



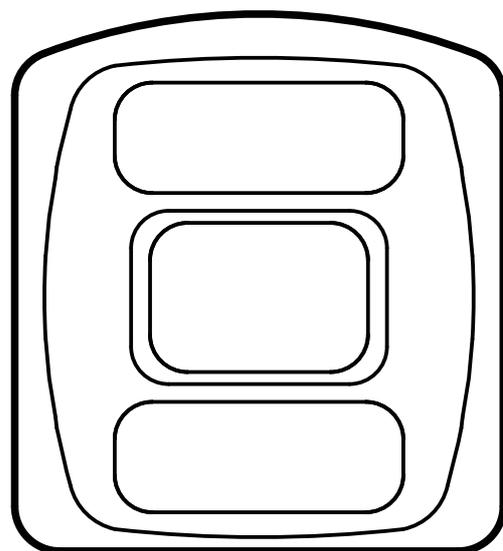
사용 설명서  
3D 센서

O3D300

O3D302

O3D310

O3D312



KR

## 목차

1. 서문	4
1.1 심볼마크	4
1.2 사용된 경고문	4
2. 안전에 관한 안내사항	4
2.1 일반사항	4
2.2 타겟 그룹	4
2.3 전기적 연결	4
2.4 디바이스 조작	4
3. 기능 및 특징	5
4. 공급 범위	5
5. 액세스서리	5
6. 설치	6
6.1 설치 위치 선택	6
6.2 추가 센서 설치 안내	7
6.2.1 O3D300 / O3D302에 대한 일반적인 경고 한계	7
6.2.2 O3D310 / O3D312에 대한 일반적인 경고 한계	8
6.2.3 표면 온도 감소	8
6.3 센서 설치	9
6.4 마운팅 액세스서리	9
7. 전기적 연결	10
7.1 배선	10
7.1.1 핀 1 / 3 (24 V / GND)	11
7.1.2 핀 2 (트리거 입력)	11
7.1.3 핀 4 / 5 / 6 (스위칭 출력)	11
7.1.4 핀 4 (아날로그 출력)	12
7.1.5 핀 7 / 8 (스위칭 입력)	12
7.2 배선 사례	13
7.2.1 근접센서로 이미지 캡처 트리거	13
7.2.2 여러 센서를 나란히 설치합니다	14
7.3 어플리케이션의 정적 선택	15
7.4 어플리케이션의 펄스 컨트롤된 선택	16
8. 표시기	17
9. 셋업	18
9.1 센서의 파라미터 설정	18
9.2 물체감지	18
9.3 프로세스 값 전송	19
9.3.1 EtherNet/IP를 통한 완전성 모니터링의 프로세스 값 전송	19
9.3.2 TCP/IP를 통한 완전성 모니터링의 프로세스 값 전송	21
9.3.3 EtherNet/IP를 통하여 물체 치수의 프로세스 값 전송	22
9.3.4 TCP/IP를 통하여 물체 치수의 프로세스 값 전송	24
9.3.5 EtherNet/IP를 통한 레벨 모니터링의 프로세스 값 전송	25
9.3.6 TCP/IP를 통한 레벨 모니터링의 프로세스 값 전송	26
10. 유지보수, 수리 및 폐기	27
10.1 세척	27
10.2 펌웨어 업데이트	27
10.3 센서 대체	27
11. 인증 / 표준	27
12. 도면	28
12.1 O3D302 / O3D312	28
12.2 O3D300 / O3D310	28
13. Appendix	29
13.1 Process Interface	29
13.1.1 Sending Commands	29
13.1.2 Receiving Images	31

13.1.3	Image data	31
13.1.4	Additional Information for CONFIDENCE_IMAGE	34
13.1.5	Configuration of PCIC Output	35
13.2	Process Interface Command Reference	41
13.2.1	t Command (Asynchronous Trigger)	41
13.2.2	T? Command (Synchronous Trigger)	41
13.2.3	l? Command	42
13.2.4	p Command	42
13.2.5	a Command	43
13.2.6	A? Command	43
13.2.7	v Command	44
13.2.8	V? Command	44
13.2.9	c Command	44
13.2.10	C? Command	45
13.2.11	S? Command	45
13.2.12	G? Command	46
13.2.13	H? Command	47
13.2.14	o Command	47
13.2.15	O? Command	48
13.3	Error codes	49
13.4	EtherNet/IP	50
13.4.1	Data structures for consuming and producing assemblies	50
13.4.2	Functionality of the Ethernet/IP application	51
13.5	PROFINET IO	55
13.5.1	Data structures for output and input frame	55
13.5.2	Functionality of PROFINET IO application	56

## 라이선스 및 상표

Microsoft®, Windows®, Windows XP®, Windows Vista®, Windows 7®, Windows 8® 그리고 Windows 8.1®은 Microsoft Corporation의 등록된 상표입니다.

Adobe®와 Acrobat®는 Adobe Systems Inc의 등록된 상표입니다.

모든 상표 및 회사이름은 해당 회사의 저작권이 적용됩니다.

본 디바이스는 (경우에 따라 변경) 특별 라이선스 조건에 따라 달라질 수 있는 오픈 소스 소프트웨어 (Open Source Software)를 포함합니다.

저작권 정보 및 라이선스 기간:

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

GNU General Public License 또는 GNU Lesser General Public License에 대한 소프트웨어 대상을 위한 소스 코드는 복사 및 배송 비용의 지불을 요청 할 수 있습니다.

## 1. 서문

본 문서는 전문가를 위해 작성되었습니다. 이러한 전문가들은 상응되는 교육과 그들의 경험으로 디바이스의 작동이나 유지보수시 발생가능한 위험을 인식하고 잠재적인 위험을 방지할 수 있는 자격이 있는 사람들을 의미합니다. 본 문서는 디바이스의 정확한 조작에 대한 정보가 포함되어 있습니다.

작동조건, 설치 및 작동을 숙지하기 위하여 사용하기 전에 설명서를 읽으십시오.  
디바이스가 사용되는 한 본 문서는 보관되어야 합니다.

### 1.1 심볼마크

- ▶ 설명서
- > 반응, 결과
- [...] 누름버튼, 스위치 또는 표시 지정
- 참고사항
-  주의사항  
부주의한 사용은 오작동이나 장애를 초래합니다.
-  정보  
추가 참고문

### 1.2 사용된 경고문

<b>참고</b>
재산 손실에 관한 경고

## 2. 안전에 관한 안내사항

### 2.1 일반사항

본 안내사항은 디바이스의 일부입니다. 안내사항은 디바이스의 정확한 취급에 관한 텍스트 및 관련된 수치가 포함되어 있으므로 설치 또는 사용전에 반드시 읽어야 합니다.

사용설명서를 참조하십시오. 다음의 참고문에 유의하지 않거나 사용에 관한 정해진 규정을 준수하지 않은 사용, 잘못된 설치 또는 취급은 사람과 설비의 안전에 심각한 영향을 미칠 수 있습니다.

### 2.2 타겟 그룹

본 사용설명서는 전자적 적합성 (EMC), 낮은 전압 지침에 따라 자격이 부여된 전문가를 위하여 작성되었습니다. 디바이스는 반드시 자격을 갖춘 전기 기술자에 의하여 설치, 접속 그리고 가동시켜야 합니다.

### 2.3 전기적 연결

디바이스를 취급하기 전에 외부 장치의 연결을 차단하십시오.

접속 핀은 기술 데이터 및/ 디바이스 라벨에 표시된 신호만으로 공급될 수 있으며, ifm의 입증된 액세서리로만 연결될 수 있습니다.

### 2.4 디바이스 조작

오작동이나 불확실한 경우, 제조업체에 직접 문의하십시오. 디바이스의 부당한 변경은 조작자 및 기계의 안전에 심각한 영향을 초래할 수 있습니다. 부당한 변경은 허용되지 않으며, 모든 책임과 보증이 배제됩니다.

### 3. 기능 및 특징

O3D3xx 3D 센서는 이동거리 시간차 원리를 사용하여 센서와 가장 가까운 표면 사이의 거리를 하나씩 측정하는 포토 센서입니다. 센서는 적외선 광원으로 장면을 조명하고 표면에서 반사된 광선을 사용하여 거리를 계산합니다.

이미지 데이터로부터 프로세스 값은 내부 이미지 처리를 통하여 생성되고 임계 값과 비교됩니다. 비교 및 프로세스 값은 디지털 출력과 연결됩니다. 이렇게하면 다음 어플리케이션이 해결될 수 있습니다:

- 완전성 모니터링
- 레벨 모니터링
- 간격 모니터링
- 사각형 물체의 치수
- 사각형 물체의 분류

측정된 데이터 및 프로세스 값은 이더넷을 통해 제공되고 사용자가 평가할 수 있습니다. 센서의 파라미터 세팅 또한 이더넷을 통하여 이루어 집니다.

센서는 데이터 시트에 명시된 작동 조건에서만 사용될 수 있습니다.

센서 안전은 다음과 같은 작동 조건에서 사용되도록 등급되어 있습니다:

- 실내 사용
- 2000 m 까지의 고도
- 상대 습도 최대 90 %, 비 응축
- 오염등급 3

전자기 간섭 방출에 대한 요구사항 때문에 본 센서는 산업 환경에서 사용하도록 설계되었습니다. 센서는 가정용으로 디자인되지 않았습니다.



센서는 데이터 시트에 명시된 환경 조건에서만 사용될 수 있습니다.

### 4. 공급 범위

- O3D3xx 3D 센서
- 소프트웨어 및 문서를 보유한 USB 메모리 스틱
- 간략한 설명서, 식별 번호 80236220



데이터시트 및 기타 문서(소프트웨어 매뉴얼 등)는 당사 웹 사이트에서 찾으실 수 있습니다:  
[www.ifm.com](http://www.ifm.com) → 데이터시트 찾기 → 예: O3D302 → 사용 설명서

### 5. 액세서리

다음 액세서리는 센서의 작동을 위하여 필요합니다:

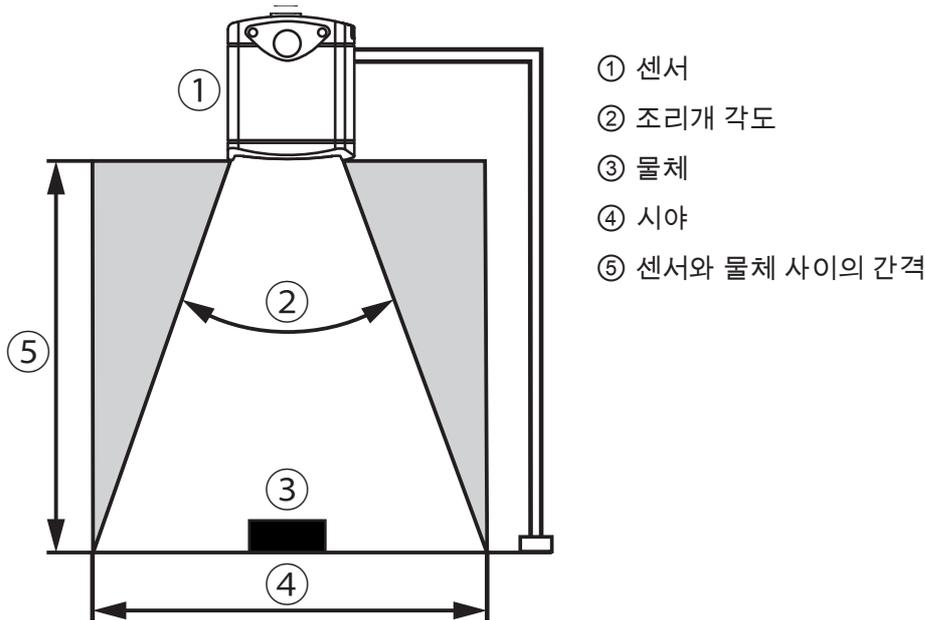
제품 번호	설명
E11950	카메라/센서 전원 공급 케이블
E11898	M12 산업용 이더넷 연결 케이블



ifm Vision Assistant 소프트웨어는 웹사이트에서 무료로 제공됩니다:  
[www.ifm.com](http://www.ifm.com) → 고객지원 → 다운로드 → 산업용 이미징

## 6. 설치

이 챕터에서는 설치 전의 유의사항 및 센서 설치 방법에 대해 설명합니다.



### 6.1 설치 위치 선택

설치 위치 선택에 대한 다음 지시사항에 유의하십시오.

- ▶ 물체 ③ 은 완전히 시야 ④내에 있어야 합니다.
- > 시야의 크기는 센서 타입에 따라 다르며 데이터 시트에 표시됩니다. 시야의 크기는 물체에 대한 센서와의 거리 ⑤에 따라 달라집니다: 거리가 멀어질수록 시야가 커집니다.
- ▶ 물체를 위치시킬때 공차를 고려하십시오.
- ▶ 센서와 물체 사이의 거리를 결정할 때 센서의 측정 범위 ⑤를 고려하십시오.
- > 측정범위는 센서의 데이터시트에 표시됩니다.
- ▶ 센서와 물체 사이의 가능한 한 작은 거리 ⑤ 선택
- > 거리가 최대한으로 짧으면 물체는 최대 해상도로 감지됩니다.
- ▶ 설치 장소에서 강한 주변 광선 및 햇빛을 피하십시오.
- > 8 klx (태양 스펙트럼 포함) 이상의 외부광선은 측정 오류의 원인이 됩니다. 사실 800 그리고 900nm의 적외선 구성요소만이 문제됩니다.
- ▶ 심하게 오염된 환경에 설치하지 마십시오.
- > 심하게 오염된 환경에서는 센서가 아래쪽 방향 ①
- ▶ 센서 ① 과 물체 ③ 사이의 투명한 유리는 삼가하십시오
- > 투명 유리는 매우 깨끗한 유리 패널을 사용하더라도 광선의 일부를 반사합니다.

**!** 설명사항을 준수하지 않으면 측정 오류가 발생할 수 있습니다.

## 6.2 추가 센서 설치 안내

센서의 표면 온도는 작동 모드, 파라미터 선택 및 환경에 대한 센서의 열 노출에 따라 다릅니다.



센서가 다음 요구사항을 준수하는지 확인하십시오:

쉽게 접근 가능한 표면온도는 최대 25 °C로써 주변 온도보다 높을 수 있습니다 (IEC 61010-2-201 준수).

다음 다이어그램에는 설치 프로그램에 대한 참조로 일반적인 경고 제한이 포함됩니다.



다이어그램은 다음 작동 모드에 유효합니다.

- 낮음 [1 노출]
- 보통 [2 노출]
- 강함 [3 노출]

보통 및 강한 노출의 경우, 일반적인 경고 한계는 노출 시간의 합계를 통하여 결정되어야 합니다. 노출시간은 ifm Vision Assistant 소프트웨어에 표시됩니다.

경고 제한을 초과하면 설명사항 중 하나를 따르십시오.

- ▶ 표면 온도를 감소시키십시오. (→ 6.2.3)
- ▶ 센서가 열원으로부터 보호되는 위치나 하우징에 설치하되 센서 주위의 공기 순환이 유지할 수 있어야 합니다.
- > 센서 표면 온도의 상승을 방지해야 합니다.

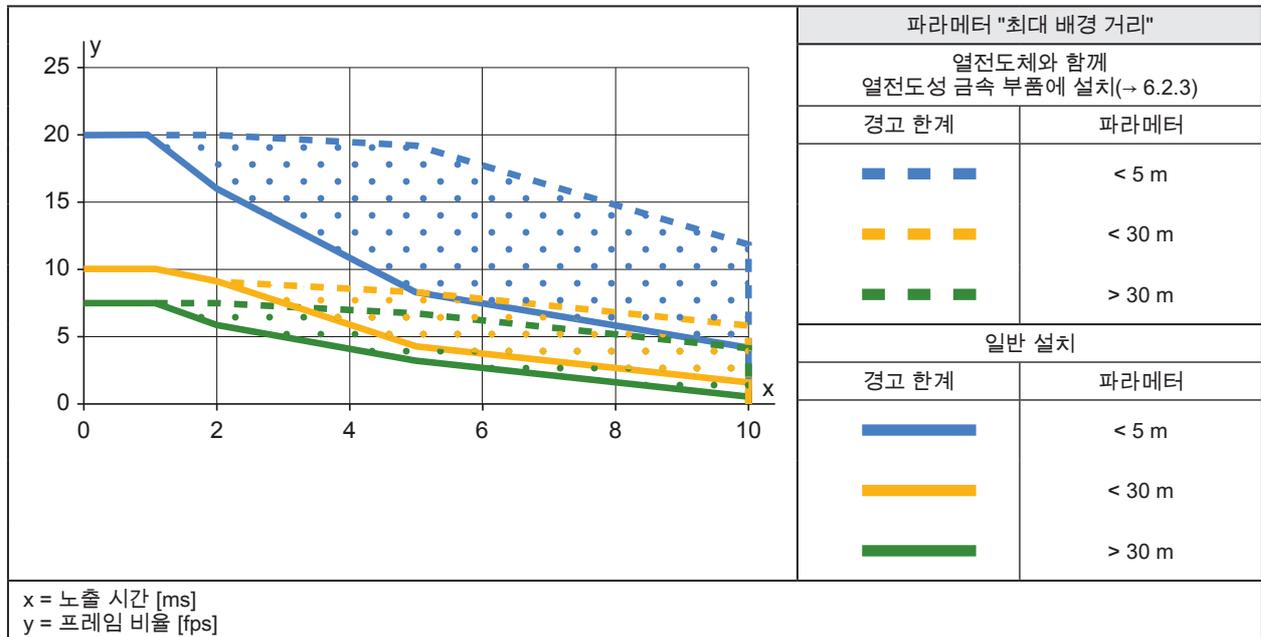


"최대 배경 거리" 파라미터는 ifm Vision Assistant에서 설정됩니다. 다이어그램에서 파라미터의 경고 한계는 파선 (dashed line)과 연속선 (continuous line)으로 표시됩니다.

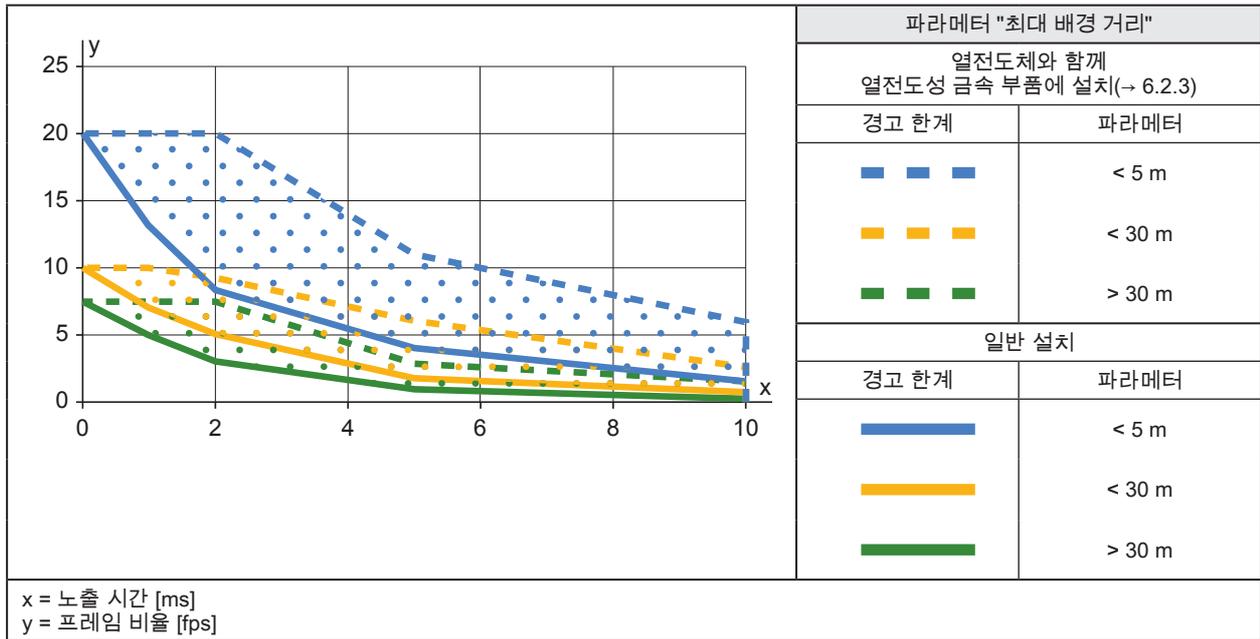
센서가 점선으로된 영역 중 하나에 있으면 표면 온도를 감소시켜야 합니다(→ 6.2.3). 방열 설치에도 불구하고 경고 한계가 초과되면, 추가로 접점 보호 장치를 장착 할 수 있습니다.

일반 설치의 경우, 일반적인 경고 제한이 초과되지 않으면 조치를 취할 필요가 없습니다.

### 6.2.1 O3D300 / O3D302에 대한 일반적인 경고 한계



### 6.2.2 O3D310 / O3D312에 대한 일반적인 경고 한계



### 6.2.3 표면 온도 감소

다음 조치를 통해 표면 온도를 낮출 수 있습니다.

- ▶ 열 전도성 금속 부품에 센서를 장착하십시오.
- > 센서와 금속 부품이 대면 접촉하면 열 방출이 증가합니다 (예: 알루미늄).
- ▶ 센서를 금속 부품에 설치할 때 열전도체를 사용하십시오.
- > 열전도 효과는 열전도체에 의해 증가됩니다. 열전도체는 액세서리로 제공됩니다 (→ 6.4).
- ▶ 센서 주변의 장애물을 감소시킵니다. 센서 주변에 설치된 물체의 밀도를 감소시킵니다.
- > 센서 주변의 장애물과 높은 설치 밀도는 대류 (공기 이동)에 부정적인 영향을 줄 수 있습니다.
- ▶ 센서에 하나 또는 두개의 열 흡수장치를 장착하십시오.
- > 열 흡수장치는 센서의 표면을 증가시켜 표면온도를 낮춥니다. 열 흡수장치는 액세서리로 제공됩니다 (→ 6.4).
- ▶ 노출 시간, 프레임 속도 또는 최대 배경거리를 감소시킵니다.
- > 사용된 동작모드 및 파라미터가 표면 온도를 높일 수 있습니다.

### 6.3 센서 설치

센서를 설치할 때 다음 지시 사항에 유의하십시오.

- ▶ 2x M5 나사 또는 마운트 세트를 사용하여 센서를 설치하십시오.
- > M5 나사의 보어 치수는 데이터 시트에 표시되어 있습니다.
- > 마운팅 세트는 액세서리로 제공됩니다 (→ 6.4).
- ▶ 센서에 연결된 모든 케이블에 스트레인 릴리프 (strain reliefs) 장치를 사용하십시오.

O3D300 및 O3D310을 설치할 때 다음 지시사항을 준수하십시오:

- ▶ 스크류 드라이버로 초점 조정나사에 접근 할 수 있도록 센서를 장착하십시오.
- > 초점 조정나사의 위치는 도면에 표시됩니다 (→ 12).



디바이스가 습기영역에서 영구적으로 사용되는 경우, M12 산업용 이더넷 케이블 (예: E11898)의 M12 너트가 부식 될 수 있습니다. 습기영역에서 영구적으로 사용하려면, 고급 스텐레스 스틸 너트를 보유한 케이블을 사용하십시오.

### 6.4 마운팅 액세서리

위치 및 설치 타입에 따라 다음 장착용 액세서리 사용이 가능합니다.

제품 번호	설명
E3D301	스마트 카메라 마운팅 세트
E3D302	스마트 카메라 냉각 부품
E3D303	스마트 카메라 열전도체
E3D304	2x 스마트 카메라 냉각 부품



액세서리에 관한 상세정보:

[www.ifm.com](http://www.ifm.com) → 데이터시트 찾기 → 예: O3D303 → 액세서리

## 7. 전기적 연결

전기적 연결 이전에 다음 지시사항에 유의하십시오.

**참고**

본 제품의 설치에 반드시 전문직업교육을 받은 전문가에 의해 이루어져야 합니다. 데이터시트의 전기적 데이터에 유의하십시오.

보호 등급 III (PC III)의 센서

전기 공급은 반드시 PELV 회로를 통해서 이루어져야 합니다.

전기적 공급은 UL61010-1, 챕터 9.4 - 제한된 에너지에 상응되어야 합니다:

과전류 방지 디바이스는 120초 동안 6.6A 전류를 차단해야 합니다. 과전류 방지 디바이스의 정확한 크기를 얻으려면 센서의 기술 데이터와 배선을 고려해야 합니다.

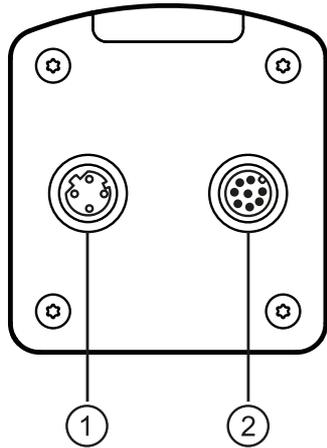
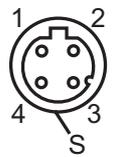
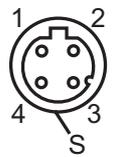
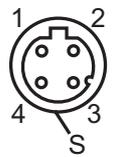
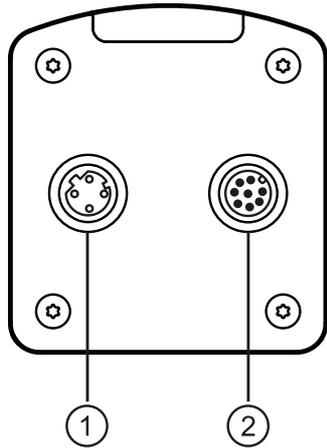
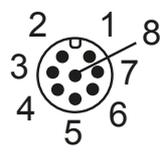
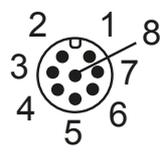
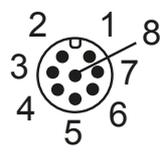
외부 회로의 분리는 UL61010-2-201 (그림 102)을 준수해야 합니다.

케이블 길이가 30m 이상인 경우 IEC 6100-4-5에 서지 전압에 대한 추가 방지용 장치를 사용하십시오.

센서를 연결하기 전에 설비 전원을 차단하십시오.

**!** cULus의 유효 범위는 다음과 같습니다:  
필드 배선 터미널에 연결되는 케이블의 최소 온도 등급: 70 °C.

### 7.1 배선

	① 이더넷																			
	<table style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;">M12 소켓, D 코드화, 4 극</td> </tr> <tr> <td style="text-align: center;">  </td> <td style="padding-left: 10px;"> <table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 20px;">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>섀드</td></tr> </table> </td> </tr> </table>	M12 소켓, D 코드화, 4 극			<table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 20px;">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>섀드</td></tr> </table>	1	TD +	2	RD +	3	TD -	4	RD -	S	섀드					
M12 소켓, D 코드화, 4 극																				
	<table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 20px;">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>섀드</td></tr> </table>	1	TD +	2	RD +	3	TD -	4	RD -	S	섀드									
1	TD +																			
2	RD +																			
3	TD -																			
4	RD -																			
S	섀드																			
	②																			
	<table style="width: 100%; border-collapse: collapse;"> <tr> <td colspan="2" style="text-align: center;">M12 커넥터, A 코드화, 8 극</td> </tr> <tr> <td style="text-align: center;">  </td> <td style="padding-left: 10px;"> <table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 20px;">1</td><td>U+</td></tr> <tr><td>2</td><td>트리거 입력</td></tr> <tr><td>3</td><td>GND</td></tr> <tr><td>4</td><td>스위칭 출력 1 (디지털 또는 아날로그)</td></tr> <tr><td>5</td><td>스위칭 출력 3 준비</td></tr> <tr><td>6</td><td>스위칭 출력 2 (디지털)</td></tr> <tr><td>7</td><td>스위칭 출력 1</td></tr> <tr><td>8</td><td>스위칭 출력 2</td></tr> </table> </td> </tr> </table>	M12 커넥터, A 코드화, 8 극			<table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 20px;">1</td><td>U+</td></tr> <tr><td>2</td><td>트리거 입력</td></tr> <tr><td>3</td><td>GND</td></tr> <tr><td>4</td><td>스위칭 출력 1 (디지털 또는 아날로그)</td></tr> <tr><td>5</td><td>스위칭 출력 3 준비</td></tr> <tr><td>6</td><td>스위칭 출력 2 (디지털)</td></tr> <tr><td>7</td><td>스위칭 출력 1</td></tr> <tr><td>8</td><td>스위칭 출력 2</td></tr> </table>	1	U+	2	트리거 입력	3	GND	4	스위칭 출력 1 (디지털 또는 아날로그)	5	스위칭 출력 3 준비	6	스위칭 출력 2 (디지털)	7	스위칭 출력 1	8
M12 커넥터, A 코드화, 8 극																				
	<table style="width: 100%; border-collapse: collapse;"> <tr><td style="width: 20px;">1</td><td>U+</td></tr> <tr><td>2</td><td>트리거 입력</td></tr> <tr><td>3</td><td>GND</td></tr> <tr><td>4</td><td>스위칭 출력 1 (디지털 또는 아날로그)</td></tr> <tr><td>5</td><td>스위칭 출력 3 준비</td></tr> <tr><td>6</td><td>스위칭 출력 2 (디지털)</td></tr> <tr><td>7</td><td>스위칭 출력 1</td></tr> <tr><td>8</td><td>스위칭 출력 2</td></tr> </table>	1	U+	2	트리거 입력	3	GND	4	스위칭 출력 1 (디지털 또는 아날로그)	5	스위칭 출력 3 준비	6	스위칭 출력 2 (디지털)	7	스위칭 출력 1	8	스위칭 출력 2			
1	U+																			
2	트리거 입력																			
3	GND																			
4	스위칭 출력 1 (디지털 또는 아날로그)																			
5	스위칭 출력 3 준비																			
6	스위칭 출력 2 (디지털)																			
7	스위칭 출력 1																			
8	스위칭 출력 2																			

**!** 보호캡 (E73004)으로 사용하지 않는 이더넷 접속을 커버하십시오.  
조임토크 0.6...0.8 Nm

**i** 스위칭 입력 및 출력의 동작은 소프트웨어 ifm Vision Assistant를 사용하여 설정할 수 있습니다. PNP 또는 NPN 설정은 항상 모든 스위칭 입력 및 출력에 적용됩니다.

액추에이터 및 센서를 설치할 때 세팅이 올바른지 확인하십시오 (예: 트리거링을 위한 포토 센서).

스위칭 출력은 조정 가능한 시간 후에 그의 스위칭 신호를 리셋하는 펄스 출력으로 또한 작동 될 수 있습니다.

아날로그 출력은 GND에 대해 전류 / 전압을 제공합니다.

### 7.1.1 핀 1 / 3 (24 V / GND)

허용되는 전압범위는 센서의 데이터시트에 표시되어 있습니다.

### 7.1.2 핀 2 (트리거 입력)

센서의 이미지 캡처는 트리거 입력을 통해 스위칭 신호로 트리거 될 수 있습니다.

다음 트리거 에지가 사용될 수 있습니다:

- 하강 에지가 이미지 캡처를 트리거합니다.
- 상승 에지가 이미지 캡처를 트리거합니다.
- 하강 및 상승 에지가 이미지 캡처를 트리거합니다.



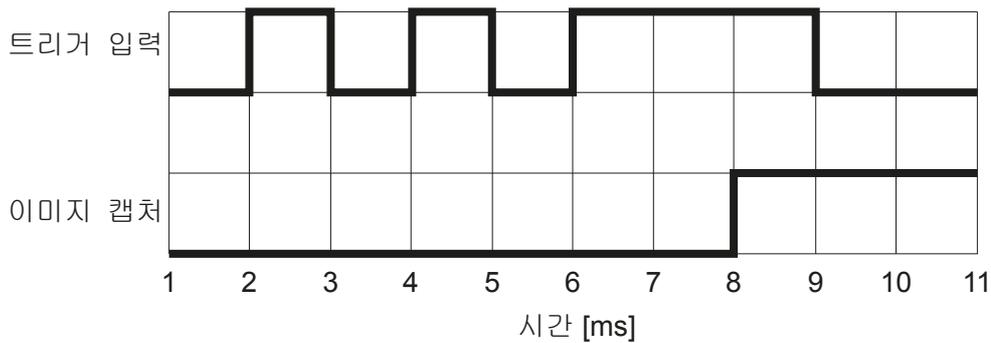
센서를 트리거 할 수 있는 더 많은 가능성:

- 프로세스 인터페이스 명령 (→ 13.2)
- 고정된 프레임 속도로 연속 이미지 캡처



트리거 입력은 내부적으로 디 바운스됩니다. 전기적 설치에 따라 트리거 와이어의 디바운싱이 요구되지 않습니다.

내부 디바운싱은 몇 개의 짧은 펄스가 트리거되는 것을 방지합니다. 펄스가 트리거로 인식 되려면 최소 2 ms의 시간이 필요합니다.



### 7.1.3 핀 4 / 5 / 6 (스위칭 출력)

스위칭 출력 1에서 3은 서로 다른 센서 상태를 제공합니다. 스위칭 출력은 센서 상태 뿐만 아니라 어플리케이션을 해결하는 데 필요한 참조값을 제공할 수 있습니다.

스위칭 출력 1에서 3의 전기적 사양은 데이터 시트에 표시되어 있습니다.

스위칭 출력 3은 센서 상태 "트리거 준비"를 기본 세팅으로 제공합니다.



"스위칭된 스위칭 출력"은 해당 센서 상태가 발생했음을 의미합니다.

센서 상태는 세팅에 따라 다음 값 중 하나를 보유할 수 있습니다:

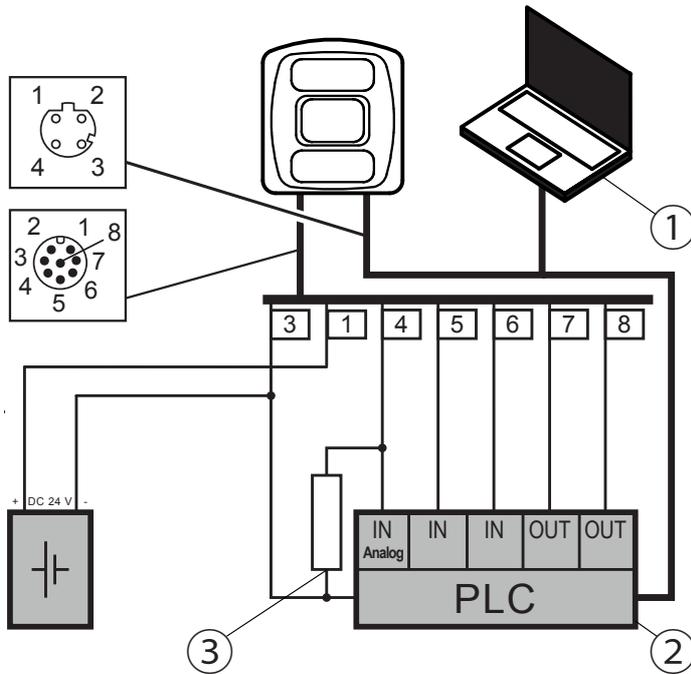
- "트리거 준비 완료"  
센서는 새로운 이미지를 캡처 할 수 있음을 신호합니다. 이 센서 상태 트리거 동작만으로 처리됩니다. 연속 이미지 캡처의 경우 상태 "트리거 준비 됨"이 출력되지 않습니다.
- "이미지 캡처 완료"  
센서는 이미지 캡처가 완료되었음을 신호해줍니다. 센서 상태는 캐스캐이딩 센서에 사용될 수 있습니다.
- "평가 완료됨"  
센서는 이미지 처리가 완료되었음을 신호해줍니다. 그 순간에 스위칭 출력은 이미 업데이트 됩니다. 이미지 데이터는 이더넷을 통하여 전송됩니다.
- "오류"  
센서가 내부 오류를 신호합니다. 오류에 대한 자세한 정보는 이더넷을 통해 요청될 수 있습니다.

### 7.1.4 핀 4 (아날로그 출력)

스위칭 출력 1 / 아날로그 출력은 스위칭 출력 또는 아날로그 전류 출력 (4-20 mA) / 아날로그 전압 출력 (0-10 V)으로 사용될 수 있습니다.

아날로그 전류 출력은 아날로그 전압 출력보다 전송 신뢰성이 높습니다. 아날로그 전류 출력은 케이블 길이와 무관하며, 산업용 컨트롤러에 대한 신호 품질을 향상시킵니다.

산업용 컨트롤러에서 아날로그 전류는 GND에 대한 부하 저항기를 통해 아날로그 전압으로 전환됩니다. 부하 저항기는 데이터시트의 표시에 따라 선택됩니다. 디바이스에서 더 낮은 열 발달로 인하여, 고저항의 부하 저항기가 저저항의 부하 저항기보다 선호됩니다.



- ① 노트북 (파라미터 세팅)
- ② 산업용 컨트롤러 (평가 / 트리거)
- ③ 부하 저항기

ifm Vision Assistant 소프트웨어를 사용하면 아날로그 출력의 시작값 (4 mA / 0 V)과 종료값 (20 mA / 10 V)에 각각 하나의 프로세스 값을 할당 할 수 있습니다.

### 7.1.5 핀 7 / 8 (스위칭 입력)

스위칭 입력은 다음 기능을 제공합니다:

- 활성화된 어플리케이션 선택

-  기능의 다른 파라미터 세팅은 프로그래밍 매뉴얼에 표시되어 있습니다.
-  스위칭 입력 1과 2의 전기적 데이터는 데이터시트에 표시되어 있습니다.

## 7.2 배선 사례

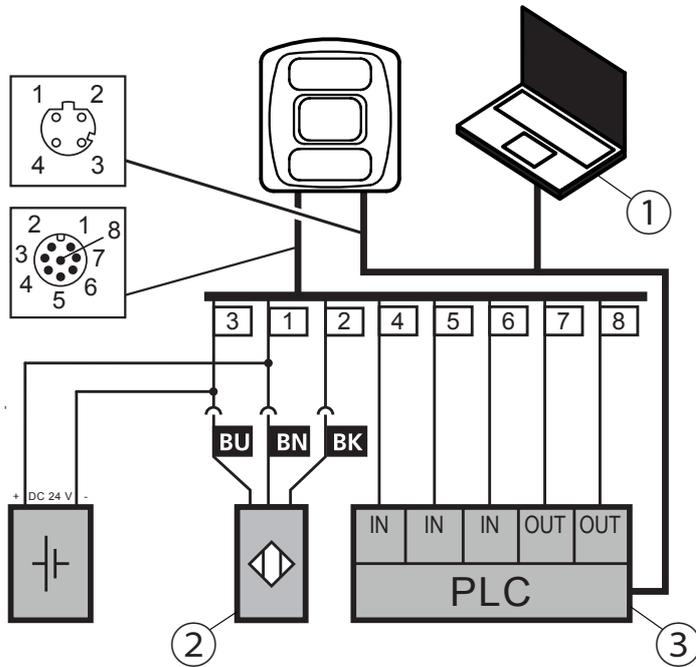
센서의 배선 사례는 아래와 같습니다.

### 7.2.1 근접센서로 이미지 캡처 트리거

센서는 외부에서 트리거될 수 있습니다.

- 이더넷을 통한
- 트리거 입력에 연결된 근접센서를 통한

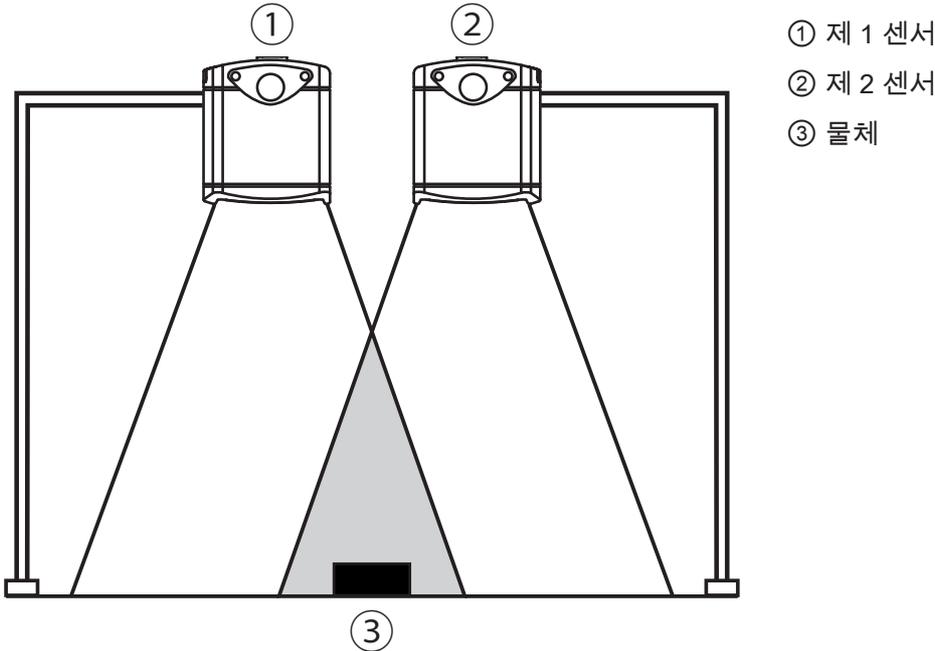
다음 그림은 근접센서를 보유한 배선을 보여줍니다.



- ① 노트북 (파라미터 세팅)
- ② 근접센서
- ③ 산업용 컨트롤러 (평가 / 트리거)

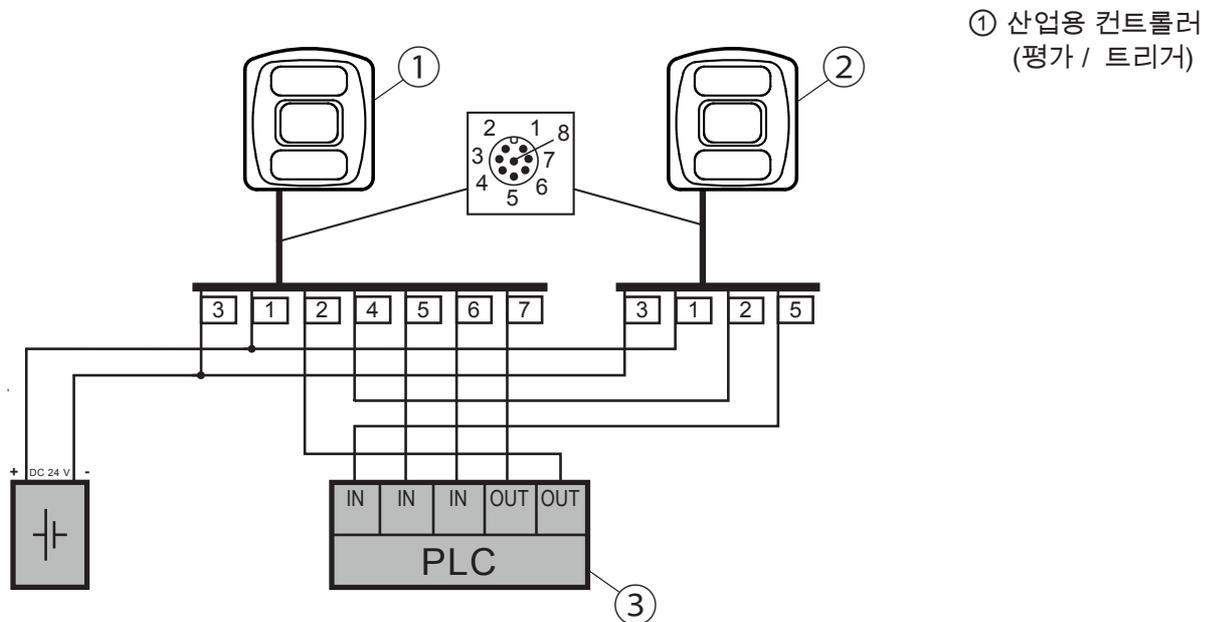
### 7.2.2 여러 센서를 나란히 설치합니다.

나란히 근접 설치된 센서는 동시 노출로 인해 측정 오류가 발생할 수 있습니다.



측정오류는 다음 두가지 방법으로 방지될 수 있습니다:

- HW 트리거를 통한 캐스캐이딩 센서  
 캐스캐이딩 도중에 컨트롤러가 센서①의 이미지 캡처를 트리거합니다 (아래 그림 참조). 이미지 캡처가 완료되면, 센서 ①이 자동으로 센서 ②를 트리거합니다. 동시에 센서 ①의 핀 4는 센서 상태 "이미지 캡처 완료"를 제공합니다. 센서 ②는 시퀀스 끝을 산업용 컨트롤러③에 신호해 줍니다.



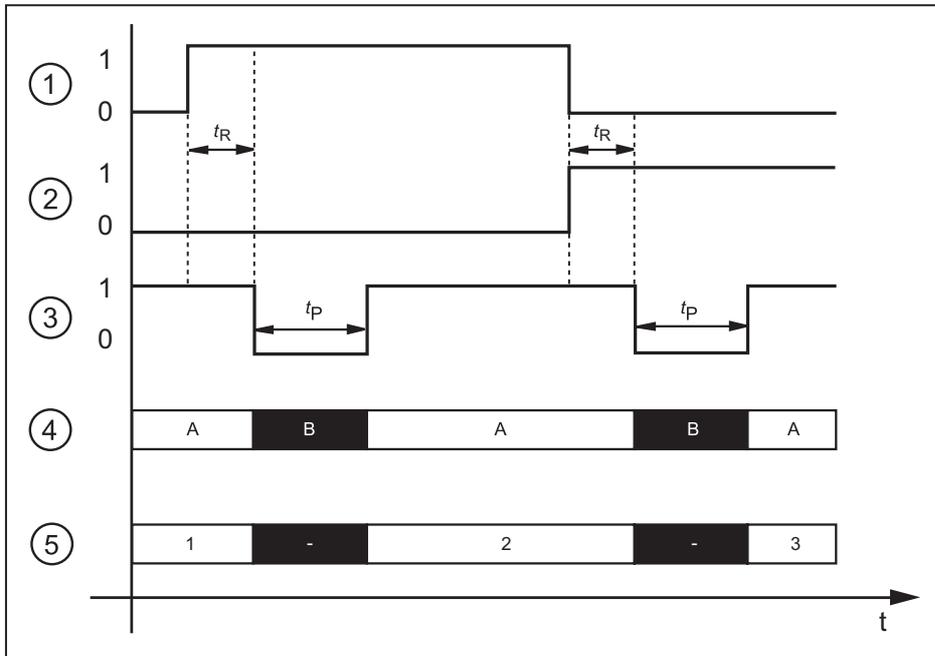
- 다른 주파수 채널을 사용하십시오.  
 ifm Vision Assistant 소프트웨어를 통하여 각각의 센서에 자체 주파수 채널이 할당될 수 있습니다. 서로 다른 주파수 채널은 측정 에러의 발생을 감소시켜 줍니다.

 ifm Vision Assistant 소프트웨어는 웹사이트에서 무료로 제공됩니다:  
[www.ifm.com](http://www.ifm.com) → 고객지원 → 다운로드 → 산업용 이미징

### 7.3 어플리케이션의 정적 선택

최대 32 개까지의 다양한 검사업무가 센서에 저장될 수 있습니다. 해당 유닛 구성으로 첫 번째 네개의 어플리케이션은 2개의 스위칭 입력을 통하여 선택될 수 있습니다.

입력 2	입력 1	어플리케이션 번호
0	0	1
0	1	2
1	0	3
1	1	4



사례: 선택 어플리케이션 1 → 어플리케이션 2 → 어플리케이션 3

①	스위칭 입력 1 = 0 → 1 → 0
②	스위칭 입력 2 = 0 → 0 → 1
③	출력 READY
④	트리거 입력
	A: 트리거 허용됨 B: 트리거 허용되지 않음
⑤	활성화된 어플리케이션의 ID 번호

어플리케이션 선택을 위하여 모니터링 시간  $t_R$  그리고 사용되지 않는 시간 트리거  $t_P$ 가 고려되어야 합니다.

모니터링 시간  $t_R$ : 양쪽 스위칭 입력에 대한 상태의 에지변화 이후 20 ms 동안 안전하게 머무르면, 어플리케이션 선택이 비로소 시작됩니다.

트리거 불가능 시간  $t_P$ : 트리거 입력은 어플리케이션 이 선택되는 동안은 불가능합니다.

불가능 시간은 다음에 의존합니다:

- 센서에 어플리케이션 수
- 활성화 되는 어플리케이션에서의 모델 수



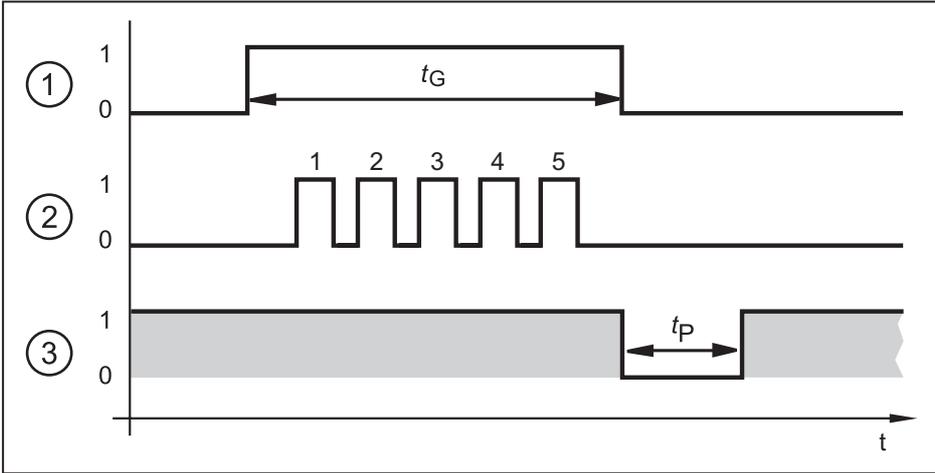
상기 그림은 PNP 출력 로직 (공장 설정)을 보여줍니다. NPN 출력 로직 동작은 PNP 출력 로직의 동작과 반대입니다.

- PNP 출력 로직: 높은 신호 (1)의 경우, 전압이 인가됩니다.
- NPN 출력 로직: 낮은 신호 (0)의 경우, 전압이 인가됩니다.

어플리케이션 선택의 구성에 관한 상세정보는 센서의 프로그래밍 매뉴얼을 참조하십시오.  
[www.ifm.com](http://www.ifm.com) → 데이터 시트 찾기 → O3D300 → 사용 설명서

### 7.4 어플리케이션의 펄스 컨트롤된 선택

정적선택에 대한 대안으로 어플리케이션의 선택은 펄스 컨트롤됩니다.



①	게이트 시그널, 스위칭 입력 1 = 0 → 1 → 0 ( $t_G$ = 시그널 활성화)
②	펄스 시그널, 스위칭 입력 2 또는 트리거 입력 = 0 → 5 펄스 → 0
③	준비 (READY) 출력

스위칭 입력 1 (게이트 시그널)에 활성화된 시그널이 있으면, 센서는 들어오는 펄스를 카운팅하고 해당 어플리케이션을 활성화합니다.

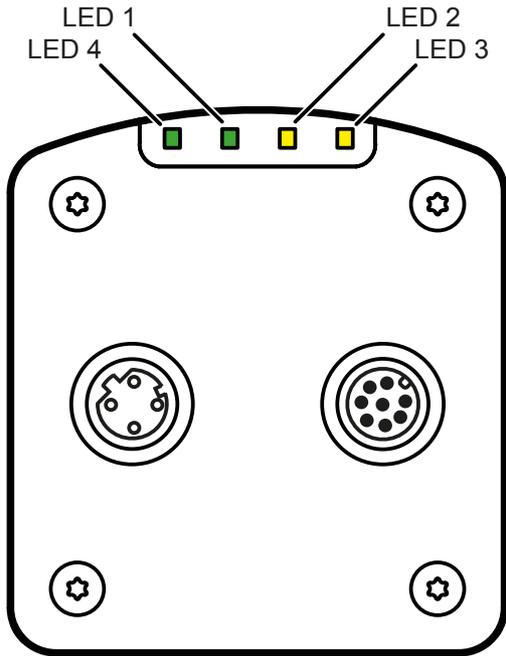
펄스 수 = 어플리케이션의 ID 번호

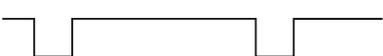
스위칭 입력 2 또는 센서의 트리거 입력은 펄스 입력으로 사용될 수 있습니다.

어플리케이션 선택의 구성에 관한 상세정보는 센서의 프로그래밍 매뉴얼을 참조하십시오.  
[www.ifm.com](http://www.ifm.com) → 데이터 시트 찾기 → O3D300 → 사용 설명서

## 8. 표시기

LED 표시기 1 - 4를 통하여 센서는 현재 동작상태를 신호해 줍니다.



LED 4 (이더넷)	LED 1 (전원)	LED 2 (Out 1)	LED 3 (Out 2)	설명
	점등됨:			센서는 작동 준비상태이며, 공급전압이 적용됨
	0.5 Hz 에서 점멸됨			파라미터 설정 또는 파라미터 세팅이 센서에 로딩되지 않음 On  Off
	0.5 Hz 에서 2번 점멸됨			센서가 파라미터 설정 모드에 있음 On  Off
		점등됨:		스위칭 출력 1이 스위칭됨
		8 Hz에서 점멸됨		스위칭 출력 1이 쇼트됨
			점등됨:	스위칭 출력 2가 스위칭됨
			8 Hz에서 점멸됨	스위칭 출력 2가 쇼트됨
점등됨:				이더넷이 연결됨
점멸됨				이더넷 전송 데이터
off				이더넷이 접속되지 않음
		8 Hz에서 점멸됨	8 Hz에서 점멸됨	센서가 내부 오류를 신호함
		2 Hz에서 점멸됨	2 Hz에서 점멸됨	센서가 수정 가능한 오류를 신호함 이더넷을 통하여 오류 정보 읽음이 가능함
		주행등 ⇒		센서 부팅
		주행등 ⇐		센서가 펌웨어 업데이트를 수행함

## 9. 셋업

전원이 켜지면 센서가 작동합니다. 15 초 후 센서는 저장된 어플리케이션이 실행되는 평가모드입니다. 표시기는 전류 작동상태를 나타냅니다 (→ 7.3).

**i** 최대 32 개까지의 어플리케이션이 센서에 저장될 수 있습니다. 어플리케이션을 다른 방법으로 활성화 할 수 있습니다:

- ifm Vision Assistant 소프트웨어
- 프로세스 인터페이스 명령
- 스위칭 입력 1과 2
- 스위칭 입력 1과 트리거 입력

### 9.1 센서의 파라미터 설정

ifm Vision Assistant 소프트웨어를 사용하여 센서가 설정됩니다 (→ 프로그래밍 매뉴얼 참조).

**i** 소프트웨어 ifm Vision Assistant 및 센서의 측정 원리에 대한 자세한 정보는 소프트웨어 매뉴얼에 설명되어 있습니다.

ifm Vision Assistant 소프트웨어는 웹사이트에서 무료로 제공됩니다:

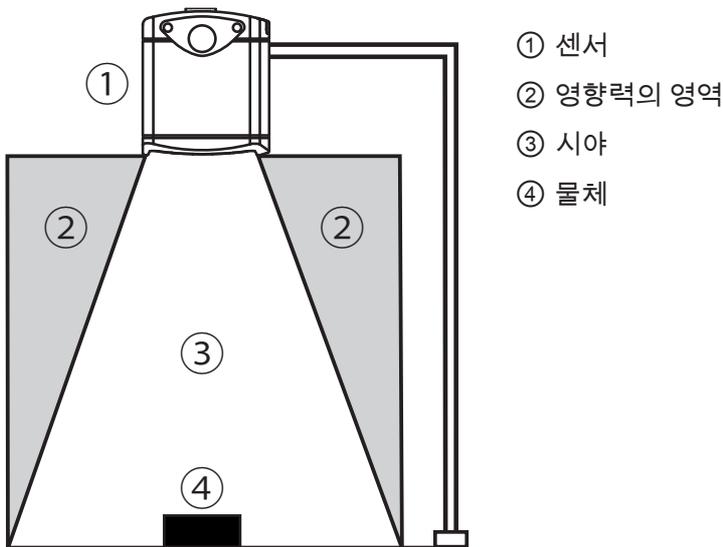
[www.ifm.com](http://www.ifm.com) → 고객지원 → 다운로드 → 산업용 이미징

소프트웨어 매뉴얼은 당사 웹사이트에서 사용할 수 있습니다.

[www.ifm.com](http://www.ifm.com) → 데이터시트 찾기 → 예: O3D303 → 사용 설명서

### 9.2 물체감지

물체의 높은 탐지율로 이어지는 조건은 아래에 설명되어 있습니다.



다음 조건이 충족되면 물체④가 최상으로 검출됩니다:

- 물체가 시야 ③에 배치됩니다.
- 물체는 센서 ①에서 가장 가깝게 보입니다.
- 영향을 주는 영역 ②는 물체 (장애물 등)로부터 명확합니다.
- 센서의 렌즈는 오염되지 않아야 합니다.

**!** 조건이 충족되지 않으면, 측정 오류가 발생할 수 있습니다.

### 9.3 프로세스 값 전송

#### 9.3.1 EtherNet/IP를 통한 완전성 모니터링의 프로세스 값 전송

디바이스는 EtherNet/IP 프로토콜을 통하여 프로세스 값을 PLC에 전송할 수 있습니다. 프로세스 값은 ifm Vision Assistant에 출력 문자열로 다음과 같이 표시됩니다:

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

출력 문자열에서 프로세스 값은 세미콜론으로 구분됩니다. 출력 문자열은 표시된 순서대로 PLC에 전송됩니다.

 출력 문자열을 PLC로 전송할때 다음 사항에 유의하십시오:

- 0 - 7 Bytes는 출력 문자열의 일부입니다. 그들은 ifm Vision Assistant에 표시되지 않습니다 (상기 스크린샷 참조).
- 출력 문자열에 있는 세미콜론 ";" 은 전송되지 않습니다.
- 플롯 값은 전송 전에 바이너리 16 비트 정수로 전환됩니다.
- 모든 숫자 값은 전송 전에 2 진수 16 비트 정수로 변환됩니다.

출력 문자열은 다음과 같이 구성됩니다 (데이터 타입: SINT):

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

Byte no.	데이터	코드화	프로세스 값	유닛	설명	코멘트
0	2#0000_0000	바이너리	1.5		중복 명령어	1.5 비트는 성공적인 트리거 명령을 보여줍니다.
1	2#0010_0000	바이너리				
2	2#0000_0000	십진법			동기식 / 비동기식 메시지 식별	
3	2#0000_0000	십진법				
4	30	십진법	30		메시지 카운터	<ul style="list-style-type: none"> <li>• 디바이스가 30개의 메시지를 받았습니다.</li> <li>• 각 작업 (트리거, 메시지 전송 등)이 있을 때마다 1 씩 증가합니다.</li> </ul>
5	0	십진법				
6	0	십진법			예약됨	
7	0	십진법				
8	s	ASCII	star		스트링 시작	
9	t	ASCII				
10	a	ASCII				
11	r	ASCII				
12	0	십진법	0		모든 ROIs 상태 (0 = 불량, 1 = 양호함)	완전성 모니터링의 상태를 보여줌
13	0	십진법				
14	0	십진법	0		ROI ID	활성화된 위치 조정으로 14 및 15 바이트가 사용됩니다. 0 = 위치가 조정되지 않음 1 = 위치가 조정됨 다음의 모든 데이터는 2 바이트씩 이동됩니다; 예를 들어 일차 ROI ID는 16 및 17 바이트로 시작합니다.
15	0	십진법				



Byte no.	데이터	코드화	프로세스 값	유닛	설명	코멘트
16	0	십진법	0		ROI 상태	ROI 상태: 0 = 양호함 1 = 참조 레벨이 터치되지 않음 2 = 터치 실패됨 3 = 참조 레벨 무효 4 = 유효 픽셀 없음 5 = 참조 레벨에 유효한 픽셀이 없음 6 = 오버필 7 = 언더필
17	0	십진법				
18	0	십진법	0	mm	ROI 값	
19	0	십진법				
20	1	십진법	1		ROI ID	
21	0	십진법				
22	7	십진법	7		ROI 상태	
23	0	십진법				
24	-67	십진법	-67	mm	ROI 값	
25	-1	십진법				
26	2	십진법	2		ROI ID	
27	0	십진법				
28	6	십진법	6		ROI 상태	
29	0	십진법				
30	14	십진법	14	mm	ROI 값	
31	0	십진법				
32	3	십진법	3		ROI ID	
33	0	십진법				
34	0	십진법	0		ROI 상태	
35	0	십진법				
36	0	십진법	0	mm	ROI 값	
37	0	십진법				
38	s	ASCII	중지		스트링 종료	
39	t	ASCII				
40	o	ASCII				
41	p	ASCII				

 명령을 잘못 실행하면 다음 상태가 됩니다.

- 오류 bit = 1
- 중복된 명령어가 표시되었습니다.
- 비동기 메시지 bit = 0
- 비동기 메시지 식별 = 0
- 메시지 카운터가 1 씩 증가합니다.

### 9.3.2 TCP/IP를 통한 완전성 모니터링의 프로세스 값 전송

디바이스는 TCP/IP 프로토콜을 통하여 프로세스 값을 PLC에 전송할 수 있습니다. 프로세스 값은 ifm Vision Assistant에 출력 문자열로 다음과 같이 표시됩니다:

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

출력 문자열에서 프로세스 값은 세미콜론으로 구분됩니다. 출력 문자열은 표시된 순서대로 PLC에 전송됩니다.



출력 문자열을 PLC로 전송할때 다음 사항에 유의하십시오:

- 출력 문자열에 있는 세미콜론 ";" 은 전송되지 않습니다.
- 모든 절대값은 전송 전에 바이너리 16 비트 정수로 변환됩니다.

출력 문자열은 다음과 같이 구성됩니다 (데이터 타입: ASCII):

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

프로세스 값	유닛	설명
star		스트링 시작
0		모든 ROIs 상태 (0 = 불량, 1 = 양호함)
00 0 +0.000	m	ROI ID ROI 상태 ROI 값
01 7 -0.068	m	ROI ID ROI 상태 ROI 값
02 6 +0.013	m	ROI ID ROI 상태 ROI 값
03 0 +0.001	m	ROI ID ROI 상태 ROI 값
중지		스트링 종료

ROI 상태:  
 0 = 양호함  
 1 = 참조 레벨이 터치되지 않음  
 2 = 티칭이 실패됨  
 3 = 참조 레벨 무효  
 4 = 유효 픽셀 없음  
 5 = 참조 레벨에 유효한 픽셀이 없음  
 6 = 오버필  
 7 = 언더필

### 9.3.3 EtherNet/IP를 통하여 물체 치수의 프로세스 값 전송

디바이스는 EtherNet/IP 프로토콜을 통하여 프로세스 값을 PLC에 전송할 수 있습니다. 프로세스 값은 ifm Vision Assistant에 출력 문자열로 다음과 같이 표시됩니다:

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

출력 문자열에서 프로세스 값은 세미콜론으로 구분됩니다. 출력 문자열은 표시된 순서대로 PLC에 전송됩니다.



출력 문자열을 PLC로 전송할때 다음 사항에 유의하십시오:

- 출력 문자열은 조정 가능합니다. 전송되는 프로세스 값은 ifm Vision Assistant에서 설정할 수 있습니다.
- 0 - 7 Bytes는 출력 문자열의 일부입니다. 그들은 ifm Vision Assistant에 표시되지 않습니다 (상기 스크린샷 참조).
- 출력 문자열에 있는 세미콜론 ";" 은 전송되지 않습니다.
- 플롯 값은 전송 전에 바이너리 16 비트 정수로 전환됩니다.
- 모든 절대값은 전송 전에 바이너리 16 비트 정수로 전환됩니다.

출력 문자열은 다음과 같이 구성됩니다 (데이터 타입: SINT):

```
star;1;0.104;0.0088;0.109;+0.021;-0.011;+0.389;158;097;094;097;stop
```

Byte 번호	데이터	코딩	프로세스 값	유닛	설명	코멘트
0	2#0000_0000	바이너리	1.5		중복 명령어	1.5 비트는 성공적인 트리거 명령을 보여줍니다.
1	2#0010_0000	바이너리				
2	2#0000_0000	바이너리	3		동기식 / 비동기식 메시지 식별	
3	2#0000_0000	바이너리				
4	2#0000_0011	바이너리				
5	2#0000_0000	바이너리			메시지 카운터	<ul style="list-style-type: none"> <li>● 디바이스가 3 개의 메시지를 받았습니다.</li> <li>● 각 작업 (트리거, 메시지 전송 등) 이 있을 때마다 1 씩 증가합니다.</li> </ul>
6	2#0000_0000	바이너리			예약됨	
7	2#0000_0000	바이너리				
8	s	ASCII	star		스트링 시작	
9	t	ASCII				
10	a	ASCII				
11	r	ASCII				
12	2#0000_0001	바이너리	1		결과 bit	0 = 박스 발견하지 못함 1 = 박스 발견
13	2#0000_0000	바이너리				
14	104	십진법	104	mm	너비	
15	0	십진법				
16	88	십진법	88	mm	높이	
17	0	십진법				
18	108	십진법	108	mm	길이	
19	0	십진법				
20	21	십진법	21		x 좌표	
21	0	십진법				
22	-11	십진법	-11		y 좌표	
23	-1	십진법				
24	-124	십진법	388		z 좌표	
25	1	십진법				

Byte 번호	데이터	코딩	프로세스 값	유닛	설명	코멘트
26	-98	십진법	158		회전 정도	
27	0	십진법				
28	97	십진법	97		품질 너비	
29	0	십진법				
30	93	십진법	93		품질 높이	
31	0	십진법				
32	97	십진법	97		품질 길이	
33	0	십진법				
34	s	ASCII	중지		스트링 종료	
35	t	ASCII				
36	o	ASCII				
37	p	ASCII				



명령을 잘못 실행하면 다음 상태가 됩니다.

- 오류 bit = 1
- 중복된 명령어가 표시되었습니다.
- 비동기 메시지 bit = 0
- 비동기 메시지 식별 = 0
- 메시지 카운터가 1 씩 증가합니다.

### 9.3.4 TCP/IP를 통하여 물체 치수의 프로세스 값 전송

디바이스는 TCP/IP 프로토콜을 통하여 프로세스 값을 PLC에 전송할 수 있습니다. 전송되는 프로세스 값은 ifm Vision Assistant에서 선택될 수 있습니다. 프로세스 값은 ifm Vision Assistant에 출력 문자열로 다음과 같이 표시됩니다:

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

출력 문자열에서 프로세스 값은 세미콜론으로 구분됩니다. 출력 문자열은 표시된 순서대로 PLC에 전송됩니다.

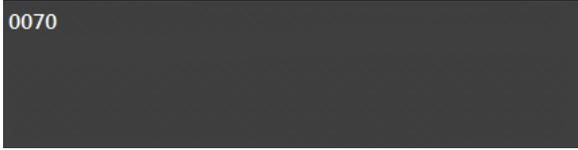
출력 문자열은 다음과 같이 구성됩니다 (데이터 타입: ASCII):

```
star;1;0.104;0.0088;0.109;+0.021;-0.011;+0.389;158;097;094;097;stop
```

프로세스 값	유닛	설명
star		스tring 시작
1		물체 찾을
0.104	m	너비
0.0088	m	높이
0.109	m	길이
+0.021		x 좌표
-0.011		y 좌표
+0.389		z 좌표
158		회전 정도
097		품질 너비
094		품질 높이
097		품질 길이
중지		스tring 종료

### 9.3.5 EtherNet/IP를 통한 레벨 모니터링의 프로세스 값 전송

디바이스는 EtherNet/IP 프로토콜을 통하여 프로세스 값을 PLC에 전송할 수 있습니다. 프로세스 값은 ifm Vision Assistant에 출력 문자열로 다음과 같이 표시됩니다:



출력 문자열은 표시된 순서대로 PLC에 전송됩니다.



출력 문자열을 PLC로 전송할때 다음 사항에 유의하십시오:

- 0 - 7 Bytes는 출력 문자열의 일부입니다. 그들은 ifm Vision Assistant에 표시되지 않습니다 (상기 스크린샷 참조).
- 출력 문자열에 있는 세미콜론 ";" 은 전송되지 않습니다.
- 플로트 값은 전송 전에 바이너리 16 비트 정수로 전환됩니다.
- 모든 절대값은 전송 전에 바이너리 16 비트 정수로 전환됩니다.

출력 문자열은 다음과 같이 구성됩니다 (데이터 타입: ASCII):

0070

Byte 번호	데이터	코딩	프로세스 값	유닛	설명	코멘트
0	2#0000_0000	바이너리	1.5		중복 명령어	1.5 비트는 성공적인 트리거 명령을 보여줍니다.
1	2#0010_0000	바이너리				
2	2#0000_0000	십진법	30		동기식 / 비동기식 메시지 식별	
3	2#0000_0000	십진법				
4	30	십진법				
5	0	십진법			메시지 카운터	<ul style="list-style-type: none"> <li>• 디바이스가 30 개의 메시지를 받았습니다.</li> <li>• 각 작업 (트리거, 메시지 전송 등)이 있을 때마다 1 씩 증가합니다.</li> </ul>
6	0	십진법			예약됨	
7	0	십진법			예약됨	
8	0	ASCII	0		모든 ROIs 상태 (0 = 불량, 1 = 양호함)	레벨 모니터링의 상태를 보여줌
9	0	ASCII	0		ROI ID	ROI Status: 0 = 양호함 6 = 오버필 7 = 언더필
10	0	ASCII	7		ROI 상태	
11	0	ASCII	7		ROI 상태	
12	0	ASCII	7		ROI 상태	
13	0	ASCII	0	mm	ROI 값	
14	0	ASCII	0			
15	0	ASCII				



명령을 잘못 실행하면 다음 상태가 됩니다.

- 오류 bit = 1
- 중복된 명령어가 표시되었습니다.
- 비동기 메시지 bit = 0
- 비동기 메시지 식별 = 0
- 메시지 카운터가 1 씩 증가합니다.

### 9.3.6 TCP/IP를 통한 레벨 모니터링의 프로세스 값 전송

디바이스는 TCP/IP 프로토콜을 통하여 프로세스 값을 PLC에 전송할 수 있습니다. 프로세스 값은 ifm Vision Assistant에 출력 문자열로 다음과 같이 표시됩니다:

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

출력 문자열에서 프로세스 값은 세미콜론으로 구분됩니다. 출력 문자열은 표시된 순서대로 PLC에 전송됩니다.

 출력 문자열을 PLC로 전송할때 다음 사항에 유의하십시오:

- 출력 문자열에 있는 세미콜론 ";" 은 전송되지 않습니다.
- 모든 절대값은 전송 전에 바이너리 16 비트 정수로 전환됩니다.

출력 문자열은 다음과 같이 구성됩니다 (데이터 타입: ASCII):

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

프로세스 값	유닛	설명
star		스tring 시작
0		모든 ROIs 상태 (0 = 불량, 1 = 양호함)
00		ROI ID
7		ROI 상태
+0.000	m	ROI Value
중지		스tring 종료

ROI 상태:  
0 = 양호함  
6 = 오버필  
7 = 언더필

## 10. 유지보수, 수리 및 폐기

다음 요구사항에 유의하십시오:

- ▶ 사용자가 유지보수 해야 하는 구성 요소가 센서내에 포함되지 않았으므로 하우징을 열지 마십시오. 고장난 센서의 수리는 반드시 제조업체에 문의하십시오.
- ▶ 국가 환경 규정에 따라 디바이스를 폐기하십시오.

### 10.1 세척

센서를 세척하기 전에 다음 지시 사항에 유의하십시오.

- ▶ 보풀이 없는 깨끗한 천을 사용하십시오.
- ▶ 유리 세척제를 세제로 사용하십시오.

 지침이 준수되지 않으면 렌즈에 흠집이 생겨 측정 오류가 발생할 수 있습니다.

### 10.2 펌웨어 업데이트

ifm Vision Assistant 소프트웨어를 사용하면 센서의 펌웨어가 업데이트될 수 있습니다.

 센서에 저장된 파라미터는 펌웨어 업데이트로 인해 손실됩니다. 펌웨어를 업데이트하기 전에 파라미터의 백업 복사본을 생성하십시오.

- ▶ 펌웨어를 업데이트하기 전에 파라미터를 내보냅니다.
- ▶ 펌웨어를 업데이트 한 후에 파라미터를 들여옵니다.

 펌웨어 업데이트는 당사 웹사이트에서 제공됩니다:  
[www.ifm.com](http://www.ifm.com) → 고객지원 → 다운로드 → 산업용 이미징

### 10.3 센서 대체

센서가 교체될 때 파라미터가 손실됩니다. 센서를 대체하기 전에 파라미터의 백업 복사본을 생성하십시오.

- ▶ 교체하기 전에 기존 센서의 파라미터를 내 보냅니다.
- ▶ 교체한 이후에 파라미터를 새로운 센서에 들여옵니다.

 파라미터의 내보내기 및 들여오기를 사용하면 동일한 파라미터로 여러개의 센서에 신속하게 제공될 수 있습니다.

## 11. 인증 / 표준

EC 적합성 선언은 다음에서 찾아 볼 수 있습니다:

[www.ifm.com](http://www.ifm.com) → 데이터시트 찾기 → 예: O3D303 → 인증서



## 13. Appendix

### 13.1 Process Interface

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

#### 13.1.1 Sending Commands

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

Structure of the protocol:

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

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

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

They are different protocol versions available:

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



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

Ticket numbers for asynchronous messages:

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

## Format of asynchronous notifications

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

Example for protocol version 3:

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

Result:

0010L000000045\r\n00100005000000:{"ID": 1034160761,"Index":1,"Name": "Pos 1"}\r\n

Explanation of the result:

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

## Asynchronous message IDs

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

### 13.1.2 Receiving Images

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

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

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

### 13.1.3 Image data

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

The header of each chunk contains different kinds of information. This information is separated into bytes. The information contains e.g. the kind of image which will be in the "PIXEL\_DATA" and the size of the chunk.

Chunk type

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

Offset	Name	Description	Size [byte]
0x0018	PIXEL_FORMAT	Pixel format	4
0x001C	TIME_STAMP	Time stamp in microseconds	4
0x0020	FRAME_COUNT	Frame counter	4
0x0024	PIXEL_DATA	The pixel data in the given type and dimension of the image. Padded to 4-byte boundary.	4

Available chunk types:

Constant	Value	Description
USERDATA	0	Undefined user data with arbitrary content
RADIAL_DISTANCE_IMAGE	100	<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:

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

## 13.1.4 Additional Information for CONFIDENCE\_IMAGE

Further information for the confidence image:

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

### 13.1.5 Configuration of PCIC Output

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

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

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

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

The layout software has the following main object properties:

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

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

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

Available values for the type property:

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

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

Common format properties:

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

Binary format properties:

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

ASCII format properties:

Format properties	Allowed values	Default
width	Output width. If the resulting value exceeds the width field the result will not be truncated.	0
fill	Fill character	" "
precision	Precision is the number of digits behind the decimalseparator.	6
displayformat	Fixed, scientific	Fixed
alignment	Left, right	Right
decimalseparator	7-bit characters for e.g. "."	."
base	Defines if the output should be: <ul style="list-style-type: none"> <li>• binary (2)</li> <li>• octal (8)</li> <li>• decimal (10)</li> <li>• hexadecimal (16)</li> </ul>	10

Example of a format configuration of the temperature (id: temp\_illu) element.

1. Illumination temperature like this "33,5\_\_":

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

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

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

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

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

The following element IDs are available:

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

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

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

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

ID	Description	Native data type
id	unique ID of the ROI within the Model	int32
procval	per ROI process value	float 32Bit
state	per ROI state ( if ROI procval is valid or not) <ul style="list-style-type: none"> <li>• ROI_PROCESS_VALUE_VALID = 0</li> <li>• ROI_PROCESS_VALUE_REFIMAGE_SET_NOT_TEACHED = 1</li> <li>• ROI_PROCESS_VALUE_TEACHING_FAILED = 2</li> <li>• ROI_PROCESS_VALUE_REFIMAGE_INVALID = 3</li> <li>• ROI_PROCESS_VALUE_NO_VALID_PIXEL = 4</li> <li>• ROI_PROCESS_VALUE_REFIMAGE_NO_VALID_PIXEL = 5</li> <li>• ROI_PROCESS_VALUE_OVERFLOW = 6</li> <li>• ROI_PROCESS_VALUE_UNDERFILL = 7</li> </ul>	uint32
quality	0..1	float32

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

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

## 13.2 Process Interface Command Reference



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

### 13.2.1 t Command (Asynchronous Trigger)

Command	t	
Description	Executes trigger. The result data is send asynchronously	
Type	Action	
Reply	*	Trigger was executed, the device captures an image and evaluates the result.
	!	<ul style="list-style-type: none"> <li>• Device is busy with an evaluation</li> <li>• Device is in an invalid state for this command, e.g. configuration mode</li> <li>• Device is set to a different trigger source</li> <li>• No active application</li> </ul>

### 13.2.2 T? Command (Synchronous Trigger)

Command	T?	
Description	Executes trigger. The result data is send synchronously	
Type	Request	
Reply	Process data within the configured layout	Trigger was executed, the device captures an image, evaluates the result and sends the process data.
	!	<ul style="list-style-type: none"> <li>• Device is busy with an evaluation</li> <li>• Device is in an invalid state for this command, e.g. configuration mode</li> <li>• Device is set to a different trigger source</li> <li>• No active application</li> </ul>
Note	Result data can be sent via EtherNet/IP (data type: SINT) or TCP/IP (data type: ASCII) (→ 9.3).	

## 13.2.3 I? Command

Command	I<image-ID>?	
Description	Request last image taken	
Type	Request	
Reply	<length><image data>	
	!	<ul style="list-style-type: none"> <li>No image available</li> <li>Wrong ID</li> </ul>
	?	<ul style="list-style-type: none"> <li>Invalid command length</li> </ul>
Note	<image-ID> 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"> <li>• Application not available</li> <li>• &lt;application number&gt; contains wrong value</li> <li>• External application switching activated</li> <li>• Device is in an invalid state for this command, e.g. configuration mode</li> </ul>
	?	Invalid command length
Note	<application number> 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"> <li>• Error in configuration</li> <li>• Wrong data length</li> </ul>
	?	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>&lt;vendor&gt;&lt;t&gt;&lt;article number&gt;&lt;t&gt; &lt;name&gt;&lt;t&gt;&lt;location&gt;&lt;t&gt;&lt;descri ption&gt;&lt;t&gt;&lt;ip&gt;  &lt;subnet mask&gt;&lt;t&gt;&lt;gateway&gt;&lt; t&gt;&lt;MAC&gt;&lt;t&gt;&lt;DHCP&gt;&lt;t&gt;&lt;port number&gt;</pre>	
Note	<ul style="list-style-type: none"> <li>• &lt;vendor&gt; IFM ELECTRONIC</li> <li>• &lt;t&gt; Tabulator (0x09)</li> <li>• &lt;article number&gt; e.g. O3D300</li> <li>• &lt;name&gt; UTF8 Unicode string</li> <li>• &lt;location&gt; UTF8 Unicode string</li> <li>• &lt;description&gt; UTF8 Unicode string</li> <li>• &lt;ip&gt; IP address of the device as ASCII character sting e.g. 192.168.0.96</li> <li>• &lt;port number&gt; port number of the XML-RPC</li> <li>• &lt;subnet mask&gt; subnet mask of the device as ASCII e.g. 192.168.0.96</li> <li>• &lt;gateway&gt; gateway of the device as ASCII e.g 192.168.0.96</li> <li>• &lt;MAC&gt; MAC adress of the device as ASCII e.g. AA:AA:AA:AA:AA:AA</li> <li>• &lt;DHCP&gt; ASCII string "0" for off and "1" for on</li> </ul>	

## 13.2.13 H? Command

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

## 13.2.14 o Command

Command	o<IO-ID><IO-state>	
Description	Sets the logic state of a specific ID	
Type	Action	
Reply	*	
	!	Invalid state (e.g. configuration mode)
	?	Invalid command length
Note	<ul style="list-style-type: none"> <li>• &lt;IO-ID&gt; 2 digits for digital output: "01" for IO1 "02" for IO2 "03" for IO3</li> <li>• &lt;IO-state&gt; 1 digit for the state: "0" for logic state low "1" for logic state high</li> </ul>	

## 13.2.15 O? Command

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

### 13.3 Error codes

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

Error code ID	Description
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

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

#### Consuming assembly data layout

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

#### Layout of producing assembly

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

#### Layout of command word

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

#### Command word

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

#### Synchronous / asynchronous message identifier

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

#### Data to send exceeds processing assembly data section size

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

### 13.4.2 Functionality of the Ethernet/IP application

The chapter describes the initialization of assembly buffers.



On initialization all buffers are set to 0.

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

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

Multiple state changes

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

Reset of command bit state by PLC

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

Blocking of asynchronous messages

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

Client disconnect

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

General reply to an implemented command

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

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

Reply to an implemented command - reply contains specific data

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

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

Reply to an implemented command with error in data section

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

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

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

Reply to an implemented command that leads to an error

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

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

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

Reply to a not implemented command

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

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

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

#### Reset of error bit

The error bit will be reset to 0, if

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

#### Functionality of asynchronous message bit

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

#### Bits for asynchronous message identifier

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

The ticket number for asynchronous results is 0.

The ticket number for asynchronous error codes is 1.

#### Message counter

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

#### Get last error

This command is used to reset the error bit.

#### Get connection ID

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

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

#### Get statistics

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

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

All values are 32 bit unsigned integers.

#### 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.

## 13.5 PROFINET IO

### 13.5.1 Data structures for output and input frame

#### Size of output frame

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

#### Size of input frame

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

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

#### Layout of command word

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

#### Command word

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

#### Synchronous / asynchronous identifier

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

### 13.5.2 Functionality of PROFINET IO application

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

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

Number of supported PROFINET connections

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

Initialisation of input and output buffers

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

Command execution triggering

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

Handling of multiple command bits

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

Command execution completion

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

Blocking of asynchronous messages

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

Client disconnect

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

General reply to an implemented command

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

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

Reply to an implemented command - reply contains specific data

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

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

Reply to an implemented command with error in data section

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

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



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

Reply to an implemented command that leads to an error

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

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



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

Reply to a not implemented command

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

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



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

#### Reset of error bit

The error bit will be reset to 0, if

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

#### Queuing of error codes

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

#### Functionality of asynchronous message bit

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

#### Bits for asynchronous message identifier

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

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

#### Message counter

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

#### Get last error

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

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

#### Get connection ID

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

#### Get statistics

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

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

All values are 32 bit unsigned integers.

### 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.