en

OPERATING INSTRUCTION

TriSpectorP1000

3D Vision

SICK
Sensor Intelligence.

Product information

The TriSpectorP1000 is a programmable 3D camera which uses laser triangulation on objects to produce 3D images. The TriSpectorP1000 acquires multiple height profiles to build a 3D image of an object.

The SICK AppStudio development environment is used for developing customer-specific applications for the TriSpectorP1000. An extensive range of tools, tutorials, and sample apps are available as an aid during the development.

For further information, see *www.sick.com/TriSpectorP1000*.

About this document

This document contains instructions and descriptions that support the setup and operation of the TriSpectorP1000.

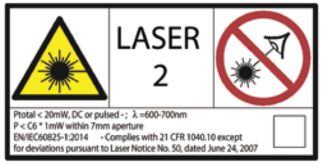
The document is available online in English and in German. Type the part number of the document in the search field on *www.sick.com*. Operating Instruction number English/German: 8022395/8022394.

TriSpectorP1000 variants

The TriSpectorP1000 is available with three different field of view (FoV) sizes and two different window materials: glass and PMMA plastics. The part numbers are shown in the table below.

Variant	Glass	PMMA
TriSpectorP1008 (Small FoV)	1091322	1091318
TriSpectorP1030 (Medium FoV)	1091321	1091319
TriSpectorP1060 (Large FoV)	1091323	1091320

Laser safety



The TriSpectorP1000 is a laser product complying to the standards EN/IEC 60825-1:2014 and EN/IEC 60825-1:2007. It also complies with 21 CFR 1040.10 except for deviations pursuant to Laser Notice No.50, dated June 24, 2007. The legal regulations on laser safety for the laser class of TriSpectorP1000 must be adhered to.

⚠ CAUTION

Optical radiation: Laser class 2

The human eye is not at risk when briefly exposed to the radiation for up to 0.25 seconds. Exposure to the laser beam for longer periods of time may cause damage to the retina. The laser radiation is harmless to human skin.

Do not look into the laser beam intentionally.

Never point the laser beam at people's eyes.

If it is not possible to avoid looking directly into the laser beam, e.g., during commissioning and maintenance work, suitable eye protection must be worn.

Avoid laser beam reflections caused by reflective surfaces. Be particularly careful during mounting and alignment work.

Do not open the housing. Opening the housing will not switch off the laser. Opening the housing may increase the level of risk.

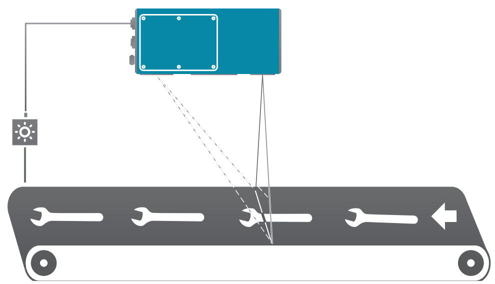
Current national regulations regarding laser protection must be observed.

The laser warning label is located on the black side panel on the opposite side of the connectors. If the TriSpectorP1000 is mounted in a system or a casing, so that the laser safety notice sign is hidden, an additional sign must be placed next to the exit aperture angle of the laser beam on the system casing. Additional signs are not included in the delivery.

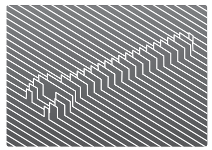
Operating principle

The TriSpectorP1000 creates 3D images of objects by laser triangulation. This means that the object is illuminated with a laser from one direction, and the camera acquires an image of the laser line from another direction.

Each acquired image contains a height profile, which corresponds to a cross-section of the object. By making a scan, which means collecting height profiles across the object while it moves, a complete 3D image can be acquired.



Height profile



3D image (heightmap)

Definitions

The **3D image**, also called heightmap or range data, contains all the acquired height profiles.

The **reflectance** is the reflected intensity along the laser line. The reflectance and the heightmap are acquired simultaneously.

Mounting the TriSpectorP1000

Mount the TriSpectorP1000 in a position above the surface to be scanned. See section E for field of view diagrams and mounting distances.

The default scan direction is shown in the figure above. If a scan is performed in the opposite direction, the acquired image will be mirrored.

Mounting requirements

For optimal performance:

- Observe the ambient conditions for the operation of the TriSpectorP1000 (e.g. ambient temperature, ground potential). ➜ *See „F. Technical data“ section on Page 8.*
- Ensure adequate heat transfer from the device, e.g. via the mounting bracket to the mounting base or by means of convection.
- Use a stable bracket with sufficient load bearing capacity and suitable dimensions for the TriSpectorP1000.
- Minimize shock and vibration.
- Ensure a clear view of the objects to be detected.

Electrical installation

⚠ WARNING

Risk of injury and damage due to electrical current!

As a result of equipotential bonding currents between the device and other grounded devices in the system, faulty grounding of the device may lead to the following dangers and faults:

- Hazardous voltages are applied to the metal housings.
- Devices will malfunction or sustain irreparable damage.
- Cable shieldings are damaged by overheating and cause cable fires.

Remedial measures:

- Only skilled electricians should be permitted to carry out work on the electrical system.
- Ensure that the ground potential is the same at all grounding points.
- If the cable insulation is damaged, disconnect the voltage supply immediately and have the damage repaired.

For a connection diagram for the TriSpectorP1000 ➜ *See „C. Connection diagram“ section on Page 7.*

Observe the following safety measures:

- Standard safety requirements must be met when working in electrical systems.
- Electrical connections between the TriSpectorP1000 and other devices may only be made or disconnected when there is no power to the system. Otherwise, the devices may be damaged.
- When cables with one end open are used, make sure that bare wire ends do not touch each other (risk of short circuit when the supply voltage is switched on). Wires must be appropriately insulated from each other.
- Wire cross sections of the supply cable from the customer's power system should be designed and protected in accordance with the applicable standards. If the supply voltage for the TriSpectorP1000 is not supplied via the CDB650-204 connection module, the TriSpectorP1000 must be protected by a separate slow-blow fuse with a maximum rating of 2.0 A. This fuse must be located at the start of the supply circuit.
- All circuits connected to the TriSpectorP1000 must be designed as SELV circuits (SELV = Safety Extra Low Voltage).

Encoder

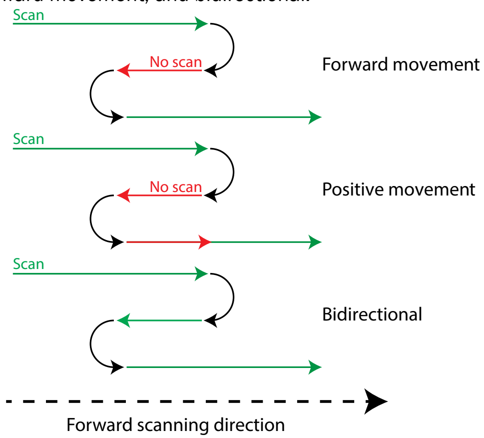
NOTICE

It is strongly recommended to use an encoder for measuring applications, e.g. shape measuring and volume measuring. If no encoder is used, analysis results may be inaccurate due to object traversing speed variations.

The encoder must fulfill the following requirements:

- The encoder must be an incremental encoder, for example SICK DFS60.
- The encoder must have a TTL RS422 interface.
- If there are strong magnetic fields in proximity to the TriSpectorP1000, it is recommended to use a SICK DBS36 Core encoder.
- The connection requires two encoder channels (A/A¯ and B/B¯) to keep track of movement and direction.

The forward scanning direction is defined as clockwise encoder shaft movement, as seen from the tip of the shaft. There are five encoder pulse counter-modes: positive/negative movement, forward/backward movement, and bidirectional.



PC system requirements

Operating systems: Windows 7 or Windows 10 (64 bit is recommended).

Hard drive: Minimum 550 MB.

Working memory: 2 GB recommended.

Processing: 1 GHz CPU recommended. Additional GPU hardware may be required depending on visualization needs of the generated data.

Browser: Google Chrome (www.google.com/chrome) is recommended due to advanced support of WebGL and WebSockets.

Installing SICK AppStudio

The SICK AppStudio development environment is used to program the TriSpectorP1000. Administrator rights are required on the PC to install the software.

License

The download and use of SICK AppStudio requires a valid license. A one-year-license can be purchased on *www.sick.com* (part number 1610199).

Installation

1. Log in to the SICK Support Portal, *supportportal.sick.com*. Registration of a user account is required before logging in for the first time.
2. Download the latest version of SICK AppStudio.
3. Run the installer and follow the instructions on the screen.

At the first use of SICK AppStudio, load or enter your personal license key to complete the installation. If the license dialog does not open automatically, open it by selecting License from the Help menu.

Starting SICK AppStudio

1. Connect the TriSpectorP1000 to the PC via Ethernet.
2. Set the PC to be on the same subnet as the TriSpectorP1000, but with a different IP address. The default IP address for the TriSpectorP1000 is 192.168.0.1.
3. Start SICK AppStudio. By default, the Connection wizard starts and scans for available devices.
4. Select the TriSpectorP1000 in the Device list.
5. Click Connect.

Getting started with the device

The TriSpectorP1000 has a pre-installed app called ImageAcquisition, which helps the user configure the camera's image acquisition settings. By default, the ImageAcquisition app runs as soon as the camera is powered on. The app can be used either with or without starting SICK AppStudio.

To open ImageAcquisition:

1. Open a web browser window.
2. Type the IP address of the TriSpectorP1000 in the address field. The default IP address is 192.168.0.1.

Views

The ImageAcquisition app has two views:

- ImageSetup, which contains a 3D image viewer and a configuration section.
- SensorView, which displays the sensor image.

Switch between the views by clicking the corresponding buttons in the left part of the GUI.

Field of view

Two regions are displayed in the image viewer in the ImageSetup view:

- The guaranteed field of view (yellow) is the region where 3D image acquisition is possible for all TriSpectorP1000 cameras with the same field of view size.
- The selected field of view (blue) is the region in which the camera acquires 3D images with the current configuration. Use the Select tool or the value boxes in the Field of view section to change the position and dimensions of the selected field of view.

Configuration

Configure the camera by adjusting the parameters in the ImageSetup view. By default, images are acquired continuously and the selected field of view is set to its maximum.

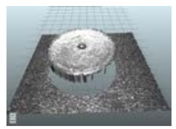
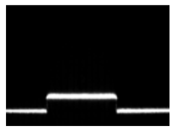
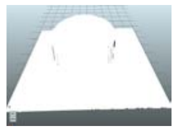
NOTICE

Applying changed parameters

When a parameter or setting has been changed, click the Set Configuration button to apply the change.

Configuring the Acquisition settings

1. Go to the SensorView and check the laser line. The optimal exposure time yields a gray line. A slightly brighter line is preferable to a slightly darker line. See the figures below for examples of sensor images and corresponding 3D images, including reflectance overlay.

	Sensor view	3D image
Short exposure time		
Normal exposure time		
Long exposure time		

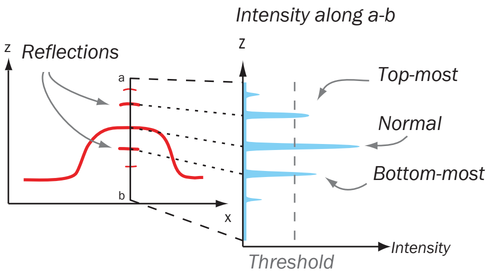
2. To change the exposure time, go to the ImageSetup view and adjust the Exposure parameter.
3. Repeat steps 1-2 until the laser line is gray and the heightmap looks good.
4. Adjust the Gain parameter if needed. A low gain value keeps the noise level down. A high gain value increases the image brightness, but also

increases the noise level.

5. Adjust the LaserThreshold parameter to determine which sensor intensity values to include when locating laser peaks. A high threshold value gives less noise in the heightmap, while a low threshold value makes it possible to detect weak laser peaks.

6. Select laser line acquisition criteria with the PeakSelection parameter:

   - Strongest locates the point with the highest intensity in each column.
   - Top Most and Bottom Most locate the highest or lowest point with an intensity higher than the value set by the LaserThreshold parameter.



Configuring the Motion settings

- Set the ProfileTrigger parameter to Encoder to acquire profiles based on encoder input. A profile is triggered each time the sensor has received a specified number of ticks from the encoder.
- Set the ProfileTrigger parameter to Freerunning to acquire profiles continuously. In free running mode, the time between two profiles is determined by the Speed and ProfileDistance parameters. The object traversing speed must be constant to get heightmaps with correct proportions.
- Set the image resolution by adjusting the ProfileDistance and X-Resolution parameters. Keep the pixels as square as possible to ensure that all image processing algorithms work smoothly.

Configuring the Field of view settings

Adjust the Width, Height, and Length parameters of the field of view to fit the object. The number of profiles in the acquired heightmap is determined by the Length parameter divided by the ProfileDistance parameter. The maximum number of height profiles is 2500.

Configuring the Image Trigger settings

- Select None (continuous) to acquire profiles continuously. A 3D image is displayed as soon as the specified number of height profiles have been acquired.
- Select Trigger on I/O 3 to start the image acquisition based on a digital input signal.
- Select Object Trigger to start the image acquisition when the object extends above a specified threshold.
- Select Software to display a Force Trigger button which enables triggering from the user interface.

Using a configuration

Click the Set Configuration button to apply the parameter changes. A green cog icon in the user interface indicates a valid configuration.

Press the Save configuration button to save the configuration to the camera's flash memory. To load the latest saved configuration, press the Load configuration button.

Image handling controls

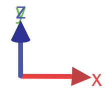
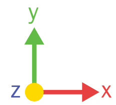
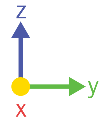
The image handling controls are used to scale, move, and rotate the 3D image in the viewer. As an alternative to the buttons, use the mouse shortcuts described below.

Button	Name	Description
	Select	Hover over a region to display the coordinates and corresponding pixel intensities. Click and drag to change the dimensions of the region of interest.
	Move	Click and drag to move the image. Mouse shortcut: Press and hold Shift.
	Rotate	Click and drag to rotate the image. Mouse shortcut: Press and hold Ctrl.
	Zoom	Click and drag upwards to zoom in and downwards to zoom out. Mouse shortcut: Use the mouse scroll wheel.

3D navigation control

Use the 3D navigation control in the lower left corner of the image viewer to change between orthographic 2D and 3D viewing angles:

- Click an arrowhead (X, Y or Z) to view an orthographic 2D projection of the object.
- Click the same arrowhead twice to flip the 2D projection, e.g. switch between the top and bottom view for the Z-axis.
- Press Home to restore the default viewing position.

View		Projection
Home		
Top view		
Right view		

Menus

The View, Color and Options menus in the image viewer contain options for visualization of the heightmap and the reflectance data.

- Select Reflectance from the View menu to display the heightmap with reflectance overlay. The reflectance image is useful for displaying surface details, such as prints.
- Select Surface or Points from the Options menu to display the data as a smooth surface or with a point representation.

Using SICK AppStudio

Tutorials

Tutorials for general and device-specific topics are available to help the user get started with the programming of the device. The relevant tutorials for TriSpectorP1000 are available in the SICK Support Portal, supportportal.sick.com.

Lua scripting

The embeddable scripting language Lua is used to create scripts in SICK AppStudio. See www.lua.org for more information about Lua.

Auto completion

SICK AppSpace has a large API which includes functionality for hardware configuration, algorithms, result processing and result communication. The API is directly accessible from the Lua scripts:

1. Click a free place in the lua file.
2. Press Ctrl+Space to display a list of all accessible functions and commands for the TriSpectorP1000.

The API documentation for TriSpectorP1000 is available in the SICK Support Portal, supportportal.sick.com.

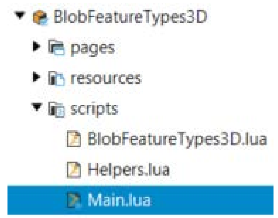
Sample apps for TriSpectorP1000

A sample app, also called programming sample, is a documented code snippet intended to help users understand how API functions or specific tasks work. The sample apps for the TriSpectorP1000 are available on the TriSpectorP1000 page in the SICK Support Portal, supportportal.sick.com. The ImageAcquisition and FirstSteps apps are recommended as an introduction to the programming of the TriSpectorP1000.

In this section, the sample app BlobFeatureTypes3D is used as an example for explaining concepts for 3D image handling. The app locates, analyzes and visualizes blobs (Binary Large Objects), which are defined as clusters of connected points within the heightmap.

Opening a sample app

1. Download the sample app as a .zip file from the SICK SupportPortal.
2. To import the app into SICK AppStudio, select Import from the File menu or drag and drop the file into the App Explorer.
3. In the AppExplorer, in the scripts folder, double-click a Lua file to open it in the editor.



App components

Scripts

An app includes one or more scripts, which contain the Lua source code. The main script, indicated by a blue arrow symbol in the AppExplorer, is executed when the app is run. To use a different script as main, right-click the lua file and select SET AS MAIN.

If multiple scripts exist for an app, use the `require()` command to execute and import the content of the other scripts to the main script. See the `Main.lua` script in the BLOBFEATURETYPES3D app for an example.

Resources

The RESOURCES folder stores files to be used by the app. For the BLOBFEATURETYPES3D app, the RESOURCES folder contains image data which is used as basis for the blob localization.

Pages

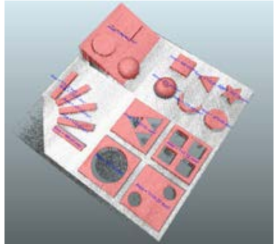
A page component is a web page (msdd file) which represents the user interface of the application. Go to the DEVICEPAGE tab to select which page component to view on the device page.

App overview

A script consists of a Global Scope, which runs every time the script is started, and a Function and Event Scope which holds the main part of the application. The Global Scope often includes a callback to the `main()` function:

```
Script.register("Engine.OnStarted", main)
```

The `Main.lua` script in the BLOBFEATURETYPES3D app gives an example of a script structure. The Global Scope contains a callback to the `main()` function, which in turn calls functions from the `BlobFeatureTypes3D.lua` script for localization and visualization of blobs. The output for each function is displayed in the image viewer, see examples below.



Output from the `area()` function

Output from the `perimeterLength()` function

App concepts

The SICK AppSpace Algorithm API contains a wide variety of functions and algorithms which can be used to fit a wide variety of applications. The BLOBFEATURETYPES3D app includes some examples on how to implement functions for blob localization and visualization.

Creating a view

The `Helpers.lua` script uses the `View.create()` function for displaying the heightmap and the blob overlays. A view is neccessary for all apps which display iconic objects, such as 2D and 3D images. When a view has been created, its content can be viewed in a web browser.

Limiting the search region

The `BlobFeatureTypes3D.lua` script uses the `thresholdPlane()` function for limiting the search region when locating features in the heightmap. By using different values and shapes as input to the `thresholdPlane()` function, the search region can be customized to fit the heightmap and application of interest.

Using overlay objects

The `Helpers.lua` script demonstrates how to create shape decorations with different colors and line types. Shape decorations are used for displaying the located blobs in this example, and are generally useful for highlighting features in 2D and 3D images.

Reading 3D images

The `Main.lua` script demonstrates the use of the `Object.load()` function to read images from the app's RESOURCES folder.

Algorithm for blob localization

The localization of blobs by the `findConnected()` function is an example of how to use an algorithm from the SICK AppSpace Algorithm API. See the SICK Support Portal for further examples of algorithms for locating and analyzing features in the heightmap.

Programming the TriSpectorP1000

This section contains examples of basic operations for programming the TriSpectorP1000.

Programming the LED indicators

There are five LED indicators on the back plate of the TriSpectorP1000 (for LED definitions, ➜ *See „B. LED definitions“ section on Page 6.*). The STATE and RESULT LEDs are programmable.

Creating a LED object

Create a Result or State LED object.

```
resultLED = LED.create("RESULT_LED")
stateLED = LED.create("STATE_LED")
```

Setting the LED color

The `setColor()` function is only applicable for the RESULT LED, which supports four different colors.

```
LED.setColor(resultLED, "green")
LED.setColor(resultLED, "red")
LED.setColor(resultLED, "yellow")
LED.setColor(resultLED, "blue")
```

Activating or deactivating a LED

Switch on (activate) or off (deactivate) the LED.

```
LED.activate(resultLED)
LED.deactivate(resultLED)
```

Making the LED blink

The `blink()` function makes the LED blink for a specified time (in milliseconds).

```
LED.blink(resultLED, 50)
```

Programming the digital I/O

There are seven digital inputs/outputs available via the Power-I/O connector on the TriSpectorP1000. For pin assignments, ➜ *See „D. Pin assignment“ section on Page 7.*

> **NOTICE**
> Input 1...3 can only be used as inputs, while inputs/outputs 4...7 are configurable and can be used for both input and output signals.

Creating an input object

```
di2 = Connector.DigitalIn.create("DI2")
```

Creating an output object

```
do4 = Connector.DigitalOut.create("DO4")
```

Setting the state of an output object

```
do4 = Connector.DigitalOut.
setLogic("ACTIVE_HIGH")
```

Reading the state of an input object

```
Connector.DigitalIn.get(di2)
```

Notification of a changed state

Register a callback function to get a notification of a changed state of an input object. In the example below, a changed state of input 2 calls the function `on_in2_change()`.

```
di2: register("OnChange", on_in2_change)
```

Connecting an encoder

When an encoder is connected, use the following functions to access the encoder and change the encoder settings.

Creating an encoder object

Create an encoder object by inserting the commands below.

```
encHandle = Encoder.create("ENC1")
incHandle = Connector.Increment.
create("INC1")
Encoder.setSource(encHandle, incHandle)
```

Setting the encoder mode

Use the `setCountMode()` function to set the encoder mode. The function's second argument is an enum representing the encoder mode:

```
Encoder.setCountMode(encHandle, "POSI-
TIVE_MOVEMENT")
```

For descriptions of the available encoder modes, ➜ *See „Encoder“ section on Page 2.*

Encoder settings

Set the number of ticks per millimeter in accordance with the encoder specification, by using the function below:

```
Image.Provider.Camera.V3TConfig3D.setEncod
erTicksPerMm(ConfigHandle, 50)
```

Loading and saving image data

Data acquired by the TriSpectorP1000 consists of a heightmap image, a reflectance image and sensor data. The sensor data contains frame numbers, time stamps and encoder values for the acquired height profiles.

Saving image data

During operation, it is recommended to use an FTP server for saving image data. Limited amounts of image data can be saved to the camera's flash memory for app development purposes.

In the example below, the image data is saved to the PUBLIC folder on the camera's file system. The PUBLIC folder can be accessed by all applications on the camera.

```
data = Object.serialize({heightMap, intensityMap, sensorData}, "JSON")

success = Object.save(data, "public/ImageData.json")
```

NOTICE

Do not use the flash memory for continuous data processing. The underlying hardware might get damaged.

Loading image data

Use the `Object.load()` function to load saved TriSpectorP1000 image data from the flash memory into SICK AppStudio:

```
data = Object.load("/public/ImageData.json")

heightMap = data[1]

reflectanceMap = data[2]

sensorData = data[3]
```

Loading TriSpector1000 images

To load an image acquired by a TriSpector1000 into SICK AppStudio, use the `TriSpector.readTriSpector1000Heightmap()` function:

```
heightmap, reflectance = TriSpector.readTriSpector1000Heightmap("public/ImageData.png")
```

License text

SICK uses open-source software. This software is licensed by the rights holders using the following licenses among others: the free licenses GNU General Public License (GPL Version2, GPL Version3) and GNU Lesser General Public License (LGPL), the MIT license, zLib license, and the licenses derived from the BSD license.

This program is provided for general use, but WITHOUT ANY WARRANTY OF ANY KIND. This warranty disclaimer also extends to the implicit assurance of marketability or suitability of the program for a particular purpose.

More details can be found in the GNU General Public License. View the complete license texts here: www.sick.com/licensetexts. Printed copies of the license texts are also available on request.

Australia
Phone +61 3 9457 0600
1800 334 802 – tollfree
Austria
Phone +43 22 36 62 28 8-0
Belgium/Luxembourg
Phone +32 2 466 55 66
Brazil
Phone +55 11 3215-4900
Canada
Phone +1 905 771 14 44
Czech Republic
Phone +420 2 57 91 18 50
Chile
Phone +56 2 2274 7430
China
Phone +86 20 2882 3600
Denmark
Phone +45 45 82 64 00
Finland
Phone +358-9-2515 800
France
Phone +33 1 64 62 35 00
Germany
Phone +49 211 5301-301
Hong Kong
Phone +852 2153 6300
Hungary
Phone +36 1 371 2680
India
Phone +91 22 6119 8900
Israel
Phone +972 4 6881000
Italy
Phone +39 02 274341
Japan
Phone +81 3 5309 2112
Malaysia
Phone +6 03 8080 7425
Mexico
Phone +52 (472) 748 9451
Netherlands
Phone +31 30 2044 000
New Zealand
Phone +64 9 415 0459
0800 222 278 – tollfree
Norway
Phone +47 67 81 50 00
Poland
Phone +48 22 539 41 00
Romania
Phone +40 356 171 120
Russia
Phone +7 495 775 05 30
Singapore
Phone +65 6744 3732
Slovakia
Phone +421 482 901201
Slovenia
Phone +386 591 788 49
South Africa
Phone +27 11 472 3733
South Korea
Phone +82 2 786 6321
Spain
Phone +34 93 480 31 00
Sweden
Phone +46 10 110 10 00
Switzerland
Phone +41 41 619 29 39
Taiwan
Phone +886 2 2375-6288
Thailand
Phone +66 2645 0009
Turkey
Phone +90 216 528 50 00
United Arab Emirates
Phone +971 4 88 65 878
United Kingdom
Phone +44 1727 831121
USA
Phone +1 800 325 7425
Vietnam
Phone +84 945452999

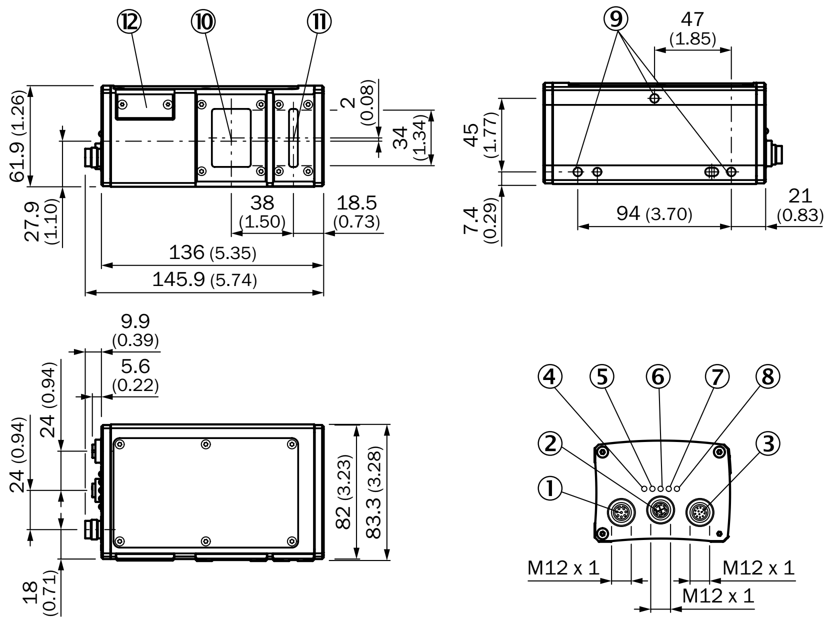
Detailed addresses and further locations at **www.sick.com**

SICK AG | Waldkirch | Germany | www.sick.com

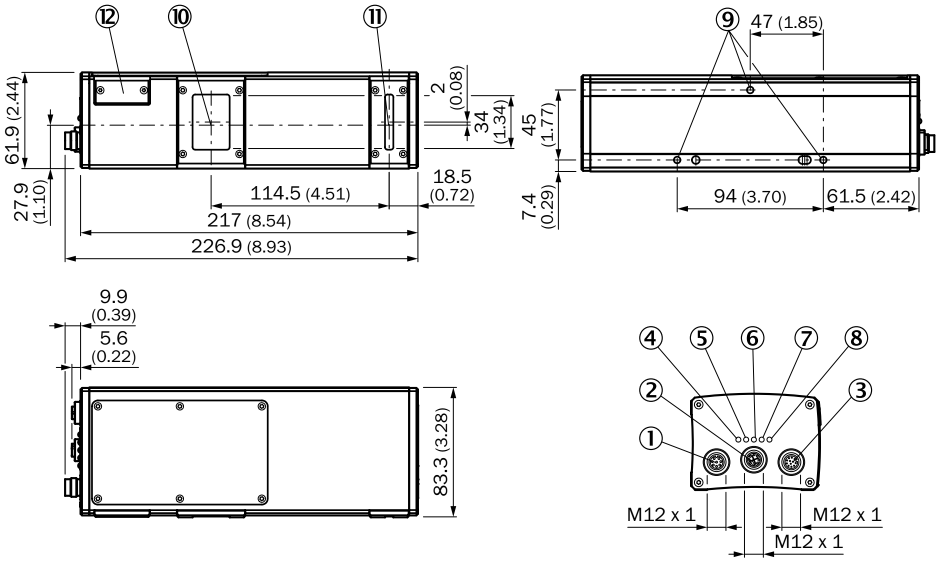
SICK
Sensor Intelligence.

A. Dimensional drawings

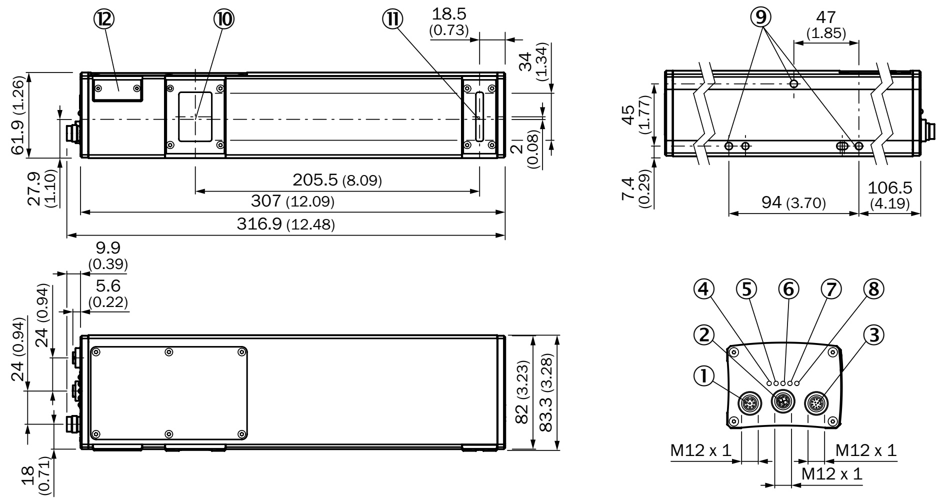
1. TriSpectorP1008 (Small FoV)



2. TriSpectorP1030 (Medium FoV)



3. TriSpectorP1060 (Large FoV)

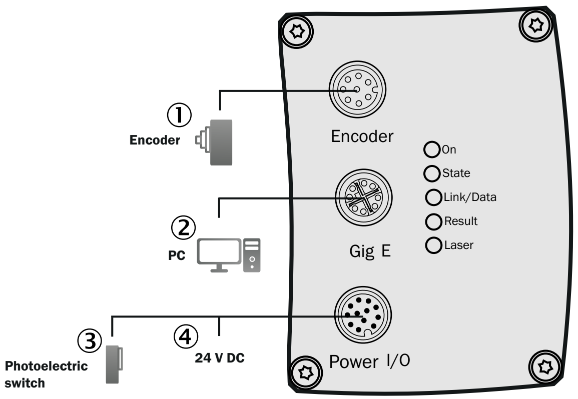


① Connector Encoder (thread inside)
② Connector Gigabit Ethernet (Gig E)
③ Connector Power I/O (thread outside)
④ LED; On
⑤ LED; State
⑥ LED; Link/Data
⑦ LED; Result
⑧ LED; Laser
⑨ Fastening threads (M5x8.5 length)
⑩ Optical receiver (center)
⑪ Optical sender (center)
⑫ SD-card

B. LED definitions

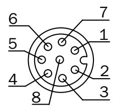
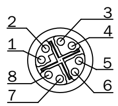
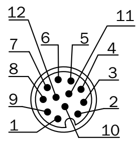
Name	Color	Function
On	Green	Power on
State	Green	Programmable
Link/Data	Green	Gigabit Ethernet (Gig E) Link: LED on Activity: LED blink
Result	Green	Programmable
	Red	Programmable
	Blue	Programmable
	Yellow	Programmable
Laser	Green	Laser on

C. Connection diagram



① Encoder
② PC/Network
③ Photoelectric switch
④ 24 V DC, power source

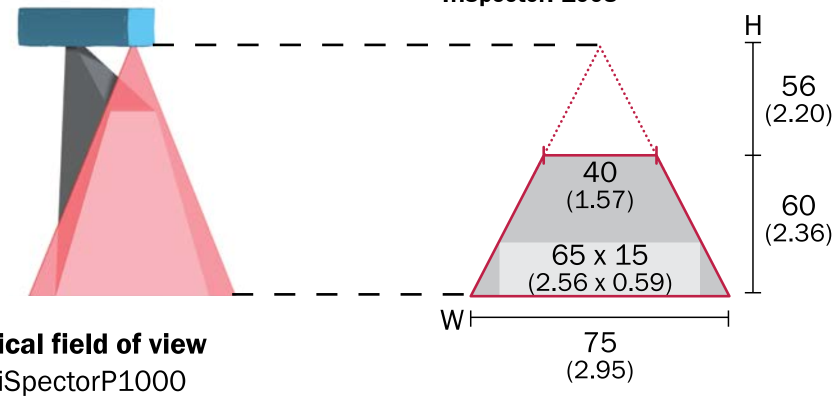
D. Pin assignment

	Encoder I/O	Gigabit Ethernet	Power I/O
	 M12 female connector, 8-pin A-coded	 M12 female connector, 8-pin X-coded	 M12 male connector, 12-pin A-coded
Pin	**Signal**	**Signal**	**Signal**
1	A/ - RS422 inverted input	GETH_L1+	24 V voltage supply input
2	A - RS422 non-inverted input	GETH_L1-	GND (power/signal)
3	B/ - RS422 inverted input	GETH_L2+	24 V - input 3 (trigger in)
4	B - RS422 non-inverted input	GETH_L3+	24 V - input/output 4 (configurable)
5	(Not connected)	GETH_L3-	24 V - input 2
6	(Not connected)	GETH_L2-	24 V - input/output 5 (configurable)
7	GND (power/signal)	GETH_L4+	24 V - input/output 6 (configurable)
8	24 V power supply output	GETH_L4-	24 V - input 1
9	-	-	24 V - input/output 7 (configurable)
10	-	-	Reserved
11	-	-	RS-232 Rx
12	-	-	RS-232 Tx
		(EIA / TIA568-B)	

E. Field of view diagrams

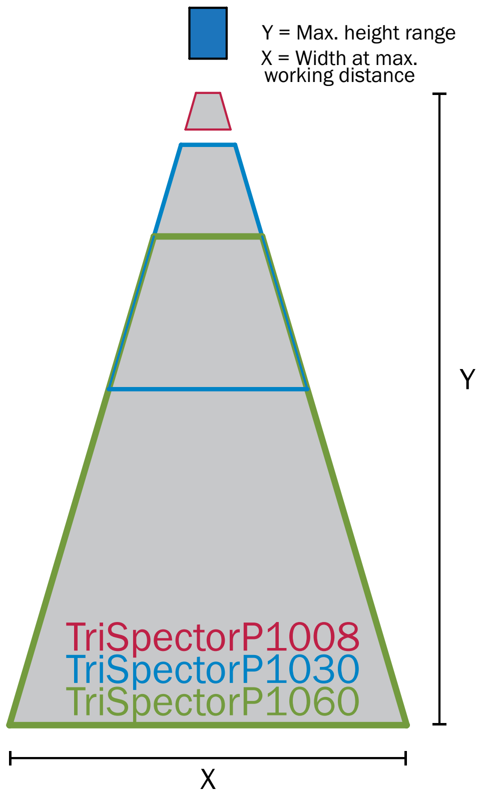
Maximum guaranteed image acquisition area in mm (inch). The brighter areas represent typical image acquisition areas.

TriSpectorP1008

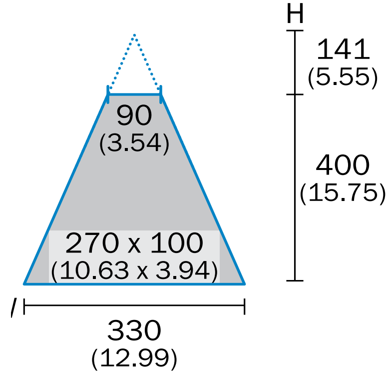


Typical field of view

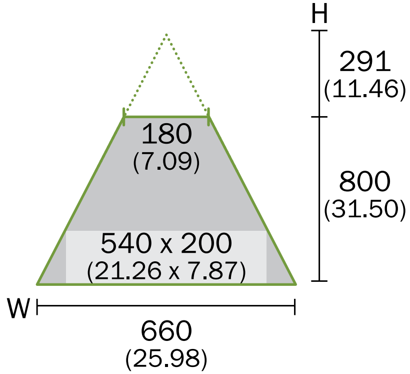
TriSpectorP1000



TriSpectorP1030



TriSpectorP1060



F. Technical data

Attribute	Value
Features	
Tasks	Positioning, inspection, measuring, reading
Technology	3D, LineScan, image analysis
Working distance (measured from front window)	1008: 56...116 mm range 1030: 141...541 mm range 1060: 291...1091 mm range
Example field of view	1008: 65 mm x 15 mm 1030: 270 mm x 100 mm 1060: 540 mm x 200 mm
Light source	Visible red light (laser, 660 nm)
Laser class	2 (EN/IEC 60825-1:2014) 2M (EN/IEC 60825-1:2007) Complies with 21 CFR 1040.10 except for deviations pursuant to Laser Notice No.50, dated June 24, 2007
Width at minimum operating-distance	1008: 40 mm 1030: 90 mm 1060: 180 mm
Width at maximum operating-distance	1008: 75 mm 1030: 330 mm 1060: 660 mm
Maximum height range	1008: 60 mm 1030: 400 mm 1060: 800 mm
Factory calibrated	Yes
Laser fan angle	45° ± 2°
Imaging angle	1008/1030: 65° 1060: 67°
Offline support	Emulator
Performance	
Scan/frame rate	2000 3D profiles/s
Maximum number of profiles	2500 per image
Height resolution	1008: 20 ... 50 µm 1030: 40 ... 280 µm 1060: 80 ... 670 µm
3D profile resolution	1008: 0.049 mm/px 1030: 0.215 mm/px 1060: 0.43 mm/px
Interfaces	
Operator interface	Web server
Configuration software	SICK AppStudio
Data storage and retrieval	Image and data logging via external FTP

Attribute	Value
Communication interfaces	Gigabit Ethernet (TCP/IP), EtherNet/IP, serial (RS-422), configurable digital I/O
Digital inputs	3 x, non-isolated
Digital inputs/ outputs	4 x, non-isolated, configurable
Encoder interface	RS-422/TTL
Maximum encoder frequency	300 kHz
Mechanics/electronics	
Connections	M12, 12-pin male connector, A-coded (voltage supply, I/O) M12, 8-pin female connector, X-coded (Gigabit Ethernet) M12, 8-pin female connector, A-coded (encoder)
Connector material	Nickel-plated brass
Supply voltage	24 V DC, ± 20 % SELV + LPS according to EN 60950-1:2014-08 or Class 2 according to UL1310 (6th Edition)
Supply current	1.5 A maximum; external fuse required
Power consumption	11 W maximum
Ripple	< 5 Vpp
Current consumption	< 400 mA with no output loads
Enclosure rating	IP 67
Safety	EN 60950-1:2014-08
Protection class	III
Housing material	Anodized aluminium
Window material	Glass/PMMA
Weight (not including cables)	1008: 900 g 1030: 1300 g 1060: 1700 g
Dimensions (L x W x H)	1008: 136 mm x 62 mm x 84 mm 1030: 217 mm x 62 mm x 84 mm 1060: 307 mm x 62 mm x 84 mm
Optics	Fixed

Attribute	Value
Ambient data	
EMC	Immunity: EN 61000-6-2:2005 Emission: EN 61000-6-3:2007
Shock load	15 g / 6 ms (EN 60068-2-27)
Vibration load	5 g, 10 Hz...150 Hz (EN 60068-2-6)
Ambient operating temperature	1008: 0 °C ... +45 °C 1030/1060: 0 °C ... +50 °C (See section „Mounting requirements“ on page 1 for information regarding adequate heat dissipation)
Ambient storage temperature	-20 °C ... +70 °C
Permissable relative air humidity	0% ... 90%, non-condensing

Input switching levels	
Input levels	Up to 30 V Stresses beyond this over-voltage level can cause permanent damage to the device
Input threshold levels	High: > 15.0 V Low: < 5.0 V
Hysteresis	> 1.0 V
Input current	High < 3.0 mA Low < 0.1 mA

Output switching levels	
Voltage level high	High: > voltage supply - 3.0 V (voltage supply = 19.2 V...28.8 V) Low: < 2.0 V
Source/sink output current	≤ 100 mA @ 24°C
Overcurrent protection	< 200 mA
Capacitive load	≤ 100 nF
Inductive load	≤ 1H (with use of external freewheeling diode, otherwise permanent damage to the device can occur)