TECHNICAL INFORMATION

safeVisionary2

Safety camera sensor

Data output via UDP

SICK
Sensor Intelligence.

Described product

safeVisionary2

Manufacturer

SICK AG
Erwin-Sick-Str. 1
79183 Waldkirch
Germany

Legal information

Original document

This document is an original document of SICK AG.

Contents

1 About this document

1.1 Purpose of this document

This document describes advanced usage options of the safeVisionary2 safety camera sensor:

- Output of measurement data and other data via the Ethernet interface

1.2 Scope

Product

This document applies to the following products:

- Product code: safeVisionary2

Document identification

Document part number:

- This document: 8027550
- Available language versions of this document: 8027548

You can find the current version of all documents at www.sick.com.

1.3 Target groups

This document is written for system specialists working in the field of hardware and software development intending to integrate the measurement data or other data and functions of the product in their application.

1.4 Symbols and document conventions

The following symbols and conventions are used in this document:

Warnings and other notes

DANGER
Indicates a situation presenting imminent danger, which will lead to death or serious injuries if not prevented.

WARNING
Indicates a situation presenting possible danger, which may lead to death or serious injuries if not prevented.

CAUTION
Indicates a situation presenting possible danger, which may lead to moderate or minor injuries if not prevented.

NOTICE
Indicates a situation presenting possible danger, which may lead to property damage if not prevented.

NOTE
Highlights useful tips and recommendations as well as information for efficient and trouble-free operation.

Instructions to action

► The arrow denotes instructions to action.

1. The sequence of instructions for action is numbered.
2. Follow the order in which the numbered instructions are given.

✓ The check mark denotes the result of an instruction.

2 Safety information

2.1 General safety notes

DANGER
Danger of using data output for safety function

The data output is suitable for general monitoring and control tasks. It does not meet the requirements for use in safety functions.

► If the data output is to be used in safety functions, define and implement additional required measures.

3 Product description

3.1 Structure and function

Structure and function

The safety camera sensor is an electro-sensitive protective device (ESPE) which scans its surroundings three-dimensionally using infrared laser beams.

As soon as an object is situated in the protective field, the camera sensor signals the detection by means of a signal change at the safety output. The machine or its controller must safely analyze the signals (for example using a safety controller or a safety relay) and stop the dangerous state.

The camera sensor operates according to the optical time-of-flight measurement principle. It emits light pulses at regular, very short intervals. If the light strikes an object, it is reflected. The camera sensor receives the reflected light. The camera sensor calculates the distance to the object based on the time interval between the moment of transmission and moment of receipt (Δt).

Field of view

The camera sensor has a field of view with an aperture angle of 68° × 58° (horizontal × vertical).

For protective field evaluation, the field of view of the camera sensor is limited to 68° × 42°. The full 68° × 58° field of view is available for all other safety functions, field types, and for data output.

Detection zones of different sizes are created depending on distance z.

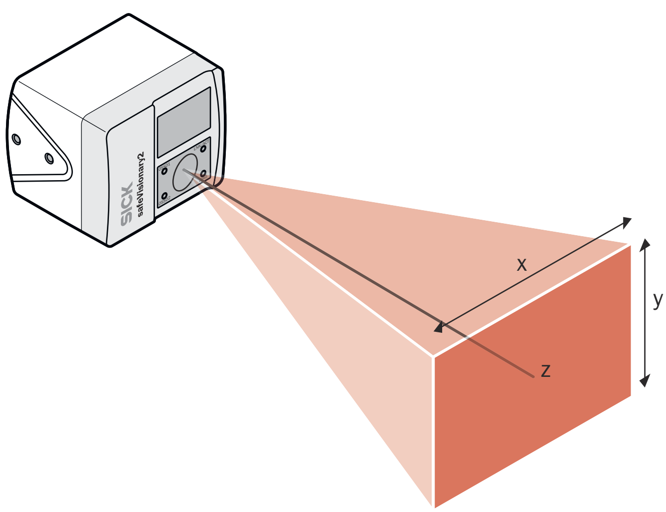

Figure 1: Size of the detection zone

Table 1: Size of the detection zone

Distance z	Field of view 68° × 58°		Limited field of view (protective field) 68° × 42°	
	Size x	Size y	Size x	Size y
0.2 m	0.27 m	0.22 m	0.27 m	0.15 m
0.5 m	0.68 m	0.56 m	0.68 m	0.39 m
1.0 m	1.35 m	1.11 m	1.35 m	0.77 m
1.5 m	2.03 m	1.67 m	2.03 m	1.14 m

Distance z	Field of view 68° × 58°		Limited field of view (protective field) 68° × 42°	
	Size x	Size y	Size x	Size y
2.0 m	2.7 m	2.22 m	2.7 m	1.51 m
4.0 m	5.4 m	4.44 m	5.4 m	3.02 m

Optical axis

The optical axis is perpendicular to the front screen of the camera module (see figure 1, page 7, z-axis).

Resolution

The object resolution indicates the minimum size that an object must be to allow it to be detected safely.

The resolution can be set to various values to suit the intended purpose.

Measurement data

Measurement data is, for example, the distance data for each individual light pulse. The measurement data can be output via the Ethernet interface. In addition to the measurement data, other data can also be output, e.g. on object detections and the device status.

4 Data output

4.1 Overview

DANGER
Danger of using data output for safety function

The data output is suitable for general monitoring and control tasks. It does not meet the requirements for use in safety functions.

► If the data output is to be used in safety functions, define and implement additional required measures.

Data output allows for the output of measurement data and other data via the Ethernet interface. Other network participants, the receivers, can call up and use the data.

You can define which data the device should output in the configuration of the data output. The actually available data depends on the operational status of the device, among other things. Therefore not all configured data is output, rather only the data which is currently available.

4.2 Data output

Overview

You can use Safety Designer to configure the device and the data output. A configuration that is created with the Safety Designer is saved in the device and is also active after restarting the device.

Data output can be used for general monitoring and control tasks. This data is used in particular for providing navigation support for AGVs (automated guided vehicles). This data must not be used for safety-related applications.

Send Mode

- **Deactivated:** No data output
- **Continuously to a target computer (router settings are configured via "Network settings"):** Data output continuously or at every xth measurement via UDP to a defined destination address

Data content selection

Data content
- **Full:** All data is output.
- **Reduced:** The intensity values are left out.
- **ROI only:** Only the ROI data is output.

Data content selection
- **Filtered (pre-processed)**
  The measurement data is pre-filtered. Noise and measurement artifacts are reduced.
- **Raw**
  The unfiltered measurement data is output.

4.3 Contents of the data output

Overview

The structure of the output data begins with a header. The header is followed by the data segments. You can influence the scope of the output data using the settings in the Safety Designer.

The following data segments are available:
- XML description
- Depth map
- Device status
- Regions of Interest (ROIs)
- Local inputs and outputs
- Fields
- Logical inputs and outputs
- IMU (inertial measuring unit)

Further topics

- "Structure of the data output", page 13

4.4 Interpretation of the data transmitted via UDP

This section describes the interpretation of the UDP datagram if continuous data output has been configured. The data receiver is clearly identified through their IPv4 address and the port number.

NOTE
Several devices (or several channels of a device) cannot send your data to the same port of the same target system. If a system should receive data from several devices or channels, then you must use a clear port for each device and each channel.

A telegram (an instance of data output) is too large for a UDP datagram. Each telegram is therefore split up into fragments and sent in several sequential UDP datagrams. The payload of each UDP datagram comprises a maximum of 1,460 bytes.

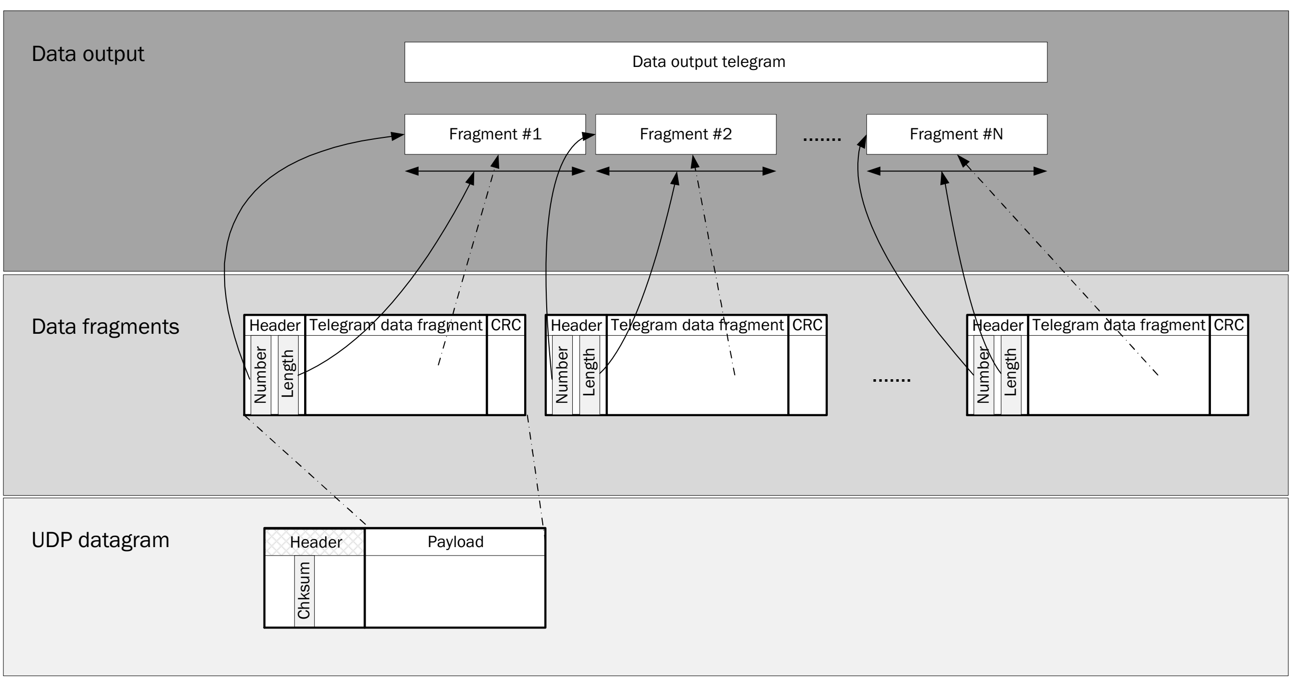

Figure 2: UDP datagram and measurement data

The data integrity of each individual UDP datagram is ensured with the UDP checksum. The UDP neither ensures the arrival of individual datagrams, nor the sequence, nor protection from duplicates. Additional control data is therefore added to each UDP datagram for data output, see table 2, page 11. Based on this data, the receiver can detect duplicates and lost datagrams, restore the sequence, and reassemble the fragmented telegrams. As UDP does not offer the opportunity to re-request lost datagrams, receivers must be able to deal with data loss. If an error is detected in a UDP datagram, that telegram must be discarded.

Table 2: UDP payload

Index	Size	Name	Description
0	16 bit	Telegram number	The numbering starts at 0. The number is incremented by 1 for each new telegram. Using the number, the receiver can recognize which UDP datagrams belong together.
2	16 bit	Fragment number	The numbering starts at 0. The number is incremented by 1 for each new fragment of a telegram. For a new telegram, the numbering restarts at 0.
4	32 bit	Time stamp	Time when the UDP datagram was created. The time is specified in microseconds (µs) from system start.
8	32 bit	Source IP address	IP address of the sensor.
12	16 bit	Source port	UDP source port of the sensor.
14	32 bit	Destination IP address	IP address of the receiver.
18	16 bit	Destination port	UDP destination port of the receiver.
20	16 bit	Protocol version	Version of the UDP telegram protocol, currently 0x0001. The protocol version will, in future, allow changes to the protocol and backward compatibility.
22	16 bit	Length	Length of the "telegram data" field in the UDP datagram.
24	8 bit	Flags	In the last fragment of a telegram, bit 7 has the value 1. In all other fragments, bit 7 has the value 0. All other bits are reserved.
25	8 bit	Package type	Packet type of the UDP datagram: 0x62 = b = data packet.
26 … n	1 … 1,430 bytes	Telegram data	Fragment of the current telegram. The field has a maximum size of 1,430 bytes.
n + 1	32 bit	CRC	CRC checksum with the polynomial 0x1EDC6F41 (CRC-32C) and initialization value 0xFFFFFF. The CRC checksum is calculated over the payload of the UDP datagram (index 0 to index n).

Because of the additional header, the actual fragment of the data output starts at offset 26 of the UDP payload.

5 Technical data

5.1 Data sheet

Features

Table 3: Features

Data output channel	1
ROI scanning range	≤ 4 m
Distance measurement range	≤ 16 m [1)]

1) For values > 9 m, quality losses (e.g. larger statistical measurement errors) must be expected.

Miscellaneous data

Table 4: Miscellaneous data

Field of view	68° × 58°
Image repeatability rate	30 Hz
Data resolution	0.25 mm

Table 5: Measurement uncertainty

	Remission 4% (typical)	**Remission 100% (typical)**		
Scanning range	2 m	2 m	4 m	8 m
Statistical error (pixel) 1 σ	± 12 mm	± 2 mm	± 2 mm	± 5 mm
Systematic error	± 20 mm	± 20 mm	± 20 mm	± 40 mm

Example of statistical error: 68.2% of all measured values at 4% remission are within the limits of +/- 12 mm.

6 Annex

6.1 Structure of the data output

The data segments must be addressed by the offset specified in the header. The data segments must not be addressed via a fixed offset because the size of a data segment and thus the offset can change at runtime.

In future versions of the protocol, additional data may be appended to the header or data segments. Access to the data remains compatible if the structure information from data segment 0 is used as the basis.

Table 6: Data output: Header

Section	Length in bytes	Description
STx	4	0x02020202 Start. 4 STx characters (0x02) mark the start of each telegram. This pattern is not exclusive. After a loss of synchronization, the beginning of a telegram can be found using this pattern and the specified telegram length.
Telegram length	4	Length of the telegram. The number of bytes that follow as the rest of the telegram. The specified number of bytes should be followed by another 4 STx characters that mark the beginning of the next telegram.
Protocol version	2	0x0001 Protocol version: 1.0
Package type	1	0x62 Package type: Data
Telegram ID	2	1 Telegram type: 3D
Number N of data segments	2	Number N of data segments in the telegram
Segment info	N × 8	**8 bytes per data segment** • 4 bytes: Offset of each data segment 0 ... (N – 1), calculated from the telegram ID field. The Telegram ID field has the offset 0. • 4 bytes: Change counter for each data segment. The counter remains the same as long as the data of the data segment remains the same. Each time the data of a data segment changes, the associated counter is incremented by 1.

Table 7: Data segment 0: XML description

Section	Length in bytes	Description
Data		For a description of the telegram content in XML format, see "Structure of the XML file", page 23. At the beginning of the communication, the receiver can recognize from this data how the telegram is structured.

Table 8: Data segment 1: Depth map

Section	Length in bytes	Description
Length	4	**Length of the data segment (in bytes)** Sum of the lengths of these fields: • Data • CRC • Length (This field is considered only once.)

<table>
<tr><th>Section</th><th>Length in bytes</th><th>Description</th></tr>
<tr><td rowspan="8">Data</td><td>8</td><td>Time stamp
Time (UTC) at which the data was collected
Bit 0 … Bit 9: Milliseconds
Bit 10 … Bit 15: Seconds
Bit 16 … Bit 21: Minutes
Bit 22 … Bit 26: Hours
Bit 27 … Bit 37: Time zone (signed, in minutes) (The value is always 0 = UTC.)
Bit 38 … Bit 42: Day
Bit 43 … Bit 46: Month
Bit 47 … Bit 58: Year
Bit 29 … Bit 63: Reserved</td></tr>
<tr><td>2</td><td>Version: 0x0002
If SICK changes the structure of the data segment, then it is given a higher version number. Each data segment is versioned independently of the others. It is recommended to check the version number in the application.</td></tr>
<tr><td>4</td><td>Image acquisition number
This number is incremented by 1 for each captured image.</td></tr>
<tr><td>1</td><td>Device status
0: Configuration
1: Waiting for inputs
2: Application stopped
3: normal operation</td></tr>
<tr><td>2</td><td>Flags
Bit 0: Depth map unfiltered/filtered
Bit 1: Detection data invalid/valid
Bit 2 … Bit 15: Reserved</td></tr>
<tr><td>434176</td><td>Distance [1)]
Depth map, 512 px × 424 px, 16 bit (depth resolution: 1 increment corresponds to 0.25 mm)</td></tr>
<tr><td>434176</td><td>Intensity [1)]
Intensity map, 512 px × 424 px, 16 bit (intensity values: 0 … 20,000. The value 20,000 represents a saturated pixel)</td></tr>
<tr><td>217088</td><td>Pixel status map, 8 bits per pixel [1)]
<table>
<tr><th></th><th colspan="2">Meaning</th></tr>
<tr><th>Bit</th><th>Value = 0</th><th>Value = 1</th></tr>
<tr><td>0</td><td>Pixel valid</td><td>Pixel invalid</td></tr>
<tr><td>1</td><td colspan="2">Reserved</td></tr>
<tr><td>2</td><td colspan="2">Reserved</td></tr>
<tr><td>3</td><td>Distance value unique</td><td>Distance value not unique</td></tr>
<tr><td>4</td><td>Ambient light neither too strong nor too weak</td><td>Ambient light too strong or too weak</td></tr>
<tr><td>5</td><td>No detection or pixel is not in active field</td><td>Pixel is in active field and reports detection</td></tr>
<tr><td>6</td><td colspan="2">Reserved</td></tr>
<tr><td>7</td><td>Distance ≤ 9 m</td><td>Distance > 9 m</td></tr>
</table></td></tr>
</table>

Section	Length in bytes	Description
CRC	4	**Checksum (CRC-32)** • Polynomial: 0x04C11DB7 • Initialization value: 0xFFFFFF • Termination: XOR with 0xFFFFFF • The checksum is calculated exclusively over the Data field.
Length	4	**Length of the data segment (in bytes)** Sum of the lengths of these fields: • Data • CRC • Length (This field is considered only once.)

1) All cards are output line by line.

Table 9: Data segment 2: Device status

Section	Length in bytes	Description
Length	4	**Length of the data segment (in bytes)** Sum of the lengths of these fields: • Data • CRC • Length (This field is considered only once.)
Data	8	**Time stamp** Time (UTC) at which the data was collected • Bit 0 ... Bit 9: Milliseconds • Bit 10 ... Bit 15: Seconds • Bit 16 ... Bit 21: Minutes • Bit 22 ... Bit 26: Hours • Bit 27 ... Bit 37: Time zone (signed, in minutes) (The value is always 0 = UTC.) • Bit 38 ... Bit 42: Day • Bit 43 ... Bit 46: Month • Bit 47 ... Bit 58: Year • Bit 29 ... Bit 63: Reserved
	2	**Version: 0x0001** If SICK changes the structure of the data segment, then it is given a higher version number. Each data segment is versioned independently of the others. It is recommended to check the version number in the application.
	2	**Device status** • Bit 0: RunModeactive (device in normal operation) • Bit 1: Device error • Bit 2: Application error • Bit 3: Sleep mode • Bit 4: Waiting for inputs • Bit 5: Reserved • Bit 6: Contamination warning • Bit 7: Contamination error • Bit 8: Close range warning • Bit 9: Temperature warning Bit 10 ... Bit 15: Reserved

Section	Length in bytes	Description
	4	**Cut-off path (safety-oriented)** The signal is ON if the currently monitored field in the cut-off path is safety-related and free. • Bit 0: Cut-off path 1 for fields • Bit 1: Cut-off path 2 for fields • Bit 2: Cut-off path 3 for fields • Bit 3 ... Bit 9: Reserved • Bit 10: Cut-off path 1 for ROIs • Bit 11: Cut-off path 2 for ROIs • Bit 12: Cut-off path 3 for ROIs • Bit 13: Cut-off path 4 for ROIs • Bit 14: Cut-off path 5 for ROIs • Bit 15 ... Bit 31: Reserved
	4	**Cut-off path (Not safety-related)** The signal is ON if the currently monitored field in the cut-off path is free. • Bit 0: Cut-off path 1 for fields • Bit 1: Cut-off path 2 for fields • Bit 2: Cut-off path 3 for fields • Bit 3 ... Bit 9: Reserved • Bit 10: Cut-off path 1 for ROIs • Bit 11: Cut-off path 2 for ROIs • Bit 12: Cut-off path 3 for ROIs • Bit 13: Cut-off path 4 for ROIs • Bit 14: Cut-off path 5 for ROIs • Bit 15 ... Bit 31: Reserved
	4	Reserved
	4	**Active monitoring case** Monitoring case number (byte array) • Byte 0: Current monitoring case (monitoring case table 1) • Byte 1: Reserved • Byte 2: Reserved • Byte 3: Reserved
	1	Degree of contamination in % (100 = front screen clear, no contamination; 0 = front screen impermeable)
CRC	4	**Checksum (CRC-32)** • Polynomial: 0x04C11DB7 • Initialization value: 0xFFFFFF • Termination: XOR with 0xFFFFFF • The checksum is calculated exclusively over the Data field.
Length	4	**Length of the data segment (in bytes)** Sum of the lengths of these fields: • Data • CRC • Length (This field is considered only once.)

Table 10: Data segment 3: Regions of Interest (ROIs)

Section	Length in bytes	Description
Length	4	**Length of the data segment (in bytes)** Sum of the lengths of these fields: • Data • CRC • Length (This field is considered only once.)

<table>
<tr><th>Section</th><th>Length in bytes</th><th>Description</th></tr>
<tr><td rowspan="3">Data</td><td>8</td><td>Time stamp
Time (UTC) at which the data was collected
Bit 0 ... Bit 9: Milliseconds
Bit 10 ... Bit 15: Seconds
Bit 16 ... Bit 21: Minutes
Bit 22 ... Bit 26: Hours
Bit 27 ... Bit 37: Time zone (signed, in minutes) (The value is always 0 = UTC.)
Bit 38 ... Bit 42: Day
Bit 43 ... Bit 46: Month
Bit 47 ... Bit 58: Year
Bit 29 ... Bit 63: Reserved</td></tr>
<tr><td>2</td><td>Version: 0x0001
If SICK changes the structure of the data segment, then it is given a higher version number. Each data segment is versioned independently of the others. It is recommended to check the version number in the application.</td></tr>
<tr><td>5 × 6</td><td>ROI data
1 byte: ID of the ROI
1 byte: Result
Bit 0: Result of the ROI evaluation
Bit 1: Result is safe
Bit 2: Result is valid
Bit 3: Distance is valid
Bit 4: Distance is safe
2 bytes: Information on the safety function
Bit 0: Invalid due to invalid pixel
Bit 1: Invalid due to deviating values
Bit 2: Invalid due to overexposure
Bit 3: Invalid due to underexposure
Bit 4: Invalid due to temporarily deviating values
Bit 5: Invalid due to measurement range exceeded
Bit 6: Invalid due to interference from a retroreflector
Bit 7: Contamination error
Bit 8 ... Bit 9: Quality
0: Invalid
1: High
2: Medium
3: Low
Bit 10: ROI configured
Bit 11 ... Bit 15: Reserved
2 bytes: Distance (1 increment corresponds to 0.25 mm)</td></tr>
<tr><td>CRC</td><td>4</td><td>Checksum (CRC-32)
Polynomial: 0x04C11DB7
Initialization value: 0xFFFFFF
Termination: XOR with 0xFFFFFF
The checksum is calculated exclusively over the Data field.</td></tr>
<tr><td>Length</td><td>4</td><td>Length of the data segment (in bytes)
Sum of the lengths of these fields:
Data
CRC
Length (This field is considered only once.)</td></tr>
</table>

Table 11: Data segment 4: Local inputs and outputs

Section	Length in bytes	Description
Length	4	**Length of the data segment (in bytes)** Sum of the lengths of these fields: • Data • CRC • Length (This field is considered only once.)
Data	8	**Time stamp** Time (UTC) at which the data was collected • Bit 0 ... Bit 9: Milliseconds • Bit 10 ... Bit 15: Seconds • Bit 16 ... Bit 21: Minutes • Bit 22 ... Bit 26: Hours • Bit 27 ... Bit 37: Time zone (signed, in minutes) (The value is always 0 = UTC.) • Bit 38 ... Bit 42: Day • Bit 43 ... Bit 46: Month • Bit 47 ... Bit 58: Year • Bit 29 ... Bit 63: Reserved
	2	**Version: 0x0001** If SICK changes the structure of the data segment, then it is given a higher version number. Each data segment is versioned independently of the others. It is recommended to check the version number in the application.
	2	**Configured universal I/Os** • 0: Not configured or OSSD • 1: Configured, but not as OSSD • Bit 0: Pin 5 (universal I/O 1) • Bit 1: Pin 6 (Universal I/O 2) • Bit 2: Pin 7 (Universal I/O 3) • Bit 3: Pin 8 (Universal I/O 4) • Bit 4 ... Bit 15: Reserved
	2	**Universal I/O used as input or output** • 0: Input • 1: Output • Bit 0: Pin 5 (universal I/O 1) • Bit 1: Pin 6 (Universal I/O 2) • Bit 2: Pin 7 (Universal I/O 3) • Bit 3: Pin 8 (Universal I/O 4) • Bit 4 ... Bit 15: Reserved
	2	**Logical state of the universal I/Os (inputs)** • Bit 0: Pin 5 (universal I/O 1) • Bit 1: Pin 6 (Universal I/O 2) • Bit 2: Pin 7 (Universal I/O 3) • Bit 3: Pin 8 (Universal I/O 4) • Bit 4 ... Bit 15: Reserved
	16	**1 byte per output: State of the universal I/Os (outputs)** • 0: OFF (0 V) • 1: Flashing, 1 Hz • 2: Flashing, 4 Hz • 3: ON (24 V) • 255: Output not used **Byte 4 ... Byte 15: Reserved**

Section	Length in bytes	Description
	1	**Logical state of the OSSDs** • Bit 0: OSSD 1.A • Bit 1: OSSD 1.B • Bit 2: OSSD 2.A • Bit 3: OSSD 2.B • Bit 4 … Bit 7: Reserved
	11	Reserved
CRC	4	**Checksum (CRC-32)** • Polynomial: 0x04C11DB7 • Initialization value: 0xFFFFFF • Termination: XOR with 0xFFFFFF • The checksum is calculated exclusively over the Data field.
Length	4	**Length of the data segment (in bytes)** Sum of the lengths of these fields: • Data • CRC • Length (This field is considered only once.)

Table 12: Data segment 5: Fields

Section	Length in bytes	Description
Length	4	**Length of the data segment (in bytes)** Sum of the lengths of these fields: • Data • CRC • Length (This field is considered only once.)
Data	8	**Time stamp** Time (UTC) at which the data was collected • Bit 0 … Bit 9: Milliseconds • Bit 10 … Bit 15: Seconds • Bit 16 … Bit 21: Minutes • Bit 22 … Bit 26: Hours • Bit 27 … Bit 37: Time zone (signed, in minutes) (The value is always 0 = UTC.) • Bit 38 … Bit 42: Day • Bit 43 … Bit 46: Month • Bit 47 … Bit 58: Year • Bit 29 … Bit 63: Reserved
	2	**Version: 0x0001** If SICK changes the structure of the data segment, then it is given a higher version number. Each data segment is versioned independently of the others. It is recommended to check the version number in the application.

Section	Length in bytes	Description

	16 × 5	**Data of the active fields**
		Data is output for the fields that are used in the active monitoring case. The data of the inactive fields are always 0.
		• 1 byte: ID of the field
		  ◦ 0 … 255
		• 1 byte: ID of the field set
		  ◦ 0 … 255
		• 1 byte: Evaluation result of the field
		  ◦ 0: Detection in field
		  ◦ 1: Field free
		  ◦ 2 … 255: Reserved
		• 1 byte: Evaluation method of the field
		  ◦ 0: Always OFF
		  ◦ 1: Always ON (safe)
		  ◦ 2: Always ON (non safe)
		  ◦ 3: Sleep mode
		  ◦ 4: Protective field
		  ◦ 5: Warning field
		  ◦ 6: Contour detection field (object resolutions: Hand, Arm, Leg, Body)
		  ◦ 7: Contour detection field (fall edge detection, object resolutions: 40x40 cm, 100x100 cm)
		  ◦ 8: ROI
		  ◦ 9 … 255: Reserved
		• 1 byte: Field in active monitoring case
		  ◦ 0: Field not active
		  ◦ 1: Field active
		  ◦ 2 … 255: Reserved

CRC	4	**Checksum (CRC-32)** • Polynomial: 0x04C11DB7 • Initialization value: 0xFFFFFF • Termination: XOR with 0xFFFFFF • The checksum is calculated exclusively over the Data field.
Length	4	**Length of the data segment (in bytes)** Sum of the lengths of these fields: • Data • CRC • Length (This field is considered only once.)

Table 13: Data segment 6: Logical inputs and outputs

Section	Length in bytes	Description
Length	4	**Length of the data segment (in bytes)** Sum of the lengths of these fields: • Data • CRC • Length (This field is considered only once.)

Section	Length in bytes	Description
Data	8	**Time stamp** Time (UTC) at which the data was collected • Bit 0 ... Bit 9: Milliseconds • Bit 10 ... Bit 15: Seconds • Bit 16 ... Bit 21: Minutes • Bit 22 ... Bit 26: Hours • Bit 27 ... Bit 37: Time zone (signed, in minutes) (The value is always 0 = UTC.) • Bit 38 ... Bit 42: Day • Bit 43 ... Bit 46: Month • Bit 47 ... Bit 58: Year • Bit 29 ... Bit 63: Reserved
	2	**Version: 0x0001** If SICK changes the structure of the data segment, then it is given a higher version number. Each data segment is versioned independently of the others. It is recommended to check the version number in the application.

	20 × 6	**1 byte: Signal type**
		• 0: OSSD
		• 1: Static control input
		• 2: Reserved
		• 3: External device monitoring (EDM)
		• 4: Reset
		• 5: Sleep mode
		• 6: Restart device completely
		• 7: Restart security function and connections
		• 8: Restart safety function
		• 9: Contamination warning
		• 10: Contamination error
		• 11: Device error
		• 12: Application error
		• 13: Reset required
		• 14: Monitoring result (non safe, 0: detection, 1 = field free)
		• 15: Pause event recording
		• 16: Parity input
		• 17 ... 254: Reserved
		• 255: Input/output not used
		
		1 byte: Instance
		For some signal types there are multiple instances. For most signal types there is only one instance; the value of this byte is then 0. Other values are only possible for the following signal types.
		• OSSD
		  ◦ 0: OSSD 1
		  ◦ 1: OSSD 2
		• Static input
		  ◦ 0: A1
		  ◦ 1: A2
		  ◦ 2: B1
		  ◦ 3: B2
		• External device monitoring (EDM)
		  ◦ 0: External device monitoring (EDM) for OSSD 1
		  ◦ 1: External device monitoring (EDM) for OSSD 2
		• Reset
		  ◦ 0: Reset for OSSD 1
		  ◦ 1: Reset for OSSD 2

Section	Length in bytes	Description

		• Reset required
		  ○ 0: Reset required for OSSD 1
		  ○ 1: Reset required for OSSD 2
		• Evaluation result
		  ○ 0: Cut-off path 1
		  ○ 1: Cut-off path 2
		  ○ 2: Cut-off path 3
		  ○ 3: Cut-off path 4
		  ○ 4: Cut-off path 5
		  ○ 5: Cut-off path 6
		  ○ 6: Cut-off path 7
		  ○ 7: Cut-off path 8
		• Parity input
		  ○ 0: P4.1
		  ○ 1: P4.2 / P3.1
		  ○ 2: P4.3 / P3.2
		  ○ 3: P4.P / P3.P
		1 bit: Input/output configured
		• 0: Not configured
		• 1: Configured
		1 bit: Input or output
		• 0: Input
		• 1: Output
		14 bit: Reserved
		2 bytes: Logical state

CRC	4	**Checksum (CRC-32)** • Polynomial: 0x04C11DB7 • Initialization value: 0xFFFFFF • Termination: XOR with 0xFFFFFF • The checksum is calculated exclusively over the Data field.
Length	4	**Length of the data segment (in bytes)** Sum of the lengths of these fields: • Data • CRC • Length (This field is considered only once.)

Table 14: Data segment 7: IMU (inertial measuring unit)

Section	Length in bytes	Description
Length	4	**Length of the data segment (in bytes)** Sum of the lengths of these fields: • Data • CRC • Length (This field is considered only once.)
Data	8	**Time stamp** The time stamp of the inertial measuring unit does not depend on the other time stamps.
	2	**Version: 0x0001** If SICK changes the structure of the data segment, then it is given a higher version number. Each data segment is versioned independently of the others. It is recommended to check the version number in the application.

<table>
<tr><th>Section</th><th>Length in bytes</th><th>Description</th></tr>
<tr><td rowspan="3"></td><td>13</td><td>Acceleration in m/s², vector
<table>
<tr><th>Byte</th><th>Contents</th><th>Data type</th></tr>
<tr><td>0 … 3</td><td>X</td><td>FP32</td></tr>
<tr><td>4 … 7</td><td>Y</td><td>FP32</td></tr>
<tr><td>8 … 11</td><td>Z</td><td>FP32</td></tr>
<tr><td>12</td><td>Accuracy</td><td>• 0 = unreliable
• 1: not very accurate
• 2: moderately accurate
• 3: very accurate</td></tr>
</table></td></tr>
<tr><td>13</td><td>Angle speed in rad/s, vector
<table>
<tr><th>Byte</th><th>Contents</th><th>Data type</th></tr>
<tr><td>0 … 3</td><td>X</td><td>FP32</td></tr>
<tr><td>4 … 7</td><td>Y</td><td>FP32</td></tr>
<tr><td>8 … 11</td><td>Z</td><td>FP32</td></tr>
<tr><td>12</td><td>Accuracy</td><td>• 0 = unreliable
• 1: not very accurate
• 2: moderately accurate
• 3: very accurate</td></tr>
</table></td></tr>
<tr><td>20</td><td>Orientation, unit quaternion
<table>
<tr><th>Byte</th><th>Contents</th><th>Data type</th></tr>
<tr><td>0 … 3</td><td>X</td><td>FP32</td></tr>
<tr><td>4 … 7</td><td>Y</td><td>FP32</td></tr>
<tr><td>8 … 11</td><td>Z</td><td>FP32</td></tr>
<tr><td>12 … 15</td><td>W</td><td>FP32</td></tr>
<tr><td>16 … 19</td><td>Accuracy in rad</td><td>FP32</td></tr>
</table></td></tr>
<tr><td>CRC</td><td>4</td><td>Checksum (CRC-32)
• Polynomial: 0x04C11DB7
• Initialization value: 0xFFFFFF
• Termination: XOR with 0xFFFFFF
• The checksum is calculated exclusively over the Data field.</td></tr>
<tr><td>Length</td><td>4</td><td>Length of the data segment (in bytes)
Sum of the lengths of these fields:
• Data
• CRC
• Length (This field is considered only once.)</td></tr>
</table>

6.2 Structure of the XML file

Overview

The XML file from data segment 0 contains a description of the telegram content and some additional information.

Overview of the elements

- `Revision`: Version of the XML schema
- `SchemaChecksum`: The element and its contents are output for compatibility reasons only.
- `ChecksumFile`: The element and its contents are output for compatibility reasons only.

- `RecordDescription`: Information about the present XML file
- `DataSets`: Contains one entry for each data segment.
  - `DataSet…`: Each data segment is described in a separate element.
    - `FormatDescription…`: For each data segment, describes the data format of the individual fields.

DataSetDepthMap

The `DataSetDepthMap` element contains a description of the data segment 1 (depth map) as well as additional information.

- `DataSetDepthMap`
  - `DeviceDescription`: Information about the device sending the data.
    - `Family`: Type code of the device
    - `Ident`: Device name as assigned in Safety Designer
    - `Version`: Version of the device
    - `SerialNumber`: Serial number of the device
    - `LocationName`: Application name, as assigned in Safety Designer
    - `IPAddress`: IP address of the device
  - `FormatDescriptionDepthMap`: Description of the individual fields in the data segment
    - `TimestampUTC`: Data format of the Time stamp field
    - `Version`: Data format of the Version field
    - `DataStream`: Information about the format of the data fields and data transformation
      - `Interleaved`: The element and its contents are output for compatibility reasons only.
      - `Width`: Width of the maps: 512 pixels
      - `Height`: height of the maps: 424 pixels
      - `CameraToWorldTransform`: Continuous float64 floating point array with 128 bytes (4 × 4 = 16 elements). Serves as a transformation matrix and transforms an uniform 3D point from (relative to the front screen) camera coordinates to world coordinates (right-handed system).
      - `CameraMatrix`: Intrinsic camera parameters
        - `FX`: Focal length in X dimension
        - `FY`: Focal length in Y dimension
        - `CX`: X pixel position of the optical center
        - `CY`: Y pixel position of the optical center
      - `CameraDistortionParams`: Intrinsic camera parameters
        - `K1`: Radial correction factor 1
        - `K2`: Radial correction factor 2
        - `P1`: Tangential correction factor 1
        - `P2`: Tangential correction factor 2
        - `K3`: Radial correction factor 3
      - `FocalToRayCross`: Optical offset between the front screen (reference point) and beam intersection point in the optical system to which the raw distance values refer.
      - `FrameNumber`: Data format of the image acquisition number field
      - `Status`: Data format of the Device status field
      - `Flags`: Data format of the Flags field
      - `Distance`: Data format of the Distance field
      - `Intensity`: Data format of the Intensity field
      - `Confidence`: Data format of the Pixel status map field
    - `DeviceInfo`: Device status: `OK` or `ERROR`

- `DataLink`: The element and its contents are output for compatibility reasons only.

6.3 Converting the depth map

Overview

The depth map of the data output contains radial distance data in units of 0.25 mm. The distance refers to an inaccessible point in the device and depends on the calibration of the specific device. The first conversion converts this depth map into 3D points (X,Y,Z) based on a right-handed coordinate system and the reference point on the front screen. This requires the elements `CameraMatrix`, `CameraDistortionParams` and `FocalToRayCross` from the XML file.

The XML file contains the transformation matrix `CameraToWorldTransform`. Using this transformation matrix, the second conversion transfers the 3D points (X,Y,Z) into the reference system specified in the mounting settings of the camera. This reference system will be referred to in the following as “world”.

Important information

NOTE
Calibration data and measurement data are device-specific. They cannot therefore be combined arbitrarily.

If the radial distance of a pixel is 0, then the pixel is invalid and the calculation for that pixel should be skipped.

The second conversion has an effect only if mounting settings are available.

Prerequisites

- (x,y) are the sensor coordinates of a pixel. $0 \le x <$ `Width`, $0 \le y <$ `Height`.
  Example: (0,0) is the upper left pixel of the two-dimensional image sensor.
- For the second conversion, a right-handed coordinate system is assumed.

Procedure

Converting the depth map to device coordinates

1. x' = (x – `CX`) / `FX`
2. y' = (y – `CY`) / `FY`
   This step corresponds to the inverse K matrix.
3. $r = \sqrt{(x'^2 + y'^2)}$
4. x'' = x' × (1 + `K1` × r^2 + `K2` × r^4 + `K3` × r^6)
5. y'' = y' × (1 + `K1` × r^2 + `K2` × r^4 + `K3` × r^6)
   This step corresponds to rectification. The model corresponds to the radial-tangential model used for example in OpenCV. The tangential distortion is always 0 for this device and is not used.
6. $div = \sqrt{(1 + x''^2 + y''^2)}$
7. X = –1 × (Distance(x,y) / 4) × (x'' / div)
8. Y = –1 × (Distance(x,y) / 4) × (y'' / div)
9. Zi = (Distance(x,y) / 4) / div – `FocalToRayCross`
   The negation in X and Y rotates the axes of the device-internal system into the target system. Division by 4 converts the unit of the data from “0.25 mm” to mm.

✓ (X,Y,Z) is the 3D point in device coordinates.

Converting device coordinates into world coordinates

1. Pd = (X,Y,Z,1)

The transformation matrix `CameraToWorldTransform` is a 4x4 matrix operating on uniform coordinates. The 3D point (X,Y,Z) is converted into an uniform 3D point (X,Y,Z,1) by appending a 1-element.

2. Pw = `CameraToWorldTransform` × Pd
   The uniform 3D point in device coordinates Pd is transformed into an uniform 3D point Pw in the world system.

✓ Pw is an uniform 3D point (X',Y',Z',W'). For this device, W' = W = 1 and the conventional 3D point (X',Y',Z') can therefore be read directly from it. (X',Y',Z') is the 3D point in world coordinates.

7 List of figures

8 List of tables

8027550/1KW2/2023-08-14/en

Australia
Phone +61 (3) 9457 0600
1800 33 48 02 – tollfree
E-Mail sales@sick.com.au

Austria
Phone +43 (0) 2236 62288-0
E-Mail office@sick.at

Belgium/Luxembourg
Phone +32 (0) 2 466 55 66
E-Mail info@sick.be

Brazil
Phone +55 11 3215-4900
E-Mail comercial@sick.com.br

Canada
Phone +1 905.771.1444
E-Mail cs.canada@sick.com

Czech Republic
Phone +420 234 719 500
E-Mail sick@sick.cz

Chile
Phone +56 (2) 2274 7430
E-Mail chile@sick.com

China
Phone +86 20 2882 3600
E-Mail info.china@sick.net.cn

Denmark
Phone +45 45 82 64 00
E-Mail sick@sick.dk

Finland
Phone +358-9-25 15 800
E-Mail sick@sick.fi

France
Phone +33 1 64 62 35 00
E-Mail info@sick.fr

Germany
Phone +49 (0) 2 11 53 010
E-Mail info@sick.de

Greece
Phone +30 210 6825100
E-Mail office@sick.com.gr

Hong Kong
Phone +852 2153 6300
E-Mail ghk@sick.com.hk

Hungary
Phone +36 1 371 2680
E-Mail ertekesites@sick.hu

India
Phone +91-22-6119 8900
E-Mail info@sick-india.com

Israel
Phone +972 97110 11
E-Mail info@sick-sensors.com

Italy
Phone +39 02 27 43 41
E-Mail info@sick.it

Japan
Phone +81 3 5309 2112
E-Mail support@sick.jp

Malaysia
Phone +603-8080 7425
E-Mail enquiry.my@sick.com

Mexico
Phone +52 (472) 748 9451
E-Mail mexico@sick.com

Netherlands
Phone +31 (0) 30 204 40 00
E-Mail info@sick.nl

New Zealand
Phone +64 9 415 0459
0800 222 278 – tollfree
E-Mail sales@sick.co.nz

Norway
Phone +47 67 81 50 00
E-Mail sick@sick.no

Poland
Phone +48 22 539 41 00
E-Mail info@sick.pl

Romania
Phone +40 356-17 11 20
E-Mail office@sick.ro

Singapore
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Slovakia
Phone +421 482 901 201
E-Mail mail@sick-sk.sk

Slovenia
Phone +386 591 78849
E-Mail office@sick.si

South Africa
Phone +27 10 060 0550
E-Mail info@sickautomation.co.za

South Korea
Phone +82 2 786 6321/4
E-Mail infokorea@sick.com

Spain
Phone +34 93 480 31 00
E-Mail info@sick.es

Sweden
Phone +46 10 110 10 00
E-Mail info@sick.se

Switzerland
Phone +41 41 619 29 39
E-Mail contact@sick.ch

Taiwan
Phone +886-2-2375-6288
E-Mail sales@sick.com.tw

Thailand
Phone +66 2 645 0009
E-Mail marcom.th@sick.com

Turkey
Phone +90 (216) 528 50 00
E-Mail info@sick.com.tr

United Arab Emirates
Phone +971 (0) 4 88 65 878
E-Mail contact@sick.ae

United Kingdom
Phone +44 (0)17278 31121
E-Mail info@sick.co.uk

USA
Phone +1 800.325.7425
E-Mail info@sick.com

Vietnam
Phone +65 6744 3732
E-Mail sales.gsg@sick.com

Detailed addresses and further locations at **www.sick.com**