмешательство в устройство прибора

В случае неисправности прибора или возникновения каких либо сомнений, пожалуйста, обратитесь к производителю. Несанкционированное вмешательство в прибор может серьезно повлиять на безопасность персонала и машин. Любое вмешательство в заводскую конфигурацию прибора приводит к аннулированию гарантийных обязательств.

3. Функции и ключевые характеристики

3D-датчик O3D3xx это фотоэлектрический датчик, точка за точкой измеряющий расстояние между датчиком и ближайшей поверхностью с помощью принципа измерения времени пролета луча. 3D-датчик O3D3xx освещает сцену с помощью внутреннего источника инфракрасного света и рассчитывает расстояние с помощью света, отраженного от поверхности.

Из данных изображения генерируются рабочие значения с помощью внутренней обработки изображения и сравниваются с пороговыми значениями. Сравнимые и рабочие значения передаются в цифровые выходы. Это позволяет решать следующие виды применения:

- Мониторинг целостности групповой упаковки
- Измерение уровня
- Определение расстояния
- Определение размеров прямоугольных объектов
- Сортировка прямоугольных объектов

Измеренные данные и рабочие значения могут обеспечиваться через Ethernet и обрабатываться пользователем. Параметрирование 3D-датчика O3D3xx также производится через Ethernet.

3D-датчик O3D3xx должен использоваться только в условиях эксплуатации, указанных в инструкции по применению.

Прибор рассчитан на использование при следующих условиях эксплуатации:

- Внутри помещений
- Высота до 2000 м
- Относительная влажность воздуха максимум до 90 %, без конденсации
- Степень загрязнения 3

Из-за требований по излучению электромагнитных помех прибор предназначен для использования в промышленной среде. Прибор не предназначен для применения в домашних условиях.



Прибор должен использоваться только в условиях эксплуатации, указанных в инструкции по применению.

4. Комплект поставки

- 3D-датчик O3D3xx
- Краткая инструкция



Технические характеристики и другая документация (руководство по программному обеспечению, и т. д.) доступны на нашем сайте: www.ifm.com

5. Принадлежности

Для функционирования прибора необходимы следующие принадлежности:

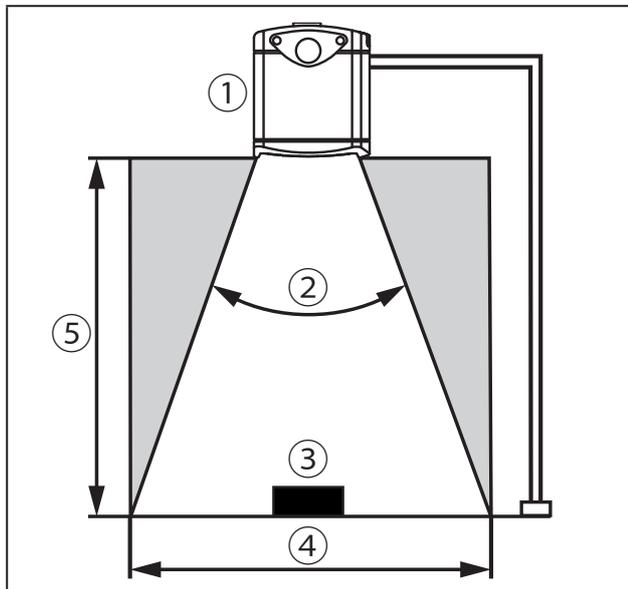
Артикульный номер	Описание
E11950	Кабель питания для камеры/датчика
E11898	M12 промышленный соединительный кабель Ethernet



ПО ifm Vision Assistant доступно бесплатно на нашей веб-странице: www.ifm.com

6. Установка

Глава описывает, что необходимо принять во внимание до установки и как устанавливать датчик.



- ① Устройство
- ② Угол обзора
- ③ Объект
- ④ Поле зрения
- ⑤ Расстояние между прибором и объектом

6.1 Выбор места установки

Соблюдайте следующие инструкции для выбора места установки:

- ▶ Объект ③ должен находиться полностью в поле зрения ④.
- > Размер поля зрения зависит от типа датчика и указан в технической спецификации. Размер поля зрения также зависит от расстояния датчика от объекта ⑤: С возрастающим расстоянием поле зрения становится больше.
- ▶ Примите во внимание отклонение при позиционировании объекта.
- ▶ При определении расстояния между датчиком и объектом ⑤ примите во внимание диапазон измерения датчика.
- > Диапазон измерения указан в технической спецификации датчика.
- ▶ Выберите наименьшее возможное расстояние между датчиком и объектом ⑤.
- > При наименьшем расстоянии объект обнаруживается с максимальным разрешением.
- ▶ Избегайте попадания сильного рассеянного или солнечного света на место установки.
- > Уровень внешнего освещения более 8 кЛюкс (с солнечным спектром) вызывает ошибки в измерении. На самом деле, проблемы вызывает только инфракрасный компонент между 800 и 900 нм.
- ▶ Избегайте установки в сильно загрязненных местах.
- > В сильно загрязненных местах линза датчика загрязнится несмотря на ориентацию вниз ①.
- ▶ Избегайте прозрачных панелей между датчиком ① и объектом ③.
- > Прозрачные панели отражают часть света даже если используется очень чистая стеклянная панель.



При несоблюдении инструкций могут возникать ошибки в измерении.

6.2 Дополнительные указания по установке датчика

Температура поверхности датчика зависит от режима работы, выбора параметров и термического воздействия датчика на окружающую среду.



Убедитесь, что датчик соответствует следующим требованиям:

Температура поверхности для легко доступных поверхностей может быть максимум на 25 °C выше, чем температура окружающей среды (по IEC 61010-2-201).

Следующие графики содержат стандартные пределы предупреждения в качестве рекомендации для установщика.



Графики действительны для следующих режимов работы:

- Низкий [1 экспозиция]
- Средний [2 экспозиции]
- Высокий [3 экспозиции]

В случае режимов средней или высокой экспозиции типичные пределы предупреждения должны быть определены по сумме времени экспозиции. Время экспозиции указано в ПО ifm Vision Assistant.

Если пределы предупреждения превышены, следуйте одной из инструкций:

- ▶ Снижение температуры поверхности (→ 6.2.3).
- ▶ Устанавливайте датчик на место или корпус, который обеспечивает защиту от источника тепла, но удерживает циркуляцию воздуха вокруг датчика.
- > Необходимо предотвратить повышение температуры поверхности датчика.

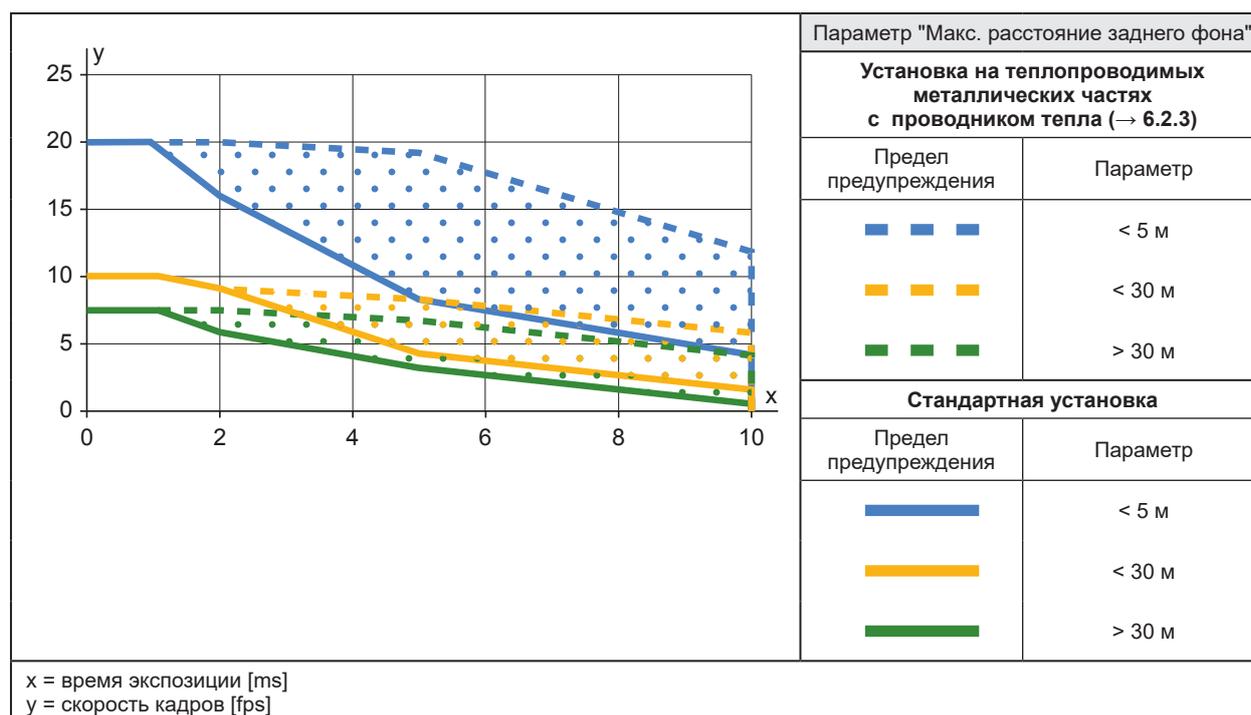


Параметр "Максимальное расстояние заднего фона" настраивается в ifm Vision Assistant. Пределы предупреждения параметров обозначены на графике пунктирными и сплошными линиями.

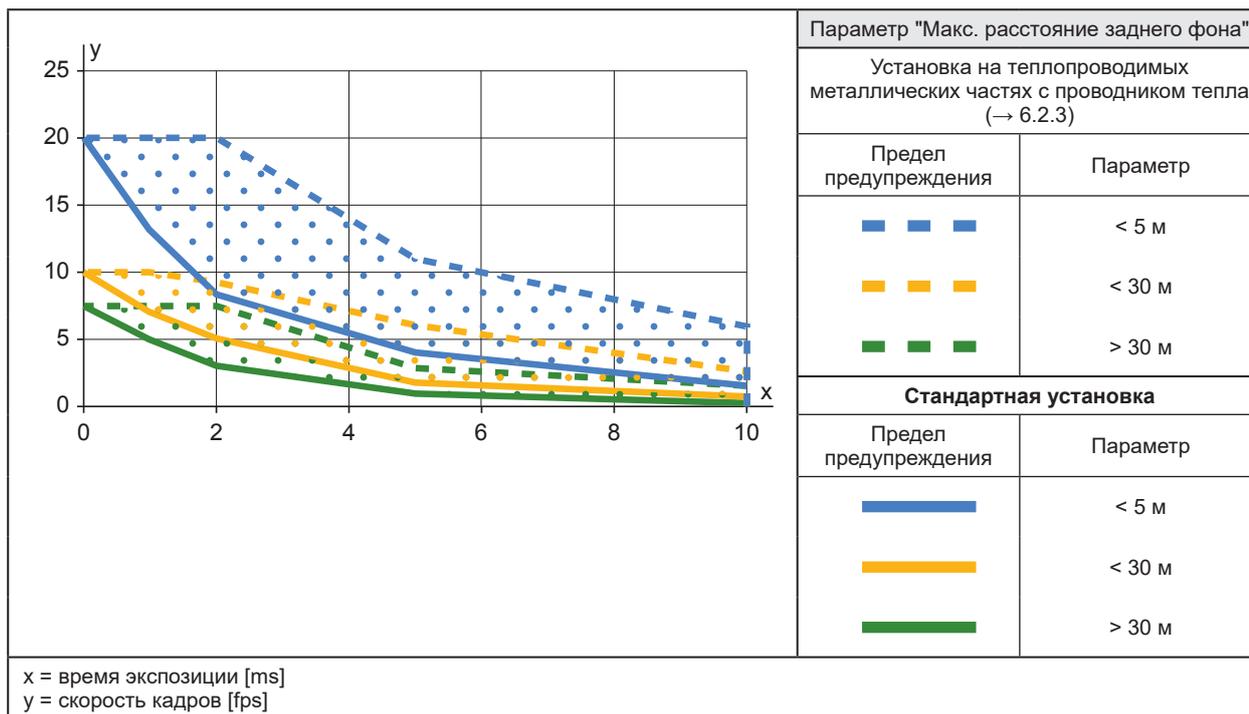
Если датчик находится в зоне с точками, температуру поверхности необходимо снизить (→ 6.2.3). Если предел предупреждения превышен даже несмотря на теплоотводную установку, возможно дополнительно установить защиту контактов.

Если при стандартной установке вы остаетесь ниже типичных пределов предупреждения, дополнительные требования не требуются.

6.2.1 Стандартные пределы предупреждения O3D300 / O3D302



6.2.2 Стандартные пределы предупреждения O3D310 / O3D312



6.2.3 Снижение температуры поверхности

Температуру можно снизить принятием следующих мер:

- ▶ Установите датчик на теплопроводимые металлические части.
- > Большая площадь контакта датчика с металлическими частями увеличивает теплоотдачу (напр. алюминий).
- ▶ При установке датчика на металлические части, используйте проводник тепла.
- > Эффект теплопроводимости повышается с помощью проводника тепла. Кондуктор предлагается в качестве принадлежностей (→ 6.4).
- ▶ Уменьшите помехи вокруг датчика. Уменьшите количество объектов, установленных рядом с датчиком.
- > Объекты вокруг датчика и высокая плотность установки может негативно отразиться на конвекции (движение воздуха).
- ▶ Установите один или два радиатора на датчик.
- > Радиаторы увеличивают поверхность датчика, снижая температуру поверхности. Радиаторы доступны в качестве аксессуаров (→ 6.4).
- ▶ Сократите время экспозиции, частоту кадров или макс. расстояние заднего фона.
- > Используемый режим работы и параметры могут повысить температуру поверхности.

6.3 Установка датчика

При установке датчика следуйте инструкциям:

- ▶ Присоедините датчик с помощью 2 винтов М5 или монтажного набора.
- > Размеры отверстия для винтов М5 указаны в технической спецификации.
- > Монтажный набор предлагается в качестве принадлежностей (→ 6.4).
- ▶ Используйте защиту от натяжения для всех кабелей, подключенных к датчику.

При установке O3D300 и O3D310 соблюдайте следующие инструкции:

- ▶ Установите датчик так, чтобы кнопка настройки фокуса была доступна для отвертки.
- > Положение кнопки настройки фокуса обозначено на чертеже (→ 12).



Если прибор постоянно используется во влажной среде, гайка М12 кабеля промышленного интерфейса Ethernet (напр. E11898) может заржаветь. Для постоянного использования во влажных средах используйте гайку из высококачественной нержавеющей стали.

6.4 Монтажные принадлежности

В зависимости от места и типа установки, можно использовать следующие монтажные принадлежности:

Артикульный номер	Описание
E3D301	Монтажный набор Smart Camera
E3D302	Охлаждающий элемент Smart Camera
E3D303	Конвектор Smart Camera
E3D304	2 охлаждающих элемента Smart Camera



Более подробная информация о принадлежностях находится на: www.ifm.com

7. Электрическое подключение

Соблюдайте следующие инструкции до электрической установки.

ПРИМЕЧАНИЕ

К работам по установке и вводу в эксплуатацию допускаются только квалифицированные специалисты - электрики. Соблюдайте электрические данные, указанные в технической спецификации.

Класс защиты прибора III (РС III).

Электрическое питание должно подаваться только через защищённые сверхнизковольтные цепи.

Электрическое питание должно соответствовать UL61010-1, глава 9.4 - Ограничение энергии:

Устройство для защиты от сверхтоков должно отключить ток 6.6 А в течение 120 с. Для правильной оценки устройства защиты от сверхтоков примите во внимание технические данные датчика и проводку.

Отделение внешних цепей должно соответствовать UL61010-2-201, рис. 102.

Для длины кабеля > 30 м, используйте дополнительную защиту от импульсного перенапряжения в соответствии с IEC 6100-4-5:

Отключите питание перед подключением устройства.



Для соблюдения требований сертификата cULus:

Предельная температура кабеля, подключаемого к полевым клеммам: 70 °С.

7.1 Схема подключения

	① Ethernet Разъем M12, D-кодировка, 4 полюса															
	<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>Экран</td></tr> </table>	1	TD +	2	RD +	3	TD -	4	RD -	S	Экран					
1	TD +															
2	RD +															
3	TD -															
4	RD -															
S	Экран															
	② Блок питания Разъем M12, A-кодировка, 8 полюсов															
	<table border="0"> <tr><td>1</td><td>U+</td></tr> <tr><td>2</td><td>Вход триггера</td></tr> <tr><td>3</td><td>GND</td></tr> <tr><td>4</td><td>Коммутационный выход 1 - (цифровой или аналоговый)</td></tr> <tr><td>5</td><td>Коммутационный выход 3 - готов</td></tr> <tr><td>6</td><td>Коммутационный выход 2 - (цифровой)</td></tr> <tr><td>7</td><td>Коммутационный вход 1</td></tr> <tr><td>8</td><td>Коммутационный вход 2</td></tr> </table>	1	U+	2	Вход триггера	3	GND	4	Коммутационный выход 1 - (цифровой или аналоговый)	5	Коммутационный выход 3 - готов	6	Коммутационный выход 2 - (цифровой)	7	Коммутационный вход 1	8
1	U+															
2	Вход триггера															
3	GND															
4	Коммутационный выход 1 - (цифровой или аналоговый)															
5	Коммутационный выход 3 - готов															
6	Коммутационный выход 2 - (цифровой)															
7	Коммутационный вход 1															
8	Коммутационный вход 2															



Неиспользуемое соединение Ethernet закройте колпачком (E73004). Момент затяжки 0.6...0.8Нм.



Поведение коммутационных входов и выходов можно настроить с помощью ПО ifm Vision Assistant. Настройка PNP или NPN всегда действительна для всех коммутационных входов и выходов.

При установке актуаторов и датчиков убедитесь, что настройка сделана правильно (напр. фотоэлектрические датчики для срабатывания).

Коммутационные выходы могут работать также как импульсные выходы, которые сбрасывают их коммутационный сигнал после истечения установленного времени.

Аналоговый выход обеспечивает ток / напряжение по отношению к GND (заземление).

7.1.1 Контакт 1 / 3 (24 В / GND)

Допустимый диапазон напряжения указан в технической спецификации датчика.

7.1.2 Контакт 2 (триггерный вход)

Захват изображения датчика можно вызвать с помощью коммутационного сигнала через триггерный выход.

Можно использовать следующие фронты триггера:

- Спадающий фронт запускает захват изображения
- Нарастающий фронт запускает захват изображения
- Спадающий и нарастающий фронт запускают захват изображения



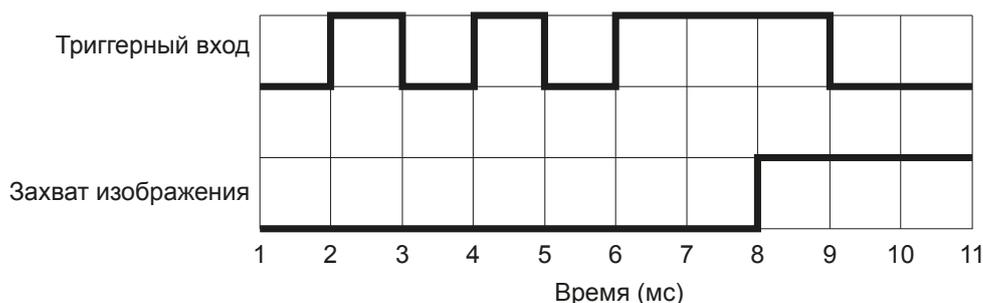
Другие возможности для запуска датчика:

- Команда интерфейса (→ 13.2)
- Постоянный захват изображения с фиксированной частотой кадров



Вход триггера внутренне защищен от дребезга контактов. В зависимости от электрической установки, защита провода триггера не обязательна.

Внутренняя защита предотвращает срабатывание от коротких импульсов. Длина импульса должна быть не менее 2 мс, чтобы он был распознан как триггер.



7.1.3 Контакт 4 / 5 / 6 (коммутационные выходы)

Коммутационные выходы от 1 до 3 обеспечивают различные состояния датчика. Кроме состояния датчика коммутационные выходы могут обеспечить исходное значение для решения задачи применения.

Электрическая характеристика коммутационных выходов от 1 до 3 указана в технической спецификации.

Коммутационный выход 3 обеспечивает состояние датчика "Готов для триггера".



"Switching output switched" (Коммутационный выход переключен) означает, что настало соответствующее состояние датчика.

В зависимости от настройки, состояние датчика может иметь одно из следующих значений:

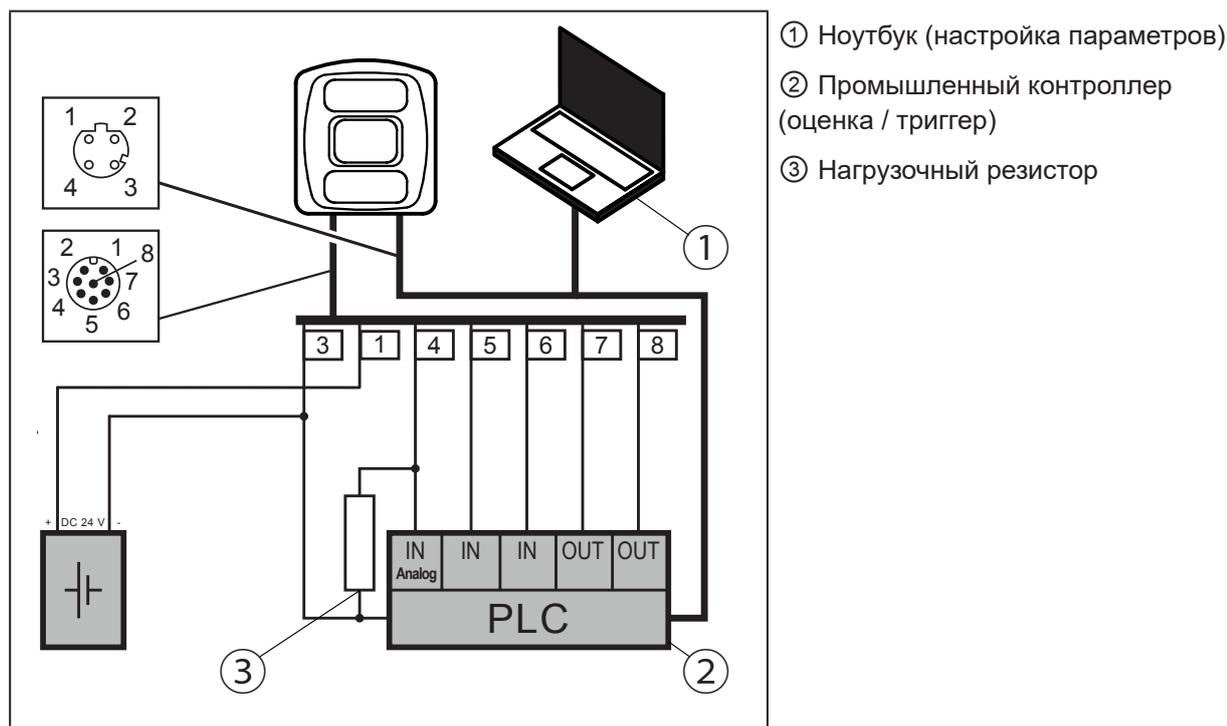
- "Ready for trigger" (Готов для триггера)
Датчик сигнализирует, что может быть захвачено новое изображение. Только в данном состоянии датчика обрабатываются операции запуска. Для текущего захвата изображения состояние "Ready for trigger" не выдается.
- "Image capture finished" (Захват изображения закончен)
Датчик сигнализирует, что захват изображения закончен. Данное состояние датчика можно использовать для каскадного подключения датчиков.
- "Evaluation finished" (Оценка закончена)
Датчик сигнализирует, что обработка изображения закончена. В этот момент коммутационные выходы готовы к обновлению. Данные изображения передаются через Ethernet.
- "Error" (Ошибка)
Датчик сигнализирует внутреннюю ошибку. Подробную информацию об ошибках можно запросить через Ethernet.

7.1.4 Контакт 4 (аналоговый выход)

Коммутационный выход 1 / аналоговый выход можно использовать как коммутационный выход или аналоговый выход по току (4-20 мА) / аналоговый выход по напряжению (0-10 В).

Аналоговый токовый выход предлагает более надежную передачу, чем аналоговый выход по напряжению. Аналоговый выход по току не зависит от длины кабеля и обеспечивает лучшее качество сигнала по отношению к промышленному контроллеру.

В промышленном контроллере аналоговый ток преобразуется в аналоговое напряжение через нагрузочный резистор. Нагрузочный резистор выбирается в соответствии с указаниями в технической спецификации. Высокоомные резисторы предпочтительны для резисторов с малой нагрузкой из-за более низкого тепловыделения в устройстве.



С помощью ПО ifm Vision Assistant возможно присоединить одно рабочее значение к начальному значению (4 мА / 0 В) и конечному значению (20 мА / 10 В) аналогового выхода.

7.1.5 Контакт 7 / 8 (коммутационные входы)

Коммутационные входы обеспечивают следующие функции:

- Выберите активное приложение (→ 7.3)



Различные настройки параметров функций указаны в руководстве по программированию.



Электрические данные коммутационных входов 1 и 2 указаны в технической спецификации.

7.2 Примеры подключения

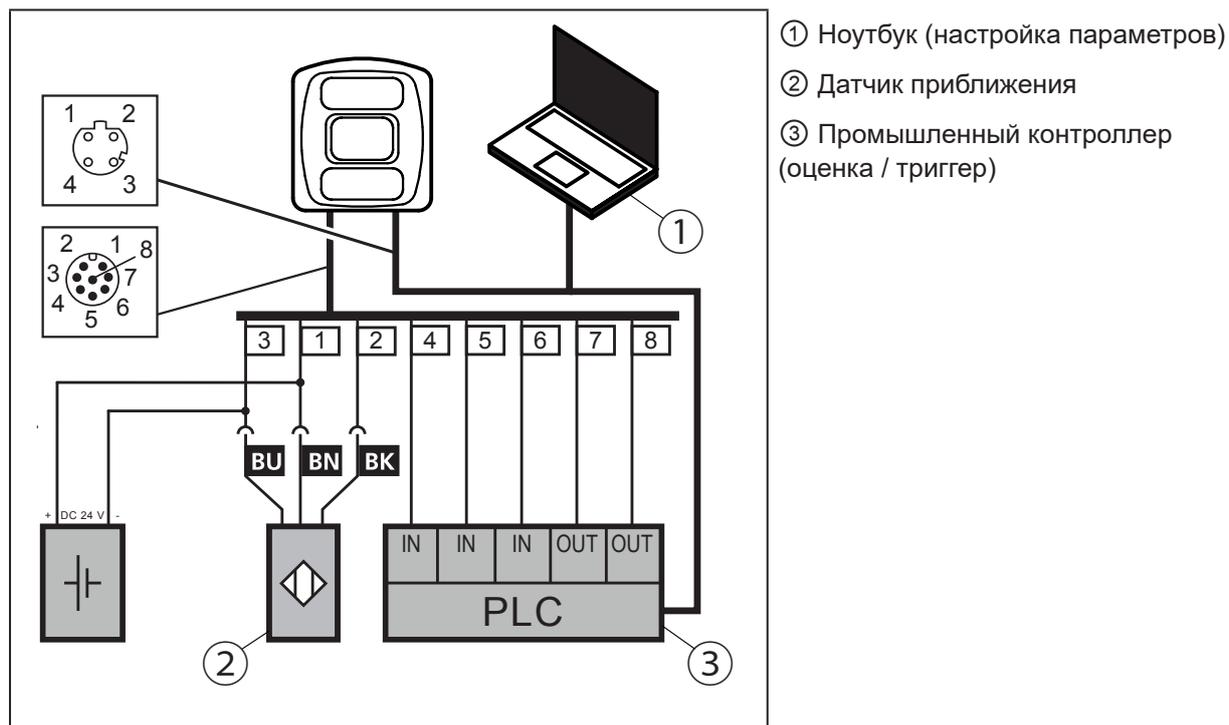
Примеры подключения датчика указаны ниже.

7.2.1 Запуск захвата изображения с помощью датчика приближения

Датчик можно запустить снаружи:

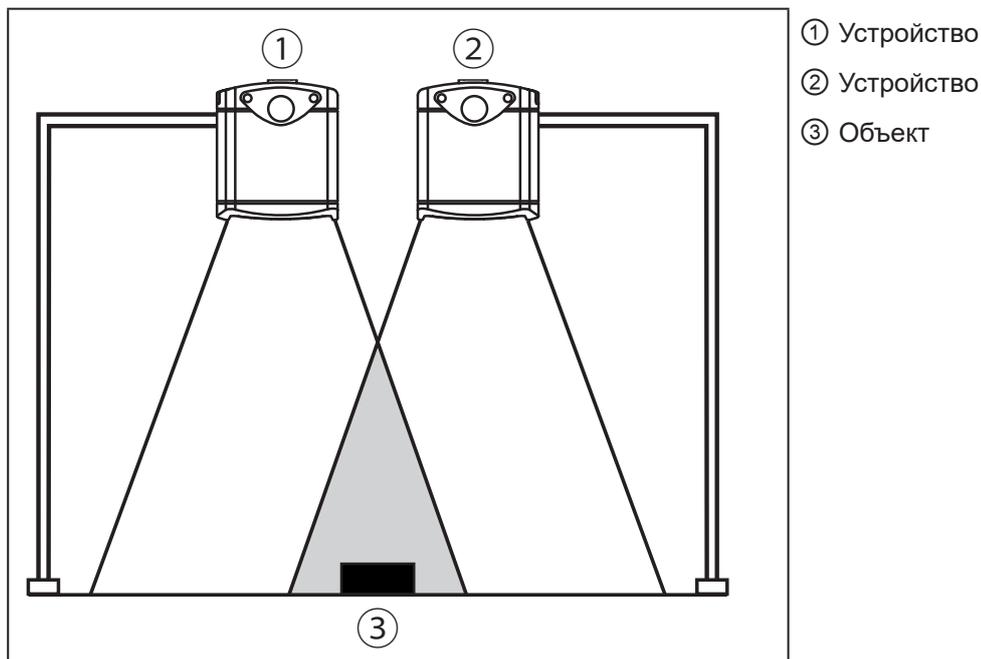
- с помощью Ethernet
- с помощью датчика приближения, подключенного к входу триггера

Следующий рисунок изображает подключение с датчиком приближения.



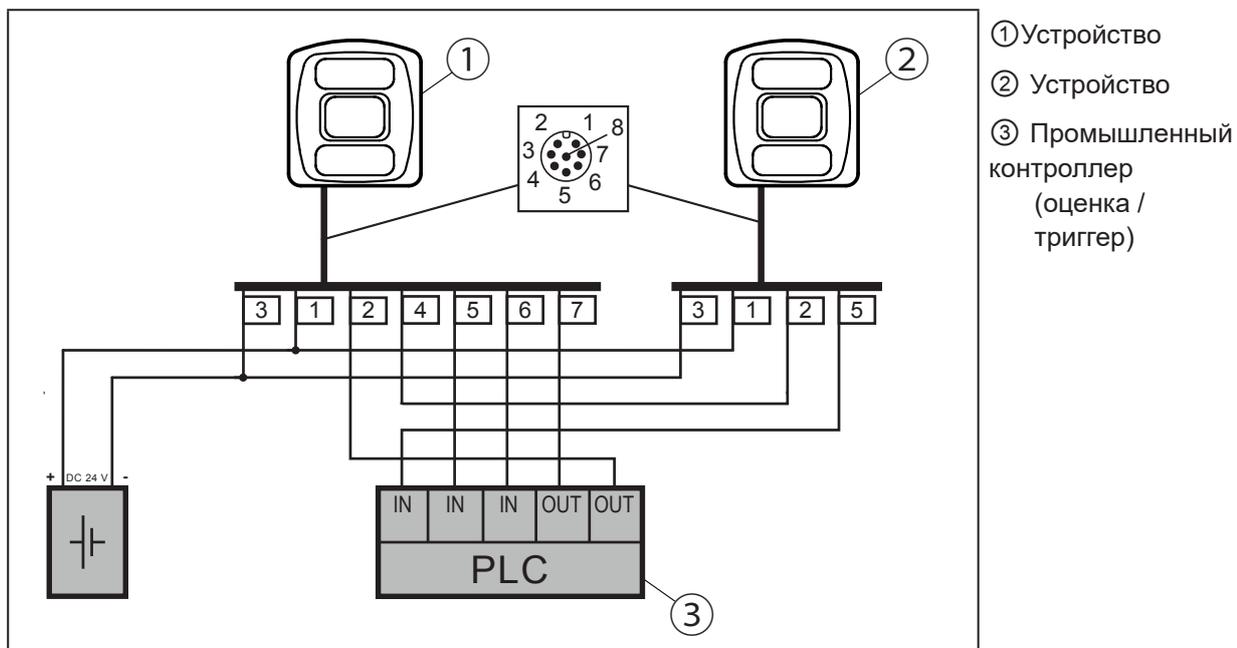
7.2.2 Установка датчиков рядом друг с другом

Датчики, установленные рядом друг с другом могут вызывать ошибки в измерении из-за одновременного воздействия.



Ошибок измерения можно избежать двумя способами:

- Каскадное подключение датчиков через HW-триггер
Во время каскадирования контроллер вызывает захват изображения датчика 1 (см. рис. ниже). После завершения захвата изображения, датчик 1 автоматически вызывает срабатывание датчика 2. Одновременно, контакт 4 датчика 1 обеспечивает состояние датчика "Захват изображения закончен". Датчик 2 сигнализирует конец последовательности в промышленный контроллер 3.



- Используйте различные частотные каналы
С помощью ПО ifm Vision Assistant к каждому датчику можно присоединить собственный частотный канал. Различные частотные каналы снижают появление ошибок измерения.

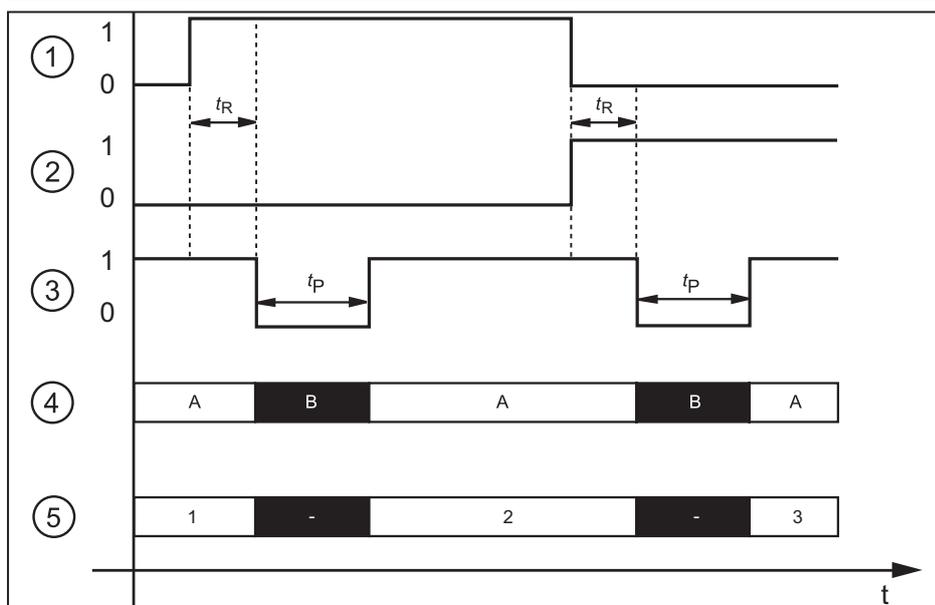


ПО ifm Vision Assistant доступно бесплатно на нашей веб-странице: www.ifm.com

7.3 Статический выбор программы

В датчике можно сохранить до 32 программ проверки. Первые четыре программы можно выбрать используя коммутационные входы.

Вход 2	Вход 1	Номер программы
0	0	1
0	1	2
1	0	3
1	1	4



Например: Выбор программы 1 → программа 2 → программа 3

①	Коммутационный вход 1 = 0 → 1 → 0
②	Коммутационный вход 2 = 0 → 0 → 1
③	Выход READY
④	Триггерный вход
	A: триггер включен B: триггер выключен
⑤	Идентификационный номер для активной программы

Для выбора программы время мониторинга t_R и время переключения триггера t_p необходимо принять во внимание.

Время мониторинга t_R : После изменений в фронте, внешний выбор программы нельзя начать до того как состояние обоих коммутационных входов остается стабильным на протяжении 20 мс.

Время переключения триггера t_p : Вход триггера отключен во время выбора программы. Время отключения зависит от:

- количества программ в датчике
- количества моделей в программе для активации



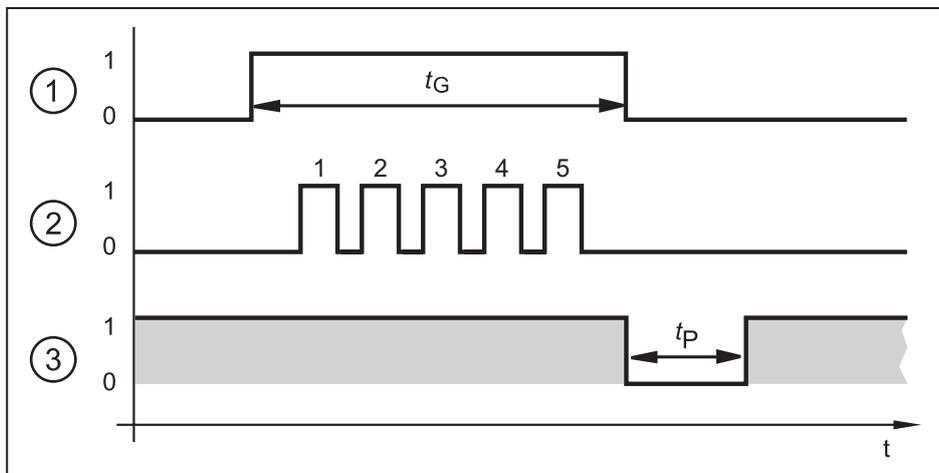
Изображение выше показывает выходную логику PNP (заводская настройка). Поведение выходной логики NPN противоположно выходной логике PNP:

- Выходная логика PNP: В случае высокого сигнала (1), используется напряжение.
- Выходная оптика NPN: В случае низкого сигнала (0), используется напряжение.

Для более подробной информации о конфигурации выбора программы мы советуем обратиться к руководству по программированию датчика. www.ifm.com

7.4 Выбор программы, управляемый импульсом

В качестве альтернативы к статическому выбору, выбор программы можно также управлять импульсом.



①	Стробирующий сигнал, коммутационный вход 1 = 0 → 1 → 0 (t_G = активный сигнал)
②	Импульсный сигнал, коммутационный вход 2 или триггерный вход = 0 → 5 импульсов → 0
③	Выход READY

Пока на коммутационном выходе 1 есть активный сигнал (стробирующий сигнал), прибор считает входящие импульсы и активирует соответствующие программы.

Количество импульсов = идентификационный номер программы

Для импульсного входа можно использовать коммутационный вход 2 или вход триггера.



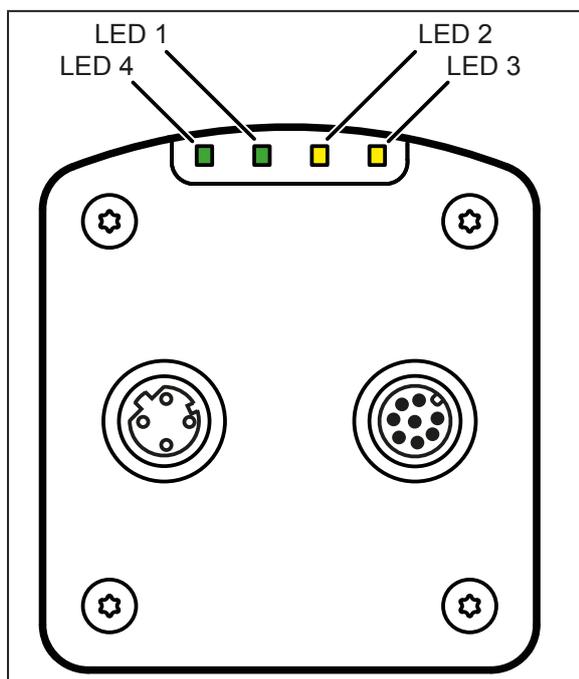
Изображение выше показывает выходную логику PNP (заводская настройка). Поведение выходной логики NPN противоположно выходной логике PNP:

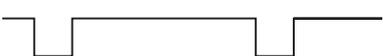
- Выходная логика PNP: В случае высокого сигнала (1), используется напряжение.
- Выходная логика NPN: В случае низкого сигнала (0), используется напряжение.

Для более подробной информации о конфигурации выбора программы мы советуем обратиться к руководству по программированию прибора. www.ifm.com

8. Индикаторы

С помощью светодиодов 1 - 4 датчик сигнализирует текущее рабочее состояние.



LED 4 (Ethernet)	LED 1 (Power)	LED 2 (Out 1)	LED 3 (Out 2)	Описание
	On			Датчик готов к работе, напряжение питания подано
	Мигает при 0,5 Гц			Наборы параметров или настройка параметров не загружена в датчик On  Off
	Мигает 2x при 0.5 Гц			Датчик находится в режиме настройки параметров On  Off
		On		Коммутационный выход 1 разомкнут
		Мигает при 8 Гц		Коммутационный выход 1 замкнут
			On	Коммутационный выход 2 разомкнут
			Мигает при 8 Гц	Коммутационный выход 2 замкнут
On				Ethernet подключен
Мигает				Передача данных через Ethernet
Off				Ethernet не подключен
		Мигает при 8 Гц	Мигает при 8 Гц	Датчик сигнализирует внутреннюю ошибку
		Мигает при 2 Гц	Мигает при 2 Гц	Датчик сигнализирует исправляемую ошибку. Информацию о ошибке можно прочитать через Ethernet
	Светится постоянно ⇒			Загрузка прибора
	Светится постоянно ⇐			Датчик производит обновление прошивки

9. Ввод в эксплуатацию

После подачи питания прибор вводится в эксплуатацию. Через 15 секунд датчик находится в режиме оценки, где выполняются сохраненные программы. Индикаторы отображают текущее рабочее состояние (→ 8).



В датчике можно сохранить до 32 программ. Приложение можно активировать различными способами:

- ПО ifm Vision Assistant
- Команда интерфейса
- Коммутационный вход 1 и 2
- Коммутационный вход 1 и триггерный вход

9.1 Настройки параметров датчика

Датчик настраивается с помощью ПО ifm Vision Assistant (→ см. руководство по программированию).



ПО ifm Vision Assistant и подробная информация о принципе измерения датчика описана в руководстве к ПО.

ПО ifm Vision Assistant доступно бесплатно на нашей веб-странице:

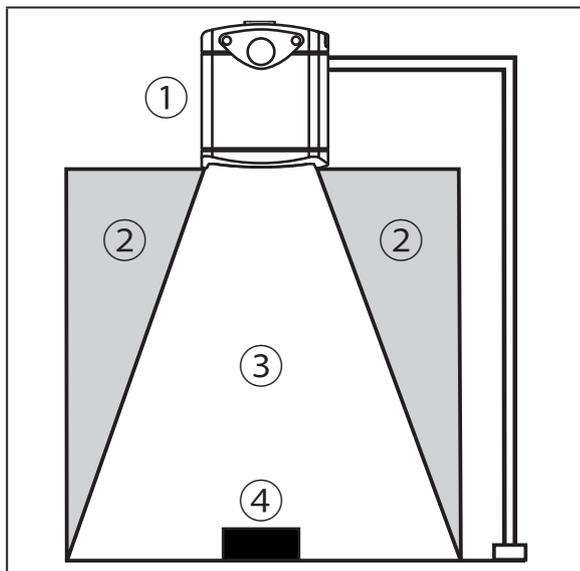
www.ifm.com

Руководство к ПО доступно на нашем сайте:

www.ifm.com

9.2 Обнаружение объекта

Условия для высокой скорости обнаружения объектов указаны ниже.



- ① Устройство
- ② Зона воздействия
- ③ Поле зрения
- ④ Объект

Оптимальное обнаружение объекта ④ обеспечивается при соблюдении следующих условий:

- Объект находится в поле зрения ③
- Объект является ближайшим видимым объектом датчика ①
- В зоне воздействия ② нет объектов (препятствия и т.д.)
- Стекло объектива датчика чистое.



Если условия не соблюдаются могут возникнуть ошибки в измерении.

9.3 Передача рабочих значений

9.3.1 Передача рабочих значений параметров мониторинга комплектности с помощью EtherNet/IP

Прибор может передавать рабочие значения в ПЛК через полевую шину EtherNet/IP. Рабочие значения отображаются в ifm Vision Assistant в виде выходной строки:

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

 Одновременно может быть активна только одна шина. Шину можно настраивать (→ инструкция по программному обеспечению).

В выходной строке рабочие значения отделяются точкой с запятой. Выходная строка передается в ПЛК в указанной последовательности.

 Соблюдайте следующие требования для передачи выходной строки в ПЛК:

- Байты от 0 до 7 являются частью выходной строки. Они не отображаются в ifm Vision Assistant (см. скриншот выше)
- Точки с запятыми ";" не передаются в выходной строке.
- Значения с плавающей запятой преобразуются в 16-разрядные целые числа до передачи.
- Все цифровые значения преобразуются в бинарные 16-разрядные целые числа до передачи:

Выходная строка состоит из следующих данных:

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

Номер байта	Данные	Кодировка	Рабочее значение	Прибор	Описание	Комментарии
0	2#0000_0000	Бинарный	1.5		Сдвоенное командное слово	<ul style="list-style-type: none"> • Бит 1,5 показывает успешную команду триггера
1	2#0010_0000	Бинарный				
2	2#0000_0000	Десятичный			Идентификация синхронного / асинхронного сообщения	
3	2#0000_0000	Десятичный				
4	30	Десятичный	30		Счетчик сообщений	<ul style="list-style-type: none"> • Прибор получил 30 сообщений • Увеличивается на 1 с каждым действием (триггер, отправленное сообщение итд.).
5	0	Десятичный				
6	0	Десятичный			Зарезервирован	
7	0	Десятичный				
8	c	ASCII	звезда		Стартовая строка	
9	t	ASCII				
10	A	ASCII				
11	r	ASCII				
12	0	Десятичный	0		Состояние всех ROI (0 = плохой, 1 = хороший)	Показывает состояние мониторинга целостности групповой упаковки
13	0	Десятичный				

Номер байта	Данные	Кодировка	Рабочее значение	Прибор	Описание	Комментарии
14	0	Десятичный				С активированной настройкой положения используются биты 14 и 15. 0 = положение не настроено 1 = положение настроено Все следующие данные сдвинуты на 2 байта; т.е. первый ROI ID начинается с байтов 16 и 17.
15	0	Десятичный	0		ROI ID	
16	0	Десятичный	0		Состояние ROI	Состояние ROI: 0 = хороший 1 = опорный уровень не обучен 2 = обучение не удалось 3 = опорный уровень не действителен 4 = нет действительных пикселей 5 = опорный уровень не содержит никаких действительных пикселей 6 = переполненный 7 = незаполненный
17	0	Десятичный			Состояние ROI	
18	0	Десятичный	0	ММ	Значение ROI	
19	0	Десятичный			Значение ROI	
20	1	Десятичный	1		ROI ID	
21	0	Десятичный			ROI ID	
22	7	Десятичный	7		Состояние ROI	
23	0	Десятичный			Состояние ROI	
24	-67	Десятичный	-67	ММ	Значение ROI	
25	-1	Десятичный			Значение ROI	
26	2	Десятичный	2		ROI ID	
27	0	Десятичный			ROI ID	
28	6	Десятичный	6		Состояние ROI	
29	0	Десятичный			Состояние ROI	
30	14	Десятичный	14	ММ	Значение ROI	
31	0	Десятичный			Значение ROI	
32	3	Десятичный	3		ROI ID	
33	0	Десятичный			ROI ID	
34	0	Десятичный	0		Состояние ROI	
35	0	Десятичный			Состояние ROI	
36	0	Десятичный	0	ММ	Значение ROI	
37	0	Десятичный			Значение ROI	
38	c	ASCII	останов		Конечная строка	
39	t	ASCII				
40	o	ASCII				
41	p	ASCII				



Неправильное выполнение команды приводит к следующим состояниям:

- Ошибка бит = 1
- Отображаются сдвоенные командные слова
- Асинхронное сообщение бит = 0
- Идентификация асинхронного сообщения = 0
- Счетчик сообщений увеличивается на 1

9.3.2 Передача рабочих значений параметров мониторинга комплектности с помощью PROFINET

Прибор может передавать рабочие значения в ПЛК через шину PROFINET. Рабочие значения отображаются в ifm Vision Assistant в виде выходной строки:

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

 Одновременно может быть активна только одна шина. Шину можно настраивать (→ инструкция по программному обеспечению).

В выходной строке рабочие значения отделяются точкой с запятой. Выходная строка передается в ПЛК в указанной последовательности.

 Соблюдайте следующие требования для передачи выходной строки в ПЛК:

- Байты от 0 до 7 являются частью выходной строки. Они не отображаются в ifm Vision Assistant (см. скриншот)
- Точки с запятыми ";" не передаются в выходной строке.
- Значения с плавающей запятой преобразуются в бинарные целые числа до передачи.
- Все цифровые значения преобразуются в бинарные 16-битные целые числа до передачи:

Выходная строка состоит из следующих данных (:

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

Номер байта	Данные	Кодировка	Рабочее значение	Прибор	Описание	Комментарии
0	2#0010_0000	Бинарный	0.5		Сдвоенное командное слово	• Бит 0,5 показывает успешную команду триггера
1	2#0000_0000	Бинарный				
2	2#0000_0000	Десятичный			Идентификация синхронного / асинхронного	
3	2#0000_0000	Десятичный			сообщения	
4	0	Десятичный	30		Счетчик сообщений	• Прибор получил 30 сообщений. • Увеличивается на 1 с каждым действием (триггер, отправленное сообщение итд.).
5	30	Десятичный				
6	0	Десятичный			Зарезервирован	
7	0	Десятичный				
8	s	ASCII	звезда		Стартовая строка	
9	t	ASCII				
10	A	ASCII				
11	r	ASCII				
12	0	Десятичный	0		Состояние всех ROI (0 = плохой, 1 = хороший)	Отображает состояния мониторинга целостности групповой упаковки
13	0	Десятичный				

Номер байта	Данные	Кодировка	Рабочее значение	Прибор	Описание	Комментарии
14	0	Десятичный				С активированной настройкой положения используются биты 14 и 15. 0 = положение не настроено 1 = положение настроено Все следующие данные сдвинуты на 2 байта; т.е. первый ROI ID начинается с байтов 16 и 17.
15	0	Десятичный	0		ROI ID	
16	0	Десятичный	0		Состояние ROI	Состояние ROI: 0 = хороший 1 = опорный уровень не обучен 2 = обучение не удалось 3 = опорный уровень не действителен 4 = нет действительных пикселей 5 = опорный уровень не содержит никаких действительных пикселей 6 = переполненный 7 = незаполненный
17	0	Десятичный	0		Состояние ROI	
18	0	Десятичный	0	мм	Значение ROI	
19	0	Десятичный	0		Значение ROI	
20	0	Десятичный	1		ROI ID	
21	1	Десятичный	1		ROI ID	
22	0	Десятичный	7		Состояние ROI	
23	7	Десятичный	7		Состояние ROI	
24	-1	Десятичный	-67	мм	Значение ROI	
25	-67	Десятичный	-67		Значение ROI	
26	0	Десятичный	2		ROI ID	
27	2	Десятичный	2		ROI ID	
28	0	Десятичный	6		Состояние ROI	
29	6	Десятичный	6		Состояние ROI	
30	0	Десятичный	14	мм	Значение ROI	
31	14	Десятичный	14		Значение ROI	
32	0	Десятичный	3		ROI ID	
33	3	Десятичный	3		ROI ID	
34	0	Десятичный	0		Состояние ROI	
35	0	Десятичный	0		Состояние ROI	
36	0	Десятичный	0	мм	Значение ROI	
37	0	Десятичный	0		Значение ROI	
38	с	ASCII				Конечная строка
39	t	ASCII				
40	о	ASCII				
41	р	ASCII				



Неправильное выполнение команды приводит к следующим состояниям:

- Ошибка бит = 1
- Отображаются сдвоенные командные слова
- Асинхронное сообщение бит = 0
- Идентификация асинхронного сообщения = 0
- Счетчик сообщений увеличивается на 1

9.3.3 Передача рабочих значений мониторинга комплектности через TCP/IP

Устройство может передавать рабочие значения в ПЛК через протокол TCP/IP. Рабочие значения отображаются в ifm Vision Assistant в виде выходной строки:

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

В выходной строке рабочие значения отделяются точкой с запятой. Выходная строка передается в ПЛК в указанной последовательности.



Соблюдайте следующие требования для передачи выходной строки в ПЛК:

- Точки с запятыми ";" не передаются в выходной строке.
- Все цифровые значения преобразуются в бинарные 16-разрядные целые числа до передачи.

Выходная строка состоит из следующих данных (тип данных: ASCII):

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

Рабочее значение	Прибор	Описание
звезда		Стартовая строка
0		Состояние всех ROI (0 = плохой, 1 = хороший)
00 0 +0,000	M	ROI ID Состояние ROI Значение ROI
01 7 -0,068	M	ROI ID Состояние ROI Значение ROI
02 6 +0,013	M	ROI ID Состояние ROI Значение ROI
03 0 +0,001	M	ROI ID Состояние ROI Значение ROI
останов		Конечная строка

Состояние ROI:
 0 = хороший
 1 = опорный уровень не обучен
 2 = обучение не удалось
 3 = опорный уровень не действителен
 4 = нет действительных пикселей
 5 = опорный уровень не содержит никаких действительных пикселей
 6 = переполненный
 7 = незаполненный

9.3.4 Передача рабочих значений, определяющих размеры объекта через EtherNet/IP

Прибор может передавать рабочие значения в ПЛК через полевую шину EtherNet/IP. Рабочие значения отображаются в ifm Vision Assistant в виде выходной строки:

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

 Одновременно может быть активной только одна шина. Шину можно настраивать (→ инструкция по программному обеспечению).

В выходной строке рабочие значения отделяются точкой с запятой. Выходная строка передается в ПЛК в указанной последовательности.

 Соблюдайте следующие требования для передачи выходной строки в ПЛК:

- Выходная строка настраивается. Передаваемые рабочие значения можно настроить в ifm Vision Assistant.
- Байты от 0 до 7 являются частью выходной строки. Они не отображаются в ifm Vision Assistant (см. скриншот)
- Точки с запятыми ";" не передаются в выходной строке.
- Значения с плавающей запятой преобразуются в 16-разрядные целые числа до передачи.
- Все цифровые значения преобразуются в бинарные 16-разрядные целые числа до передачи.

Выходная строка состоит из следующих данных:

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

Номер байта	Данные	Кодировка	Рабочее значение	Прибор	Описание	Комментарии
0	2#0000_0000	Бинарный			Сдвоенное командное слово	• Бит 1.5 показывает успешную команду триггера
1	2#0010_0000	Бинарный	1.5			
2	2#0000_0000	Бинарный			Идентификация синхронного / асинхронного	
3	2#0000_0000	Бинарный			сообщения	
4	2#0000_0011	Бинарный			Счетчик сообщений	• Прибор получил 3 сообщения. • Увеличивается на 1 с каждым действием (триггер, отправленное сообщение и т.д.).
5	2#0000_0000	Бинарный	3			
6	2#0000_0000	Бинарный			Зарезервирован	
7	2#0000_0000	Бинарный				
8	s	ASCII			Стартовая строка	
9	t	ASCII	звезда			
10	A	ASCII				
11	r	ASCII				
12	2#0000_0001	Бинарный		1		Результат бита
13	2#0000_0000	Бинарный				
14	104	Десятичный	104	мм	Ширина	
15	0	Десятичный				
16	88	Десятичный	88	мм	Высота	
17	0	Десятичный				
18	108	Десятичный	109	мм	Длина	
19	0	Десятичный				

Номер байта	Данные	Кодировка	Рабочее значение	Прибор	Описание	Комментарии
20	21	Десятичный	21		Координата x	
21	0	Десятичный				
22	-11	Десятичный	-11		Координата y	
23	-1	Десятичный				
24	-124	Десятичный	389		Координата z	
25	1	Десятичный				
26	-98	Десятичный	158		Степень вращения	
27	0	Десятичный				
28	97	Десятичный	97		Идеальность формы по ширине	
29	0	Десятичный				
30	93	Десятичный	94		Идеальность формы по высоте	
31	0	Десятичный				
32	97	Десятичный	97		Идеальность формы по длине	
33	0	Десятичный				
34	с	ASCII	останов		Конечная строка	
35	t	ASCII				
36	o	ASCII				
37	p	ASCII				



Неправильное выполнение команды приводит к следующим состояниям:

- Ошибка бит = 1
- Отображаются сдвоенные командные слова
- Асинхронное сообщение бит = 0
- Идентификация асинхронного сообщения = 0
- Счетчик сообщений увеличивается на 1

9.3.5 Передача рабочих значений, определяющих размеры объекта через PROFINET

Прибор может передавать рабочие значения в ПЛК через шину PROFINET. Рабочие значения отображаются в ifm Vision Assistant в виде выходной строки:

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

 Одновременно может быть активной только одна шина. Шину можно настраивать (→ инструкция по программному обеспечению).

В выходной строке рабочие значения отделяются точкой с запятой. Выходная строка передается в ПЛК в указанной последовательности.

 Соблюдайте следующие требования для передачи выходной строки в ПЛК:

- Выходная строка настраивается. Передаваемые рабочие значения можно настроить в ifm Vision Assistant.
- Байты от 0 до 7 являются частью выходной строки. Они не отображаются в ifm Vision Assistant (см. скриншот)
- Точки с запятыми ";" не передаются в выходной строке.
- Значения с плавающей запятой преобразуются в 16-разрядные целые числа до передачи.
- Все цифровые значения преобразуются в бинарные 16-разрядные целые числа до передачи.

Выходная строка состоит из следующих данных:

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

Номер байта	Данные	Кодировка	Рабочее значение	Прибор	Описание	Комментарии
0	2#0010_0000	Бинарный	0.5		Сдвоенное командное слово	• Бит 0,5 показывает успешную команду триггера
1	2#0000_0000	Бинарный				
2	2#0000_0000	Бинарный	3		Идентификация синхронного / асинхронного сообщения	• Прибор получил 3 сообщения. • Увеличивается на 1 с каждым действием (триггер, отправленное сообщение итд.).
3	2#0000_0000	Бинарный				
4	2#0000_0000	Бинарный				
5	2#0000_0011	Бинарный			Счетчик сообщений	
6	2#0000_0000	Бинарный			Зарезервирован	
7	2#0000_0000	Бинарный				
8	s	ASCII	звезда		Стартовая строка	
9	t	ASCII				
10	A	ASCII				
11	r	ASCII				
12	2#0000_0000	Бинарный	1		Результат бита	0 = коробка не найдена 1 = коробка найдена
13	2#0000_0001	Бинарный				
14	0	Десятичный	104	мм	Ширина	
15	104	Десятичный				
16	0	Десятичный	88	мм	Высота	
17	88	Десятичный				
18	0	Десятичный	109	мм	Длина	
19	109	Десятичный				

Номер байта	Данные	Кодировка	Рабочее значение	Прибор	Описание	Комментарии
20	0	Десятичный	21		координата x	
21	21	Десятичный				
22	-1	Десятичный	-11		координата y	
23	-11	Десятичный				
24	1	Десятичный	389		координата z	
25	-124	Десятичный				
26	0	Десятичный	158		Степень вращения	
27	-98	Десятичный				
28	0	Десятичный	97		Идеальность формы по ширине	
29	97	Десятичный				
30	0	Десятичный	94		Идеальность формы по высоте	
31	94	Десятичный				
32	0	Десятичный	97		Идеальность формы по длине	
33	97	Десятичный				
34	с	ASCII	останов		Конечная строка	
35	t	ASCII				
36	o	ASCII				
37	p	ASCII				

 Неправильное выполнение команды приводит к следующим состояниям:

- Ошибка бит = 1
- Отображаются сдвоенные командные слова
- Асинхронное сообщение бит = 0
- Идентификация асинхронного сообщения = 0
- Счетчик сообщений увеличивается на 1

9.3.6 Передача рабочих значений, определяющих размеры объекта через TCP/IP

Устройство может передавать рабочие значения в ПЛК через протокол TCP/IP. Передаваемые рабочие значения можно выбрать в ifm Vision Assistant. Рабочие значения отображаются в ifm Vision Assistant в виде выходной строки:

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

В выходной строке рабочие значения отделяются точкой с запятой. Выходная строка передается в ПЛК в указанной последовательности.



Соблюдайте следующие требования для передачи выходной строки в ПЛК:

- Точки с запятыми ";" не передаются в выходной строке.
- Все цифровые значения преобразуются в бинарные 16-разрядные целые числа до передачи:

Выходная строка состоит из следующих данных (тип данных: ASCII):

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

Рабочее значение	Прибор	Описание
звезда		Стартовая строка
1		Объект найден
0,104	м	Ширина
0,088	м	Высота
0,109	м	Длина
+0,021		координата x
-0,011		координата y
+0,389		координата z
158		Степень вращения
097		Идеальность формы по ширине
094		Идеальность формы по высоте
097		Идеальность формы по длине
останов		Конечная строка

9.3.7 Передача рабочих значений мониторинга уровня через EtherNet/IP

Прибор может передавать рабочие значения в ПЛК через полевую шину EtherNet/IP. Рабочие значения отображаются в ifm Vision Assistant в виде выходной строки:

0070

 Одновременно может быть активная только одна шина. Шину можно настраивать (→ инструкция по программному обеспечению).

Выходная строка передается в ПЛК в указанной последовательности.

 Соблюдайте следующие требования для передачи выходной строки в ПЛК:

- Байты от 0 до 7 являются частью выходной строки. Они не отображаются в ifm Vision Assistant (см. скриншот)
- Точки с запятыми ";" не передаются в выходной строке.
- Значения с плавающей запятой преобразуются в бинарные целые числа до передачи.
- Все цифровые значения преобразуются в бинарные 16-битные целые числа до передачи:

Выходная строка состоит из следующих данных (:

0070

Номер байта	Данные	Кодировка	Рабочее значение	Прибор	Описание	Комментарии
0	2#0000_0000	Бинарный	1.5		Сдвоенное командное слово	Бит 1.5 показывает успешную команду триггера
1	2#0010_0000	Бинарный				
2	2#0000_0000	Десятичный			Идентификация синхронного / асинхронного сообщения	
3	2#0000_0000	Десятичный				
4	30	Десятичный	30		Счетчик сообщений	<ul style="list-style-type: none"> • Прибор получил 30 сообщений. • Увеличивается на 1 с каждым действием (триггер, отправленное сообщение и т.д.).
5	0	Десятичный				
6	0	Десятичный			Зарезервирован	
7	0	Десятичный				
8	0	Десятичный	0		Состояние всех ROI (0 = плохой, 1 = хороший)	Отображает состояние измерения уровня
9	0	Десятичный				
10	0	Десятичный	0		ROI ID	Состояние ROI: 0 = хороший 6 = переполненный 7 = незаполненный
11	0	Десятичный				
12	7	Десятичный	7		Состояние ROI	
13	0	Десятичный				
14	0	Десятичный	0	MM	Значение ROI	
15	0	Десятичный				

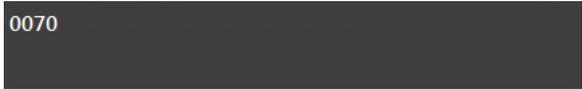
 Неправильное выполнение команды приводит к следующим состояниям:

- Ошибка бит = 1
- Отображаются сдвоенные командные слова
- Асинхронное сообщение бит = 0
- Идентификация асинхронного сообщения = 0
- Счетчик сообщений увеличивается на 1

RU

9.3.8 Передача рабочих значений измерения уровня через PROFINET

Прибор может передавать рабочие значения в ПЛК через шину PROFINET. Рабочие значения отображаются в ifm Vision Assistant в виде выходной строки:



0070

 Одновременно может быть активной только одна шина. Шину можно настраивать (→ инструкция по программному обеспечению).

Выходная строка передается в ПЛК в указанной последовательности.

 Соблюдайте следующие требования для передачи выходной строки в ПЛК:

- Байты от 0 до 7 являются частью выходной строки. Они не отображаются в ifm Vision Assistant (см. скриншот выше)
- Точки с запятыми ";" не передаются в выходной строке.
- Значения с плавающей запятой преобразуются в 16-разрядные целые числа до передачи.
- Все цифровые значения преобразуются в бинарные 16-разрядные целые числа до передачи:

Выходная строка состоит из следующих данных:

0070

Номер байта	Данные	Кодировка	Рабочее значение	Прибор	Описание	Комментарии
0	2#0010_0000	Бинарный	0.5		Сдвоенное командное слово	Бит 0,5 показывает успешную команду триггера
1	2#0000_0000	Бинарный				
2	2#0000_0000	Десятичный			Идентификация синхронного / асинхронного сообщения	
3	2#0000_0000	Десятичный				
4	0	Десятичный	30		Счетчик сообщений	<ul style="list-style-type: none"> • Прибор получил 30 сообщений. • Увеличивается на 1 с каждым действием (триггер, отправленное сообщение и т.д.).
5	30	Десятичный				
6	0	Десятичный			Зарезервирован	
7	0	Десятичный				
8	0	Десятичный	0		Состояние всех ROI (0 = плохой, 1 = хороший)	Отображает состояние измерения уровня
9	0	Десятичный				
10	0	Десятичный	0		ROI ID	Состояние ROI:
11	0	Десятичный				
12	0	Десятичный	7		Состояние ROI	0 = хороший
13	7	Десятичный				
14	0	Десятичный	0	MM	Значение ROI	6 = переполненный
15	0	Десятичный				

 Неправильное выполнение команды приводит к следующим состояниям:

- Ошибка бит = 1
- Отображаются сдвоенные командные слова
- Асинхронное сообщение бит = 0
- Идентификация асинхронного сообщения = 0
- Счетчик сообщений увеличивается на 1

9.3.9 Передача рабочих значений мониторинга уровня через TCP/IP

Устройство может передавать рабочие значения в ПЛК через протокол TCP/IP. Рабочие значения отображаются в ifm Vision Assistant в виде выходной строки:

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

В выходной строке рабочие значения отделяются точкой с запятой. Выходная строка передается в ПЛК в указанной последовательности.



Соблюдайте следующие требования для передачи выходной строки в ПЛК:

- Точки с запятыми ";" не передаются в выходной строке.
- Все цифровые значения преобразуются в бинарные 16-разрядные целые числа до передачи:

Выходная строка состоит из следующих данных (тип данных: ASCII):

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

Рабочее значение	Прибор	Описание
звезда		Стартовая строка
0		Состояние всех ROI (0 = плохой, 1 = хороший)
00		ROI ID
7		Состояние ROI
+0,000	m	Значение ROI
останов		Конечная строка

Состояние ROI:
0 = хороший
6 = переполненный
7 = незаполненный

10. Техническое обслуживание, ремонт и утилизация

Соблюдайте следующие указания:

- ▶ Не открывайте корпус прибора, так как в устройстве отсутствуют компоненты, которые могут обслуживаться пользователем. Ремонт прибора осуществляет только производитель.
- ▶ Утилизацию устройства выполняйте только в соответствии с национальными нормами о защите окружающей среды.

10.1 Очистка

Соблюдайте следующие инструкции до очистки датчика:

- ▶ Используйте чистую и безворсовую ткань.
- ▶ В качестве очистителя используйте средство для чистки стекол.



Если инструкция не соблюдается, царапины на стекле объектива могут вызвать ошибки в измерении.

10.2 Обновление прошивки

С помощью ifm Vision Assistant можно обновлять прошивку датчика.



Параметры, сохраненные в датчике после обновления прошивки теряются. Создайте резервную копию параметров перед обновлением прошивки.

- ▶ До обновления прошивки произведите экспорт параметров
- ▶ Произведите импорт параметров после обновления прошивки.



Обновления прошивки доступны на нашем сайте: www.ifm.com

10.3 Замена прибора

При замене прибора параметры теряются. Сделайте резервную копию параметров до замены прибора:

- ▶ Сделайте экспорт параметров старого датчика до замены.
- ▶ Сделайте импорт параметров в новый прибор после замены.



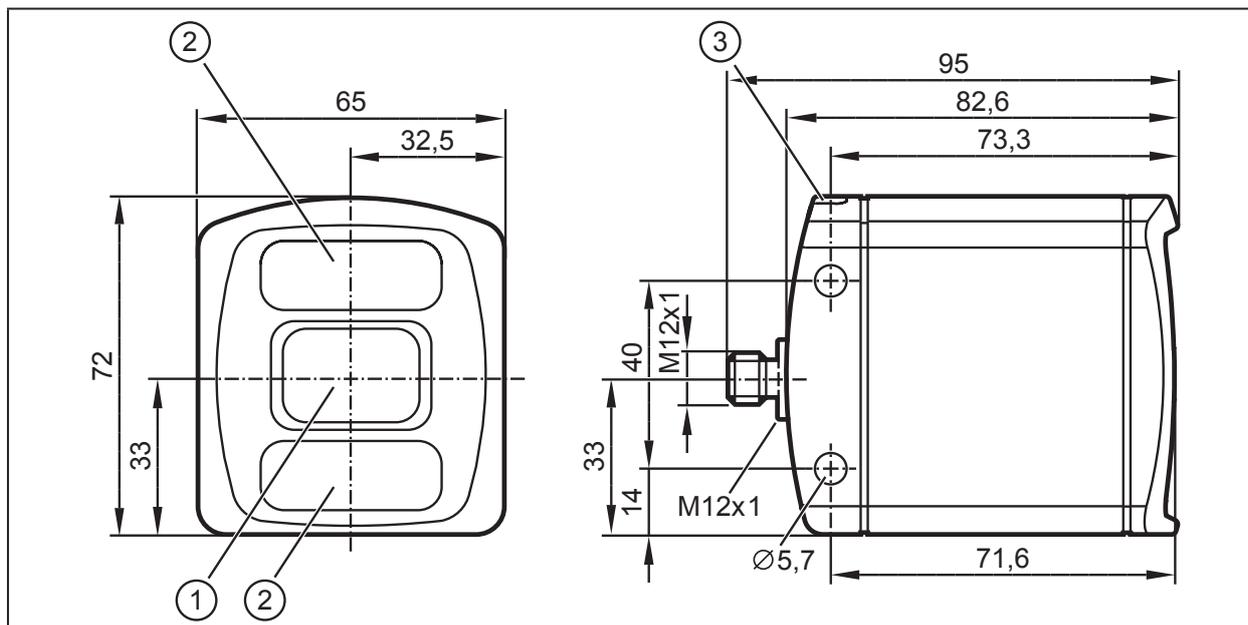
С помощью экспорта и импорта параметров в несколько датчиков можно быстро записать одинаковые параметры.

11. Разрешения/стандарты

Сертификат соответствия CE доступен на нашем сайте: www.ifm.com

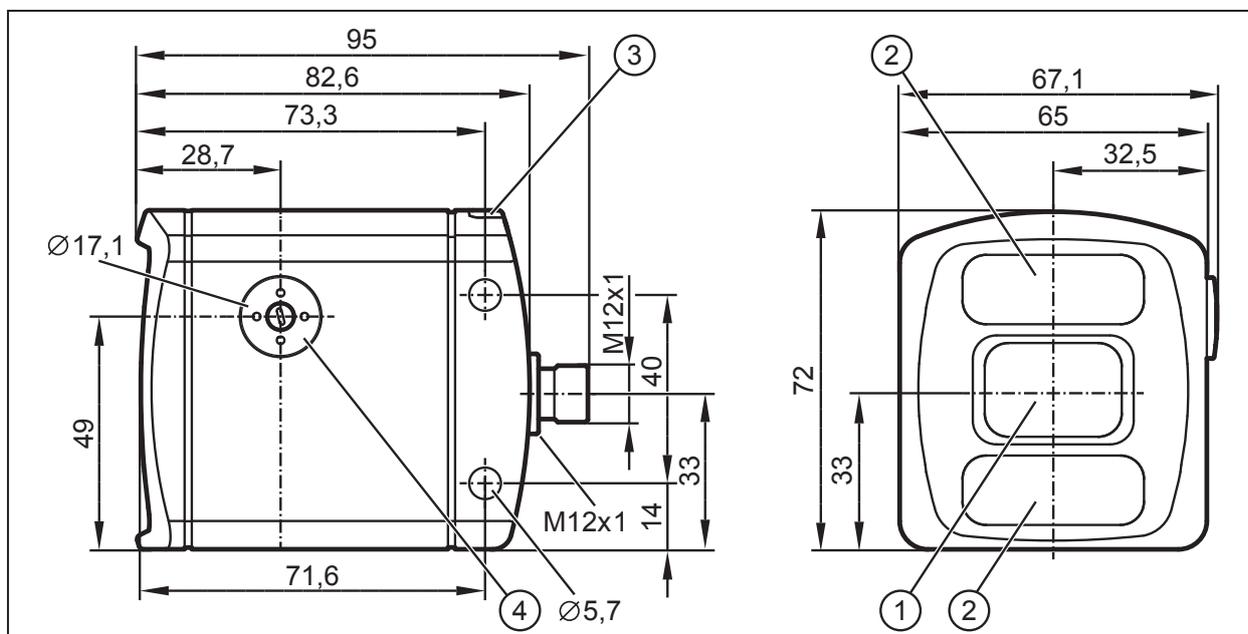
12. Типовые размеры

12.1 O3D302 / O3D312



- ① Стекло объектива
- ② Устройство подсветки
- ③ Двухцветный светодиод (желтый / зеленый)

12.2 O3D300 / O3D310



- ① Стекло объектива
- ② Устройство подсветки
- ③ Двухцветный светодиод (желтый / зеленый)
- ④ Кнопки настройки

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 (e.g. <code> is a placeholder for code)		
[]	Optional argument (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.  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:

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\n0010000500000:{"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:

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

Offset	Name	Description	Size [byte]
0x0014	IMAGE_HEIGHT	Image height in pixel	4
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	<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, i.e. 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. 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:

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

13.1.5 Configuration of PCIC Output

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

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

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

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

The layout software has the following main object properties:

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:

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

Available values for the type property:

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:

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"> • 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:

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

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

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

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

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

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

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.
	!	<ul style="list-style-type: none"> • 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.
	!	<ul style="list-style-type: none"> • 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>	
	!	<ul style="list-style-type: none"> No image available Wrong ID
	?	<ul style="list-style-type: none"> 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.

13.2.5 a Command

Command	a<application number>	
Description	Activates the selected application	
Type	Action	
Reply	*	
	!	<ul style="list-style-type: none"> • 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	*	
	!	<ul style="list-style-type: none"> • Error in configuration • Wrong data length
	?	Invalid command length
Note	<length> 9 digits as decimal value for the data length <configuration> configuration data	

13.2.10 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	<pre><vendor><t><article number><t> <name><t><location><t><descri ption><t><ip> <subnet mask><t><gateway>< t><MAC><t><DHCP><t><port number></pre>	
Note	<ul style="list-style-type: none"> • <vendor> IFM ELECTRONIC • <t> Tabulator (0x09) • <article number> e.g. O3D300 • <name> UTF8 Unicode string • <location> UTF8 Unicode string • <description> UTF8 Unicode string • <ip> IP address of the device as ASCII character sting e.g. 192.168.0.96 • <port number> port number of the XML-RPC • <subnet mask> subnet mask of the device as ASCII e.g. 192.168.0.96 • <gateway> gateway of the device as ASCII e.g 192.168.0.96 • <MAC> MAC adress of the device as ASCII e.g. AA:AA:AA:AA:AA:AA • <DHCP> ASCII string "0" for off and "1" for on 	

13.2.13 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<io-id><io-state> - sets IO state</p> <p>O<io-id>? - get IO state</p> <p>I<image-id>? - get last image of defined type</p> <p>A? - get application list</p> <p>p<state> - activate / deactivate data output</p> <p>a<application number> - set active application</p> <p>V? - get current protocol version</p> <p>v<version> - sets protocol version</p> <p>c<length of configuration file><configuration file> - 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"> • <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>	
	!	<ul style="list-style-type: none"> Invalid state (e.g. configuration mode) Wrong ID
	?	Invalid command length
Note	<ul style="list-style-type: none"> <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	<ul style="list-style-type: none"> <code> Error code with 8 digits as a decimal value. It contains leading zeros. 	

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

Command word

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.

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.

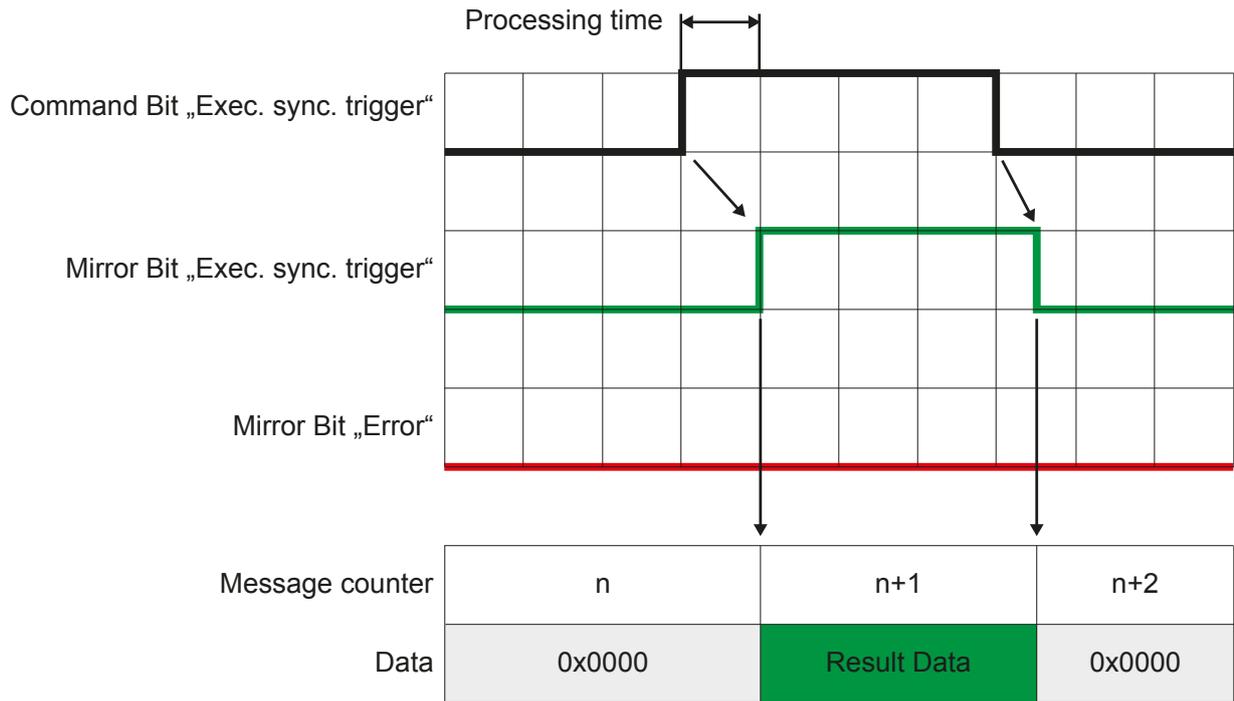
The data content of the processing assembly is set to 0.

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

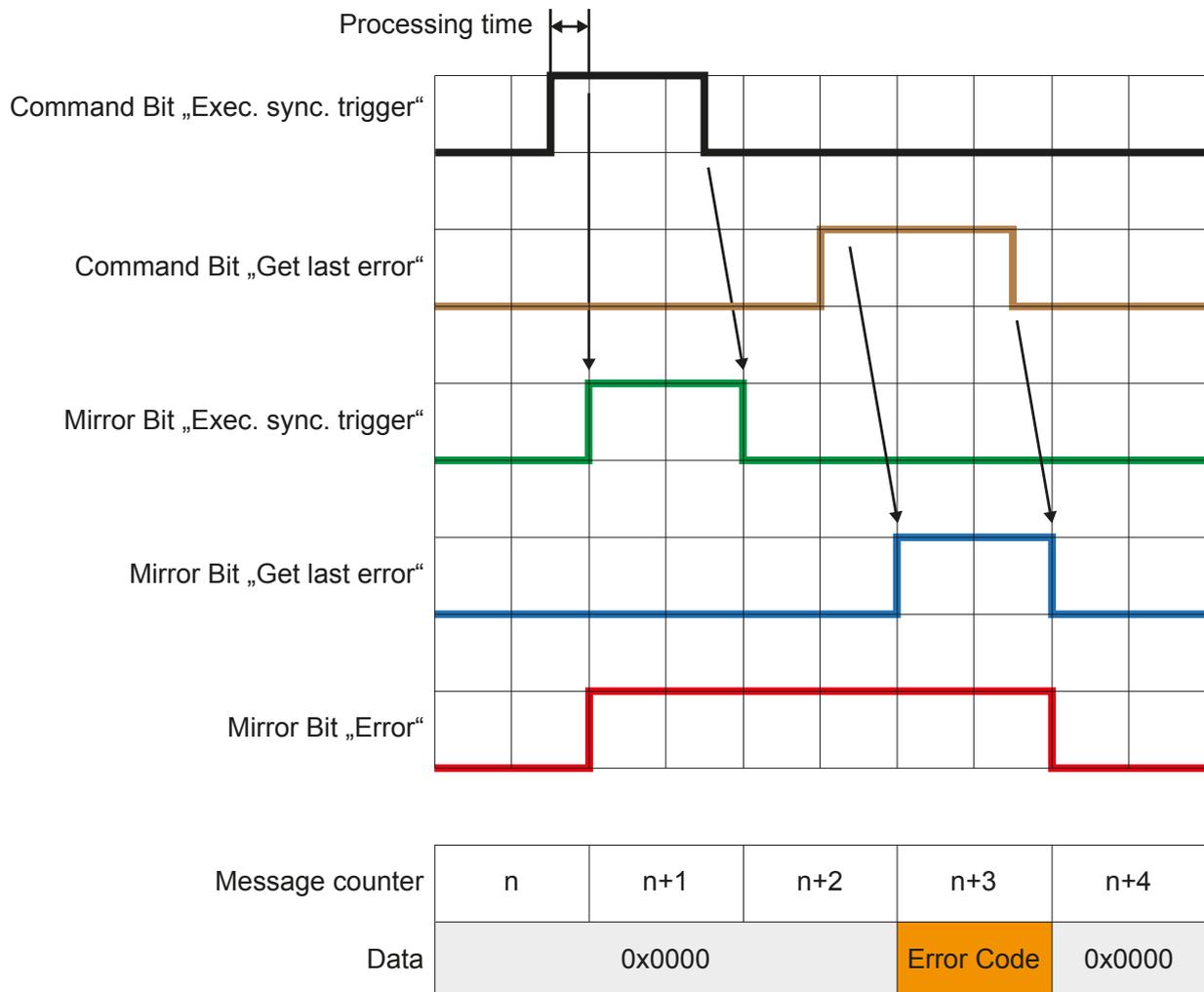
Default endianness

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.

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

Layout of command word

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

Command word

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

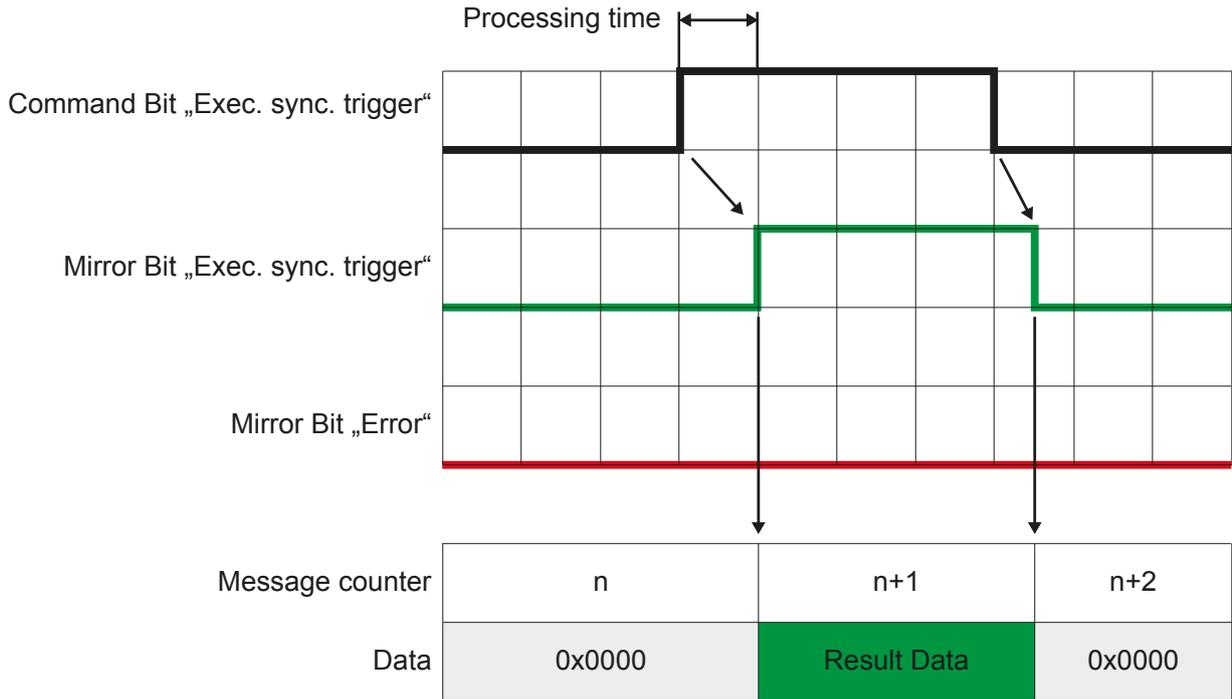
The bytes 2-5 have to be set to 0. An error is risen if bytes 2-5 are not set to 0.

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



13.5.4 Signal sequence with failed trigger

