OPERATING INSTRUCTIONS

TriSpector1000

3D vision

SICK
Sensor Intelligence.

Described product

TriSpector1000

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 Information on the operating instructions

These operating instructions provide important information on how to use devices from SICK AG.

Prerequisites for safe work are:

- Compliance with all safety notes and handling instructions supplied.
- Compliance with local work safety regulations and general safety regulations for device applications

The operating instructions are intended to be used by qualified personnel and electrical specialists.

NOTE
Read these operating instructions carefully to familiarize yourself with the device and its functions before commencing any work.

The instructions constitute an integral part of the product and are to be stored in the immediate vicinity of the device so they remain accessible to staff at all times. Should the device be passed on to a third party, these operating instructions should be handed over with it.

These operating instructions do not provide information on operating the machine or system in which the device is integrated. For information about this, refer to the operating instructions of the specific machine.

1.2 Explanation of symbols

Warnings and important information in this document are labeled with symbols. The warnings are introduced by signal words that indicate the extent of the danger. These warnings must be observed at all times and care must be taken to avoid accidents, personal injury, and material damage.

DANGER
... indicates a situation of imminent danger, which will lead to a fatality or serious injuries if not prevented.

WARNING
... indicates a potentially dangerous situation, which may lead to a fatality or serious injuries if not prevented.

CAUTION
... indicates a potentially dangerous situation, which may lead to minor/slight injuries if not prevented.

NOTICE
... indicates a potentially harmful situation, which may lead to material damage if not prevented.

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

1.3 Further information

NOTE
Further documentation for the device can be found on the online product page at:

- www.sick.com/TriSpector1000

There, additional information has been provided depending on the product, such as:

- Model-specific online data sheets for device variants, containing technical data, dimensional drawing, and specification diagrams
- EU declarations of conformity for the product family
- Dimensional drawings and 3D CAD dimension models in various electronic formats
- This documentation, in English and German and other languages if applicable
- Other publications related to the devices described here
- Publications dealing with accessories

2 Safety information

2.1 Intended use

The TriSpector1000 is an industrial 3D sensor that uses laser triangulation on objects to produce 3D images. Embedded 3D image analysis tools are applied to the 3D images. The results are sent to a control system via external interfaces.

The TriSpector1000 series is primarily designed for use in industrial and logistics areas, and it meets the requirements for industrial ruggedness, interfaces and data processing.

2.2 Improper use

Any use outside of the stated areas, in particular use outside of the technical specifications and the requirements for intended use, will be deemed to be incorrect use.

- The device does not constitute a safety component in accordance with the respective applicable safety standards for machines.
- The device must not be used in explosion-hazardous areas, in corrosive environments or under extreme environmental conditions.
- The device must not be operated in the temperature range below 0 °C.
- Any use of accessories not specifically approved by SICK AG is at your own risk.

WARNING
Danger due to improper use!

Any improper use can result in dangerous situations.

Therefore, observe the following information:

- Device should be used only in accordance with its intended use.
- All information in these operating instructions must be strictly observed.

2.3 Internet protocol (IP) technology

NOTE

SICK uses standard IP technology in its products. The emphasis is placed on availability of products and services.

SICK always assumes the following prerequisites:

- The customer ensures the integrity and confidentiality of the data and rights affected by its own use of the aforementioned products.
- In all cases, the customer implements the appropriate security measures, such as network separation, firewalls, virus protection, and patch management.

2.4 Limitation of liability

Relevant standards and regulations, the latest technological developments, and our many years of knowledge and experience have all been taken into account when compiling the data and information contained in these operating instructions. The manufacturer accepts no liability for damage caused by:

- Non-adherence to the product documentation (e.g., operating instructions)
- Incorrect use
- Use of untrained staff
- Unauthorized conversions

- Technical modifications
- Use of unauthorized spare parts, consumables, and accessories

With special variants, where optional extras have been ordered, or owing to the latest technical changes, the actual scope of delivery may vary from the features and illustrations shown here.

2.5 Modifications and conversions

NOTICE
Modifications and conversions to the device may result in unforeseeable dangers.

Interrupting or modifying the device or SICK software will invalidate any warranty claims against SICK AG. This applies in particular to opening the housing, even as part of mounting and electrical installation.

2.6 Requirements for skilled persons and operating personnel

WARNING
Risk of injury due to insufficient training.
Improper handling of the device may result in considerable personal injury and material damage.

- All work must only ever be carried out by the stipulated persons.

This product documentation refers to the following qualification requirements for the various activities associated with the device:

- **Instructed personnel** have been briefed by the operator about the tasks assigned to them and about potential dangers arising from improper action.
- **Skilled personnel** have the specialist training, skills, and experience, as well as knowledge of the relevant regulations, to be able to perform tasks delegated to them and to detect and avoid any potential dangers independently.
- **Electricians** have the specialist training, skills, and experience, as well as knowledge of the relevant standards and provisions to be able to carry out work on electrical systems and to detect and avoid any potential dangers independently. In Germany, electricians must meet the specifications of the BGV A3 Work Safety Regulations (e.g. Master Electrician). Other relevant regulations applicable in other countries must be observed.

The following qualifications are required for various activities:

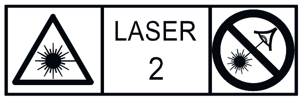
Table 1: Activities and technical requirements

Activities	Qualification
Mounting, maintenance	▪ Basic practical technical training ▪ Knowledge of the current safety regulations in the workplace
Electrical installation, device replacement	▪ Practical electrical training ▪ Knowledge of current electrical safety regulations ▪ Knowledge of the operation and control of the devices in their particular application
Commissioning, configuration,	▪ Basic knowledge of the Windows™ operating system in use ▪ Basic knowledge of the design and setup of the described connections and interfaces ▪ Basic knowledge of data transmission ▪ Knowledge of the programming of image-processing systems and network components

Activities	Qualification
Operation of the device for the particular application	■ Knowledge of the operation and control of the devices in their particular application ■ Knowledge of the software and hardware environment for the particular application

2.7 Warning symbols on the product

The product carries the following warning:



Laser radiation – Never look into the light beam – Laser class 2

If the laser warning label is concealed when the product is built into a machine or into paneling, a suitable additional laser warning label must be applied next to the laser output aperture on the machine or paneling.

2.8 Operational safety and particular hazards

Please observe the safety notes and the warnings listed here and in other chapters of this product documentation to reduce the possibility of risks to health and avoid dangerous situations.

The product is fitted with LEDs in risk group 0. The accessible radiation from these LEDs does not pose a danger to the eyes or skin.

If the product is operated in conjunction with external illumination systems, the risks described here may be exceeded. This must be taken into consideration by users on a case-by-case basis.

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 may increase the level of risk.
- Current national regulations regarding laser protection must be observed.

Caution – Use of controls or adjustments or performance of procedures other than those specified herein may result in hazardous radiation exposure.

WARNING
Electrical voltage!

Electrical voltage can cause severe injury or death.

- Work on electrical systems must only be performed by qualified electricians.
- The power supply must be disconnected when attaching and detaching electrical connections.
- The product must only be connected to a voltage supply as set out in the requirements in the operating instructions.
- National and regional regulations must be complied with.
- Safety requirements relating to work on electrical systems must be complied with.

WARNING
Risk of injury and damage caused by potential equalization currents!

Improper grounding can lead to dangerous equipotential bonding currents, which may in turn lead to dangerous voltages on metallic surfaces, such as the housing. Electrical voltage can cause severe injury or death.

- Work on electrical systems must only be performed by qualified electricians.
- Follow the notes in the operating instructions.
- Install the grounding for the product and the system in accordance with national and regional regulations.

3 Product description

3.1 System overview

The TriSpector1000 uses laser triangulation on objects to produce 3D images. Embedded 3D image analysis tools are applied to the 3D images. The results are sent to a control system via external interfaces.

Laser triangulation 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.

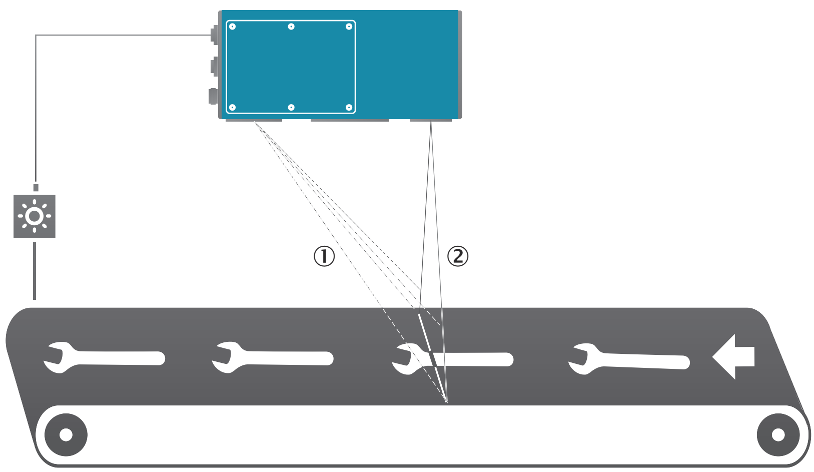


Figure 1: Laser triangulation principle

① Field of view

② Laser

Figure 2: Height profile

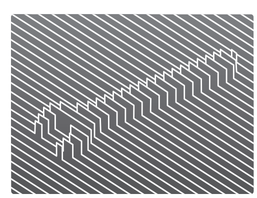


Figure 3: 3D image

Definitions

The **3D image**, which is 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 in parallel.

3.2 Scope of delivery

The delivery of the device includes the following components:

Table 2: Scope of delivery

Item	Component	Comments
1	Device in the version ordered	Depending on version

Item	Component	Comments
1	Set of class 2 laser warning labels in German/US English and French/US English	Self-adhesive to be applied over the warning label on the housing of the device (if necessary)
1	Printed safety notes, multilin-gual	Short information and general safety notes

Accessories

Accessories, such as brackets and connecting cables, are only supplied if ordered separately (see "Accessories", page 81).

3.3 System requirements

For adequate SOPAS ET performance, use a PC with Intel Core I5 540M (2.53 GHz, 4 GB RAM) or better, and a screen resolution of at least 1024x768.

Graphics card: Intel® HD Graphics video card (or NVIDIA® NVS 3100M 512MB GDDR3), or better. Make sure to use the latest graphic card drivers.

Ethernet connection is required, 100 Mbit/s or better. 1 Gbit/s or faster is recom-mended for best performance during configuration of the device via SOPAS ET.

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

SOPAS ET version: 2018.2 or higher

Hard drive: Minimum 550 MB.

Supported web browsers: Google Chrome (recommended due to advanced support of WebGL and WebSockets) and Microsoft Edge.

SD memory card (optional): SICK microSD memory card (part no 4051366 or 4077575).

A mouse with at least three buttons (or a scroll wheel) is recommended.

3.4 Device variants

The device is available with three different field of view (FoV) sizes and two different window materials. PMMA is a plastic material used as an alternative material to glass in food processing environments.

Table 3: TriSpector1008 (Small FoV)

Window material	No.
Glass	1075604
PMMA	1060426

Table 4: TriSpector1030 (Medium FoV)

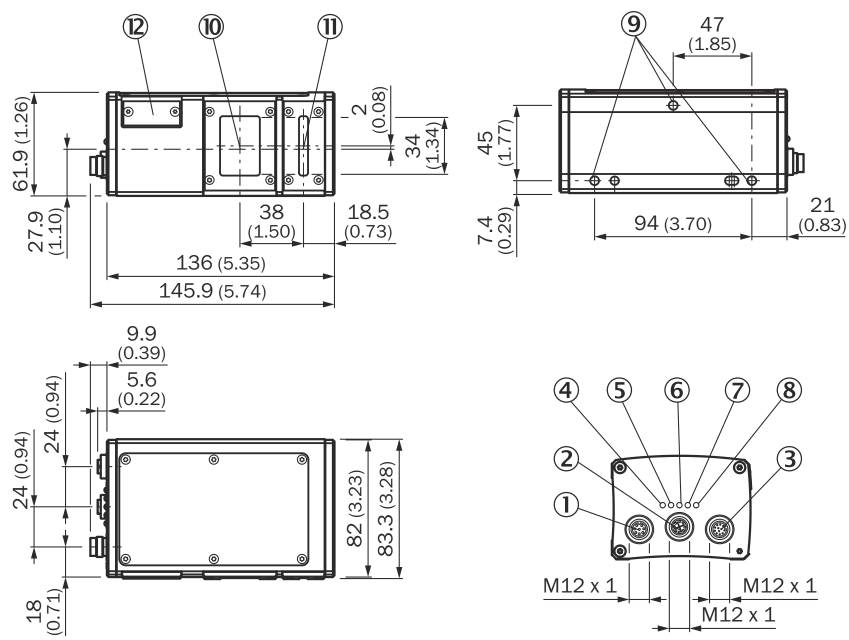
Window material	No.
Glass	1072923
PMMA	1060427

Table 5: TriSpector1060 (Large FoV)

Window material	No.
Glass	1075605
PMMA	1060428

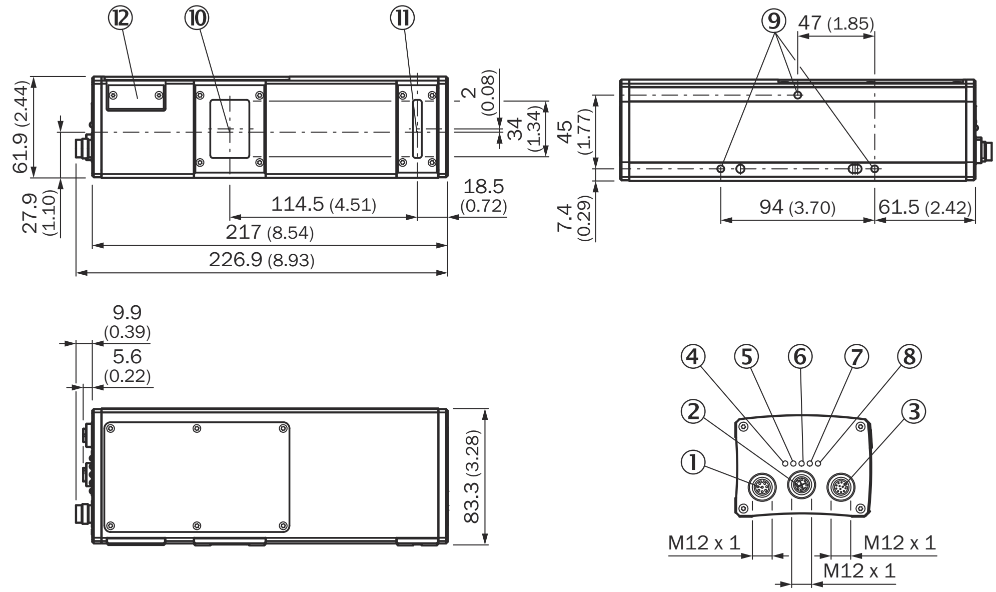
3.5 Dimensional drawings

NOTE
The dimensional drawings can also be downloaded as CAD files from the online product page:
www.sick.com/TriSpector1000

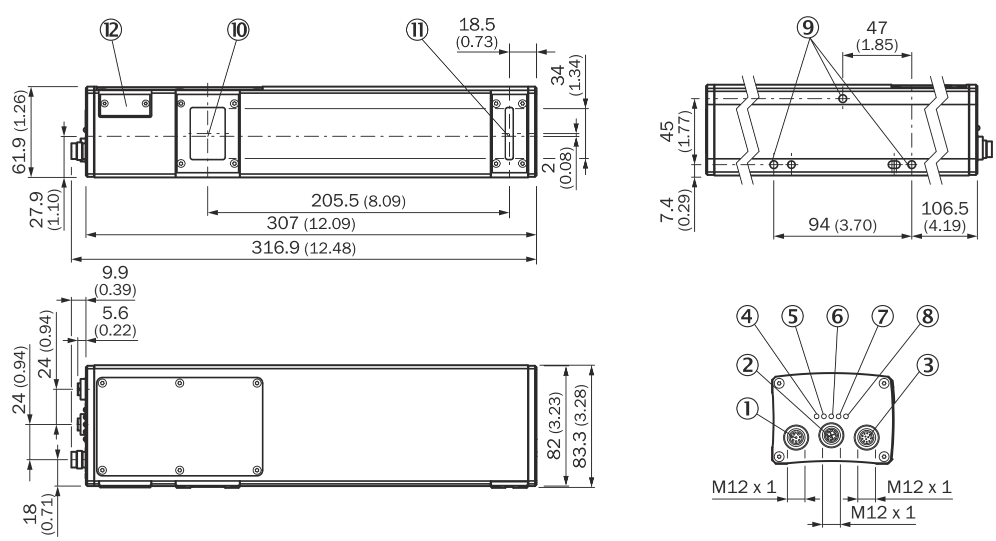


① Connector Encoder (thread inside)
② Connector Gigabit Ethernet (Gig E)
③ Connector Power I/O (thread outside)
④ LED: On
⑤ LED: State
⑥ LED: Link/Data
⑦ LED: Result
⑧ LED: Laser
⑨ Fastening threads (M5x8.5 length)
⑩ Optical receiver (center)
⑪ Optical sender (center)
⑫ SD-card

TriSpector1030

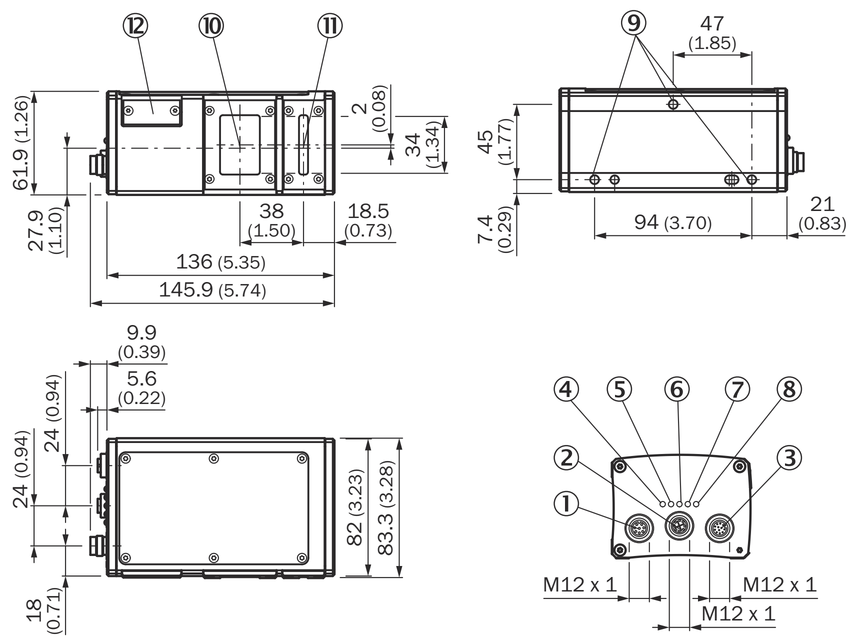


① Connector Encoder (thread inside)
② Connector Gigabit Ethernet (Gig E)
③ Connector Power I/O (thread outside)
④ LED: On
⑤ LED: State
⑥ LED: Link/Data
⑦ LED: Result
⑧ LED: Laser
⑨ Fastening threads (M5x8.5 length)
⑩ Optical receiver (center)
⑪ Optical sender (center)
⑫ SD-card



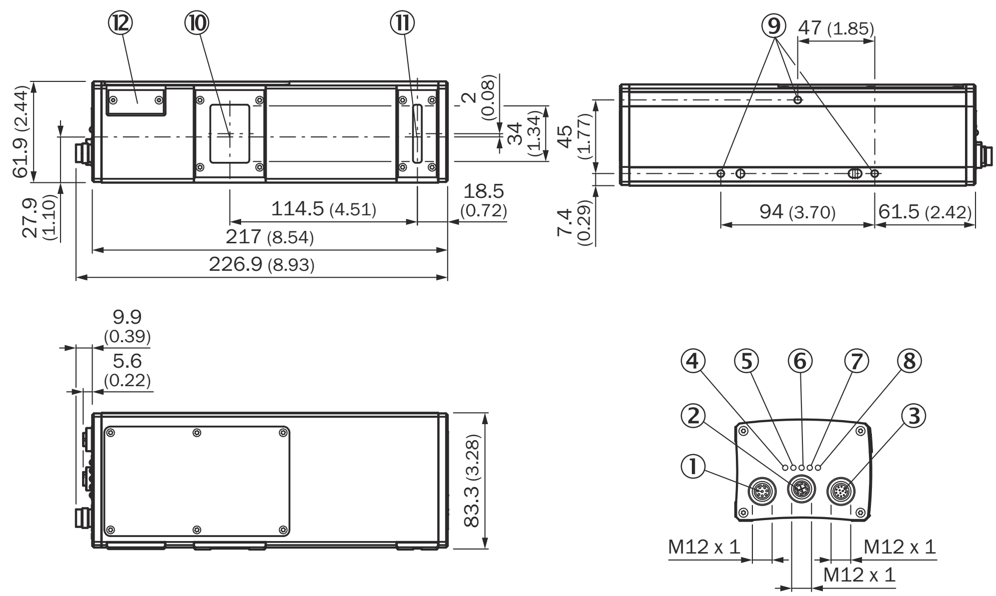
① Connector Encoder (thread inside)
② Connector Gigabit Ethernet (Gig E)
③ Connector Power I/O (thread outside)
④ LED: On

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

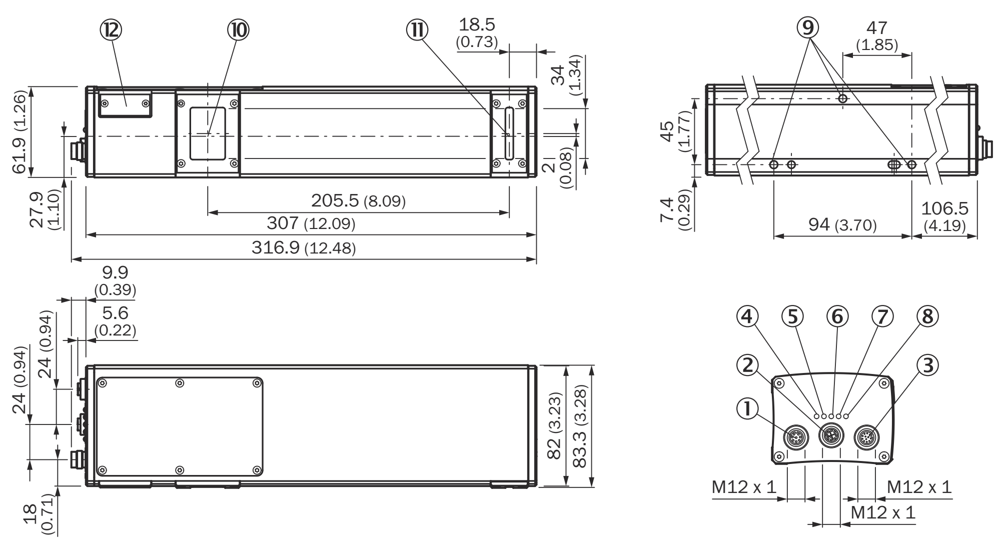


① Connector Encoder (thread inside)
② Connector Gigabit Ethernet (Gig E)
③ Connector Power I/O (thread outside)
④ LED: On
⑤ LED: State
⑥ LED: Link/Data
⑦ LED: Result
⑧ LED: Laser
⑨ Fastening threads (M5x8.5 length)
⑩ Optical receiver (center)
⑪ Optical sender (center)
⑫ SD-card

TriSpector1030



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



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

3.6 LED indicators

Name	Color	Function
On	Green	Power on
State	Green	Ready for trig input and image acquisition
Link/Data	Green	Gigabit Ethernet (Gig E) Link: LED on Activity: LED blink
Result	Green	Overall result: Pass
	Red	Overall result: Fail
	Blue	Result not found
	No light	Result invalid or neutral
Laser	Green	Laser on

4 Transport and storage

4.1 Transport

For your own safety, please read and observe the following notes:

NOTICE
Damage to the product due to improper transport.

- The device must be packaged for transport with protection against shock and damp.
- Recommendation: Use the original packaging as it provides the best protection.
- Transport should be performed by trained specialist staff only.
- The utmost care and attention is required at all times during unloading and transportation on company premises.
- Note the symbols on the packaging.
- Do not remove packaging until immediately before you start mounting.

4.2 Unpacking

- Before unpacking, it may be necessary to equalize the temperature to protect the device from condensation.
- Handle the device with care and protect it from mechanical damage.
- Remove the protective caps on the electrical connections immediately before connecting the connecting cable to prevent dirt and water from entering.

4.3 Transport inspection

Immediately upon receipt in Goods-in, check the delivery for completeness and for any damage that may have occurred in transit. In the case of transit damage that is visible externally, proceed as follows:

- Do not accept the delivery or only do so conditionally.
- Note the scope of damage on the transport documents or on the transport company's delivery note.
- File a complaint.

NOTE
Complaints regarding defects should be filed as soon as these are detected. Damage claims are only valid before the applicable complaint deadlines.

4.4 Storage

Store the device under the following conditions:

- Recommendation: Use the original packaging.
- Electrical connections are provided with protective caps and plugs (as they are on delivery).
- Do not store outdoors.
- Store in a dry area that is protected from dust.
- So that any residual damp can evaporate, do not package in airtight containers.
- Do not expose to any aggressive substances.
- Protect from sunlight.
- Avoid mechanical shocks.
- Storage temperature: see "Technical data", page 78.

- Relative humidity: see "Technical data", page 78.
- For storage periods of longer than 3 months, check the general condition of all components and packaging on a regular basis.

5 Mounting

5.1 Mounting instructions

- Observe the technical data.
- To prevent condensation, avoid exposing the device to rapid changes in temperature.
- The mounting site has to be designed for the weight of the device.
- It should be mounted so that it is exposed to as little shock and vibration as possible. Optional mounting accessories are available, see "Accessories", page 81.
- Use a stable bracket with sufficient load-bearing capacity and suitable dimensions for the device.
- Ensure that the device has a clear view of the object to be detected.
- Protect the device from moisture, contamination, and damage.
- Ensure a sufficient level of cooling using ambient air/convection and/or heat dissipation through mechanical mounting. Observe the permitted operating temperature, see "Technical data", page 78.

5.2 Mounting the device

Mount the TriSpector1000 in a position above the surface to be scanned. See "Field of view diagrams", page 22 for field of view diagrams and mounting distances. The default scan direction is shown in figure 4. If a scan is performed in the opposite direction, the acquired image will be mirrored.

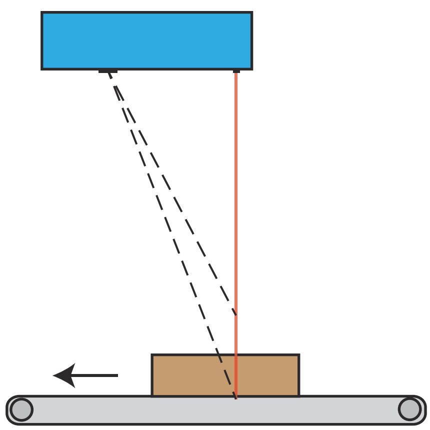


Figure 4: Mounting position

5.3 Field of view diagrams

The maximum guaranteed image acquisition area in mm (inch) for each TriSpector1000 variant is shown in figure 5. The brighter areas represent typical image acquisition areas.

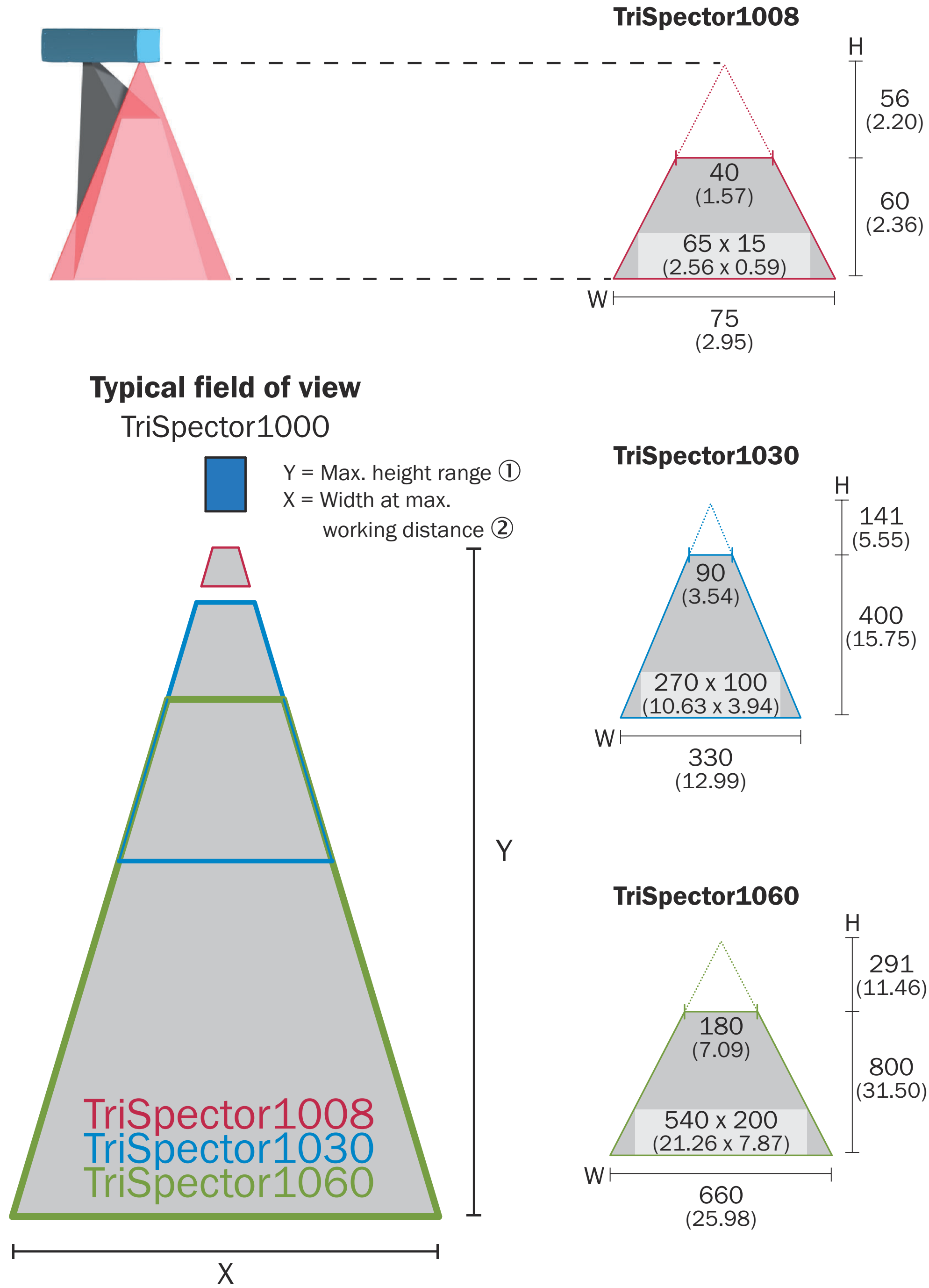


Figure 5: Field of view diagrams

① Maximum height range

② Width at maximum working distance

5.4 Mounting a microSD memory card

NOTICE

If the TriSpector1000 is powered on, go to the **SD Card** section under the **Interfaces** workflow step in SOPAS ET and click **Eject** before removing the microSD memory card.

NOTICE
The screws for the microSD memory card lid must be tightened with a torque of 0.3 Nm to keep the IP65 classification.

Follow the steps below to insert or remove a microSD memory card. The microSD memory card slot is located next to the camera window.

1. Release the two screws to remove the lid.
2. Insert or remove the microSD memory card.
3. Re-mount the lid and tighten the screws with a torque of 0.3 Nm.

For a specification of supported microSD memory cards, see "System requirements", page 14.

5.5 Replacing the windows

The laser window and the camera window on the device are replaceable and available as spare parts. Each spare part contains one window (in glass or PMMA) and four screws.

Figure 6: Laser (sender) window to the left, camera (receiver) window to the right

The following spare parts are available:

Glass windows
- Laser window: 2078269
- Camera window: 2078266

PMMA windows
- Laser window: 2078268
- Camera window: 2078267

Mounting instructions

General:
- Switch off the power to the device before replacing the window.
- At delivery, the PMMA spare part windows are sealed with a white protective foil. Remove the foil before replacing the window.
- Replace the window in a clean area, to avoid getting dust or dirt inside the camera housing.

To keep the IP67 classification:

- Make sure that the protective seal around the window is correctly re-assembled.
- Tighten the screws with a torque of 0.7 - 0.8 Nm.

6 Electrical installation

6.1 Wiring instructions

NOTE
Pre-assembled cables can be found online at:

- www.sick.com/TriSpector1000

NOTICE
Faults during operation and device or system defects!
Incorrect wiring may result in operational faults and defects.

- Follow the wiring notes precisely.

Recommendation: use shielded cables.

The electrical connections of the device are configured as M12 round connectors.

The protection class stated in the technical data is achieved only with screwed plug connectors or protective caps.

Configure the circuits connected to the device as ES1 circuits or as SELV circuits (SELV = Safety Extra Low Voltage). The voltage source must meet the requirements of ES1 and PS2 (EN 62368-1) or SELV and LPS (EN 60950-1).

Connect the connecting cables in a de-energized state. Do not switch on the supply voltage until installation is complete and all connection work on the device and controller has been finished.

Wire cross-sections in the supply cable from the customer's power system must be implemented in accordance with the applicable standards.

In the case of open end cables, make sure that bare wire ends do not touch. Wires must be appropriately insulated from each other.

If the supply voltage for the device is not supplied via the CDB650-204 connection module, the device must be protected by a separate slow-blow fuse at the start of the supply circuit. The required fuse rating (nominal value) is maximum 2 A slow-blow.

Shielding requirements

- To ensure a fault-free data transmission, an effective and comprehensive shielding solution must be implemented.
- Apply a cable shield at each end, i.e. in the control cabinet and at the device. The cable shield of the pre-assembled cables is connected to the knurled nut and thus also to a large area of the device housing.
- The cable shield in the control cabinet must be connected to a large area of the signal ground.
- Take appropriate measures to prevent equipotential bonding currents flowing through the cable shield.
- During installation, pay attention to the different cable groups. The cables are grouped into the following four groups according to their sensitivity to interference or radiated emissions:
  - Group 1: cables very sensitive to interference, such as analog measuring cables
  - Group 2: cables sensitive to interference, such as device cables, communication signals, bus signals

- Group 3: cables that are a source of interference, such as control cables for inductive loads and motor brakes
- Group 4: cables that are a powerful source of interference, such as output cables from frequency inverters, welding system power supplies, power cables

▶ Cables in groups 1, 2 and 3, 4 must be crossed at right angles (see figure 7).

▶ Route the cables in groups 1, 2 and 3, 4 in different cable channels or use metallic separators (see figure 8 and see figure 9). This applies particularly if cables of devices with a high level of radiated emission, such as frequency converters, are laid parallel to device cables.

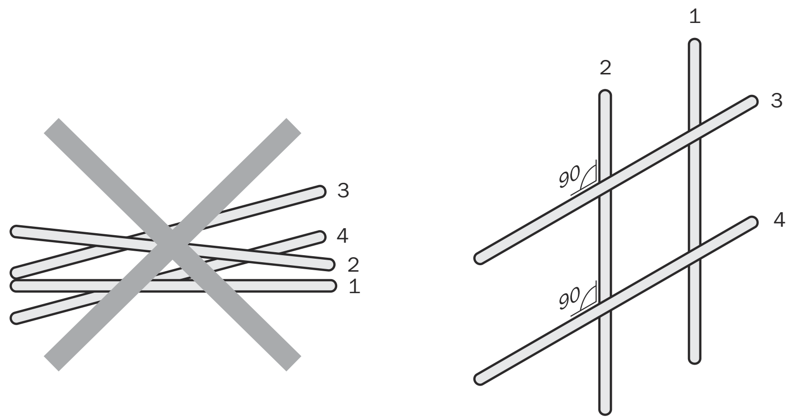


Figure 7: Cross cables at right angles

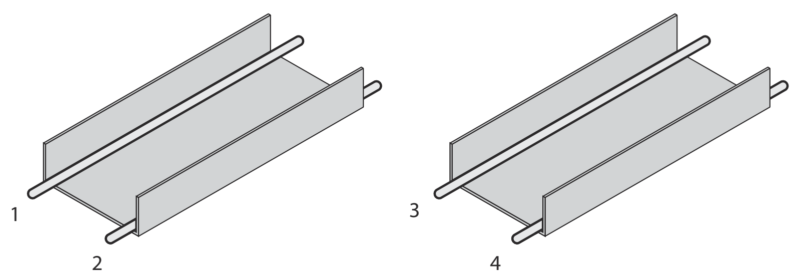


Figure 8: Ideal laying – Place cables in different cable channels

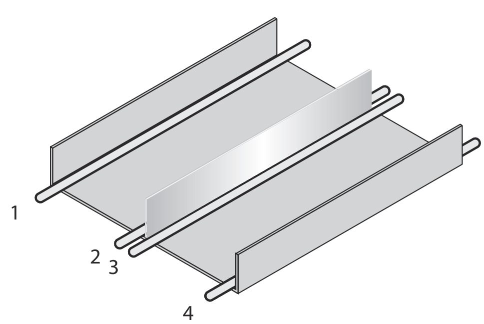


Figure 9: Alternative laying – Separate cables with metallic separators

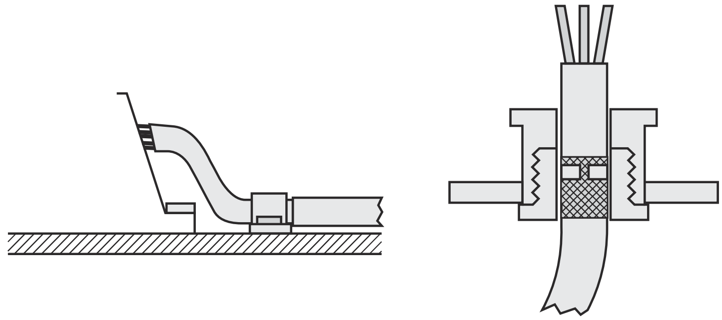


Figure 10: Shield connection in plastic housings

NOTE
Use an appropriate earthing method to prevent equipotential bonding currents flowing through the cable shield. If necessary, ground currents on the EtherNet/IP cabling can be prevented by using an EtherNet/IP adapter (part no. 2044264).

6.2 Prerequisites for safe operation of the device

WARNING
Risk of injury and damage caused by electrical current!
As a result of equipotential bonding currents between the device and other grounded devices in the system, faulty grounding of the device can give rise to the following dangers and faults:

- Dangerous voltages are applied to the metal housings.
- Devices will behave incorrectly or be destroyed.
- Cable shielding will be damaged by overheating and cause cable fires.

Remedial measures

- Only skilled electricians should be permitted to carry out work on the electrical system.
- If the cable insulation is damaged, disconnect the voltage supply immediately and have the damage repaired.
- Ensure that the ground potential is the same at all grounding points.
- Where local conditions do not meet the requirements for a safe earthing method, take appropriate measures (e.g., ensuring low-impedance and current-carrying equipotential bonding).

The device is connected to the peripheral devices (voltage supply, any local trigger sensor(s), system controller) via shielded cables. The cable shield – for the data cable, for example – rests against the metal housing of the device. The device can be grounded through the cable shield or through a blind tapped hole in the housing, for example.

If the peripheral devices have metal housings and the cable shields are also in contact with their housings, it is assumed that all devices involved in the installation have the **same ground potential**.

This is achieved by complying with the following conditions:

- Mounting the devices on conductive metal surfaces
- Correctly grounding the devices and metal surfaces in the system
- If necessary: low-impedance and current-carrying equipotential bonding between areas with different ground potentials

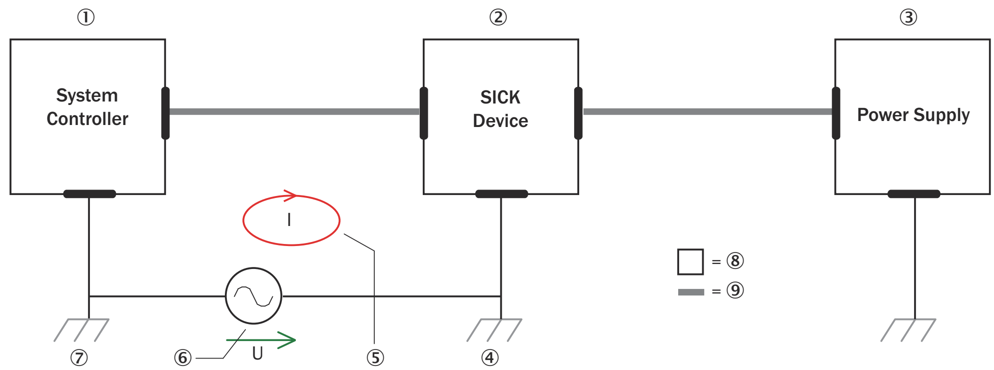


Figure 11: Example: Occurrence of equipotential bonding currents in the system configuration

① System controller
② Device
③ Voltage supply
④ Grounding point 2
⑤ Closed current loop with equalizing currents via cable shield
⑥ Ground potential difference
⑦ Grounding point 1
⑧ Metal housing
⑨ Shielded electrical cable

If these conditions are not fulfilled, equipotential bonding currents can flow along the cable shielding between the devices due to differing ground potentials and cause the hazards specified. This is, for example, possible in cases where there are devices within a widely distributed system covering several buildings.

Remedial measures

The most common solution to prevent equipotential bonding currents on cable shields is to ensure low-impedance and current-carrying equipotential bonding. If this equipotential bonding is not possible, the following solution approaches serve as a suggestion.

NOTICE
We expressly advise against opening up the cable shields. This would mean that the EMC limit values can no longer be complied with and that the safe operation of the device data interfaces can no longer be guaranteed.

Measures for widely distributed system installations

On widely distributed system installations with correspondingly large potential differences, the setting up of local islands and connecting them using commercially available **electro-optical signal isolators** is recommended. This measure achieves a high degree of resistance to electromagnetic interference.

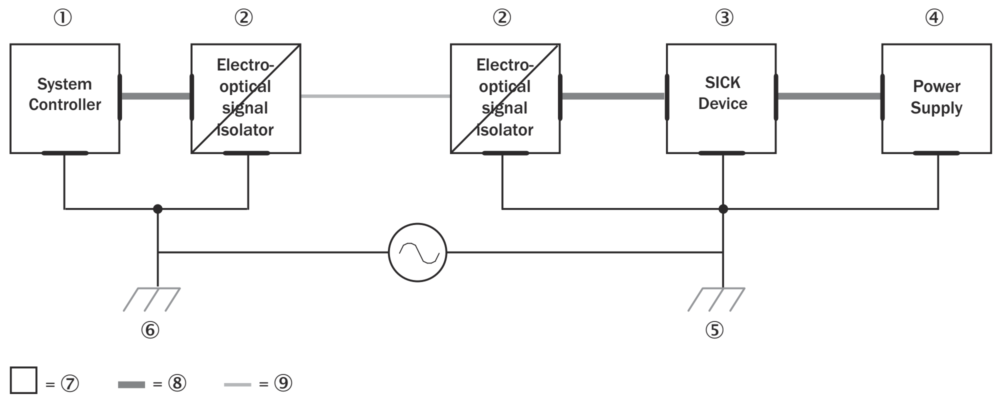


Figure 12: Example: Prevention of equipotential bonding currents in the system configuration by the use of electro-optical signal isolators

① System controller
② Electro-optical signal isolator
③ Device
④ Voltage supply
⑤ Grounding point 2
⑥ Grounding point 1
⑦ Metal housing
⑧ Shielded electrical cable
⑨ Optical fiber

The use of electro-optical signal isolators between the islands isolates the ground loop. Within the islands, a stable equipotential bonding prevents equalizing currents on the cable shields.

Measures for small system installations

For smaller installations with only slight potential differences, insulated mounting of the device and peripheral devices may be an adequate solution.

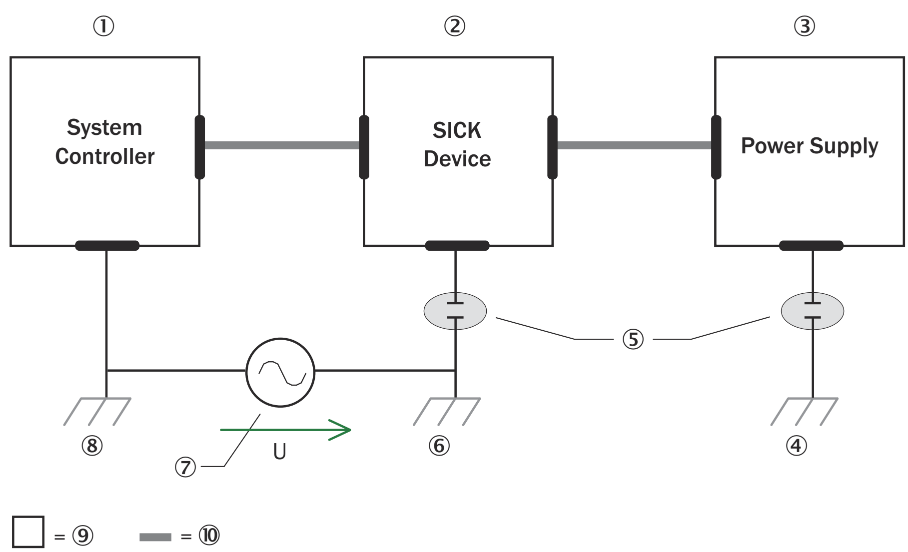


Figure 13: Example: Prevention of equipotential bonding currents in the system configuration by the insulated mounting of the device

① System controller
② Device
③ Voltage supply

④ Grounding point 3
⑤ Insulated mounting
⑥ Grounding point 2
⑦ Ground potential difference
⑧ Grounding point 1
⑨ Metal housing
⑩ Shielded electrical cable

Even in the event of large differences in the ground potential, ground loops are effectively prevented. As a result, equalizing currents can no longer flow via the cable shields and metal housing.

! **NOTICE**
The voltage supply for the device and the connected peripheral devices must also guarantee the required level of insulation.
Under certain circumstances, a tangible potential can develop between the insulated metal housings and the local ground potential.

6.3 Connection diagram

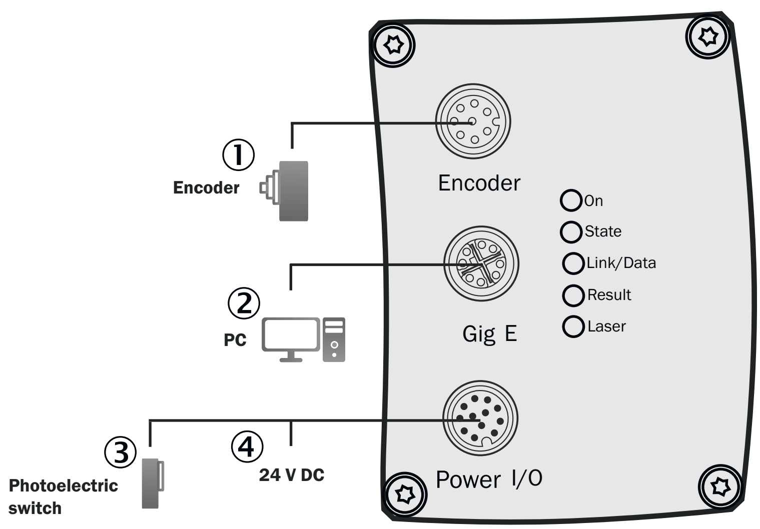


Figure 14: Connection diagram

① Encoder
② PC/Network
③ Photoelectric switch
④ 24 V DC, voltage supply

6.4 Encoder

! **NOTICE**
It is strongly recommended to use an encoder for measuring applications, such as 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.
- The encoder must have a RS-422/TTL interface. In the case of strong magnetic fields in proximity to the TriSpector1000, use a recommended encoder (no. 1068997) to ensure optimal performance.
- The connection requires two encoder channels (A/A¯ and B/B¯) to keep track of movement and direction.

6.5 Pin assignment

Encoder I/O

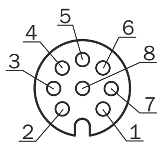


Figure 15: M12 female connector, 8-pin A-coded

Pin	Signal
1	A/ - RS-422 inverted input
2	A - RS-422 non-inverted input
3	B/ - RS-422 inverted input
4	B - RS-422 non-inverted input
5	(Not connected)
6	(Not connected)
7	GND (Power / Signal)
8	24 V Voltage supply output

Gigabit Ethernet

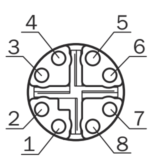


Figure 16: M12 female connector, 8-pin X-coded

Pin	Signal
1	GETH_L1+
2	GETH_L1-
3	GETH_L2+
4	GETH_L3+
5	GETH_L3-
6	GETH_L2-
7	GETH_L4+
8	GETH_L4-

Power I/O

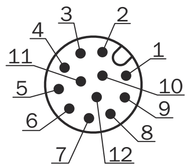


Figure 17: M12 male connector, 12-pin A-coded

Pin	Signal
1	24 V Voltage supply input
2	Ground (Power / Signal)
3	24 V - I/O 3, Trigger in
4	24 V - I/O 4, Configurable
5	24 V - I/O 2, Input
6	24 V - I/O 5, Configurable
7	24 V - I/O 6, Configurable
8	24 V - I/O 1, Input
9	24 V - I/O 7, Configurable
10	Reserved
11	RS-232 Rx
12	RS-232 Tx

7 Operation

7.1 Commissioning

7.1.1 Installing SOPAS

The SOPAS Engineering Tool (ET) software for PC is used to connect and configure SICK devices.

To install SOPAS ET:

1. Download the latest version of SOPAS ET from www.sick.com. Version 2018.2 or higher is required.
2. Run the downloaded installation file.
3. Follow the instructions on the screen.

7.1.2 Connecting the hardware

1. Connect the Ethernet cable between the TriSpector1000 and the PC.
2. Connect the TriSpector1000 Power I/O cable to a 24 V DC voltage supply:
   Brown: +24 V DC, Pin 1
   Blue: Ground, Pin 2
3. Connect the encoder cable between the TriSpector1000 and the encoder (if applicable).

7.1.3 Connecting the TriSpector1000 to SOPAS ET

1. Start SOPAS ET.

✓ The SOPAS ET main window opens. The main window is split into two panes, the project pane to the left and the device pane to the right.

NOTICE
At the first use, the TriSpector1000 requires a SICK Device Driver (SDD). When adding the device to a project, a prompt will appear with instructions on how to install the driver.

NOTE
In the device list, the TriSpector1000 device is available on port 2111 and 2112. Do not use port 2111. This port is used for debugging and may decrease the operation speed.

2. To add the TriSpector1000 to the SOPAS project, select the device on port 2112 in the device pane and click **Add**.

✓ A product icon appears in the project pane.

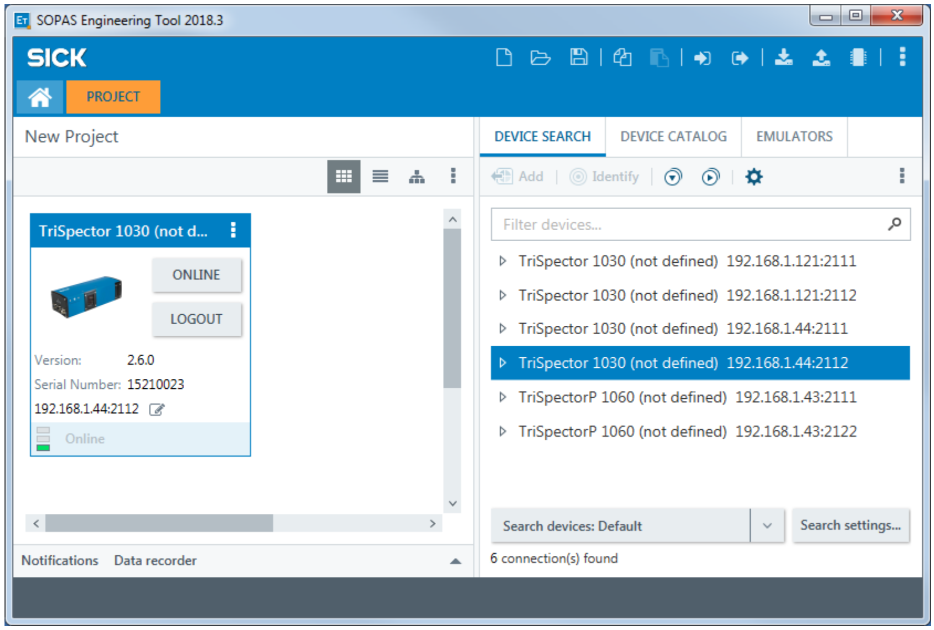


Figure 18: SOPAS ET main window

3. Double-click the product icon to open the SOPAS ET device window and start the configuration.
4. If there are IP address connection issues, click the Edit icon in the device tile to make adjustments. The default IP address is 192.168.0.30.

7.2 Description of the user interface

The TriSpector1000 is configured through the SOPAS ET device window.

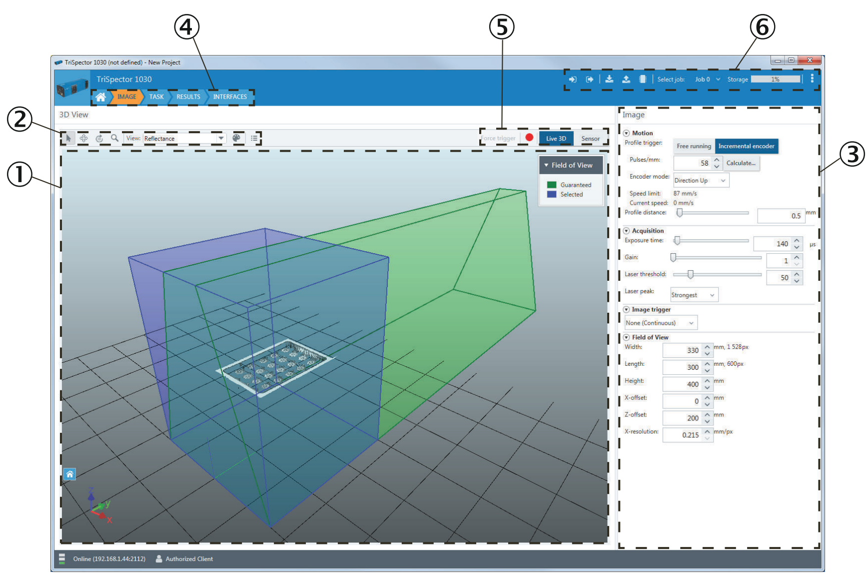


Figure 19: SOPAS ET device window

① Image area
② Image handling controls and view options
③ Parameter pane
④ Workflow steps
⑤ Controls for image view, image recording (red), and **Force trigger**
⑥ SOPAS ET functions panel

7.2.1 Menus

Three menus are available in the SOPAS ET functions panel: the **Device** menu, the **Parameter** menu and the **Functions** menu.

Device menu

Go online	Establish a connection between SOPAS ET and the TriSpector1000.
Go offline	Switch the connection mode to offline. In this mode, parameter values cannot be written to the device and measurements cannot be monitored. The connection information in SOPAS ET remains intact.
Switch to physical device	Switch from the emulator to the TriSpector1000 and import the configuration. This option is only enabled when running the emulator. For details, see "Copying a configuration from the emulator", page 72.
Switch to simulated device	Switch from the TriSpector1000 to the emulator. For details, see "Starting the emulator", page 71.
Login...	Log in to the device at a certain user level. When a device is added to a SOPAS project, the user level is automatically set to **Authorized Client**, which is the required level for saving parameter values to the device. The current user level is shown below the image area.
Logout	Cancel the login. Once the user has logged out, parameters secured by the user level can no longer be changed.
Import...	Import a saved device configuration from a PC. For details, see "Exporting and importing a configuration", page 72.
Export...	Export the current device configuration to a PC. For details, see "Exporting and importing a configuration", page 72.
Create PDF	Create a PDF which includes device parameters and their values.
Properties	Open the **Device Settings** window.
Close	Close the SOPAS ET device window.

Parameter menu

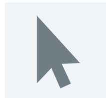
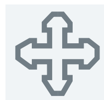
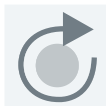
Write to device	Write the configuration saved in the SOPAS project to the device.
Read from device	Read the configuration from the device and saves it in the SOPAS project.
Save permanent	Save the configuration in the device flash memory and to the microSD memory card (when applicable).

Functions menu

Load factory defaults	Reset all SOPAS ET parameters including IP address.
Load application defaults	Reset all SOPAS ET parameters except IP address.
Save image to file	Save the current image to disk.
Image recording	Save multiple images to disk. For details, see "Recording images", page 44.

7.2.2 Image handling controls

Use the image handling controls to manipulate regions and perspective when viewing images. As an alternative to the buttons, you can use a mouse with a scroll wheel, as described below.

Button	Name	Description
	Select	Click and drag to change the size and position of the image. Shortcut command: **Ctrl+Q**.
	Move	Click and drag to move the image. Shortcut command: **Ctrl+W**. **Shift+** press and hold the mouse scroll wheel.
	Rotate	Click and drag to rotate the image. Shortcut command: **Ctrl+E**. Press and hold the mouse scroll wheel.
	Zoom	Click and drag upwards to zoom in and downwards to zoom out. Shortcut command: **Ctrl+R**. Rotate the mouse scroll wheel.

3D navigation control

Use the 3D navigation control in the lower left corner of the image viewer to switch between different viewing angles:

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

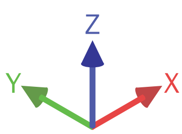
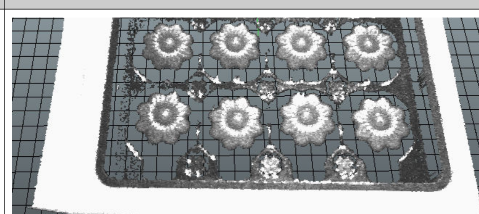
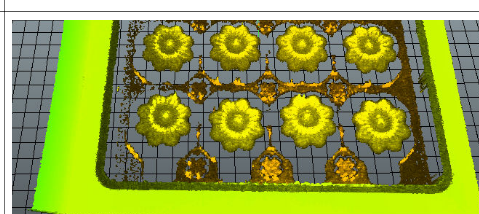
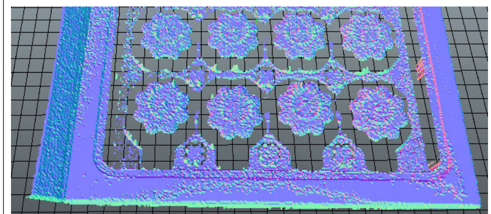
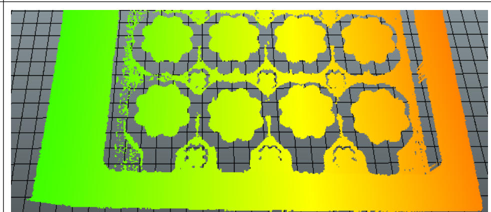
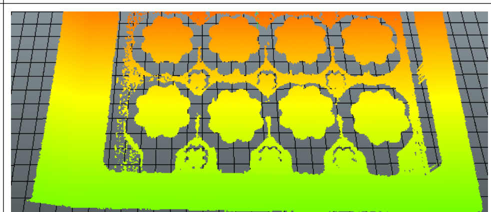
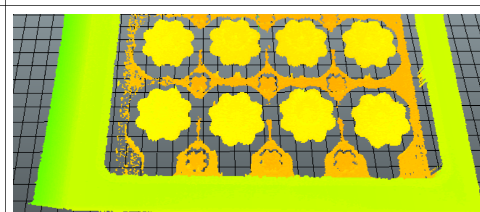


Figure 20: 3D navigation control

7.2.3 Image view modes

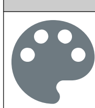
The following view modes are available:

View mode	Description	Example
Reflectance	Color is proportional to the reflectance values along the laser line. Suitable to show the surface details of an object, such as a print.	 *Figure 21: Reflectance view mode*
Reflectance (Hybrid)	Color is proportional to depth (z-coordinate), color brightness is proportional to the laser reflectance. Suitable to show large variations in depth together with surface details.	 *Figure 22: Reflectance (Hybrid) view mode*

View mode	Description	Example
Normals	Color is proportional to the orientation of the surface normal vector. Suitable to show small variations in depth, such as surface structures.	 *Figure 23: Normals view mode*
X	Color is proportional to x-coordinate.	 *Figure 24: X view mode*
Y	Color is proportional to y-coordinate.	 *Figure 25: Y view mode*
Z	Color is proportional to depth (z-coordinate). Suitable to show large variations in depth.	 *Figure 26: Z view mode*

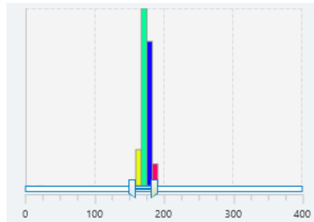
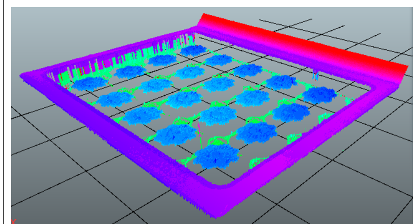
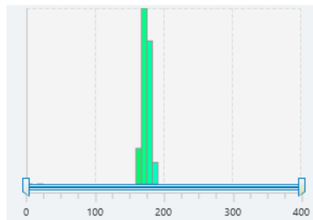
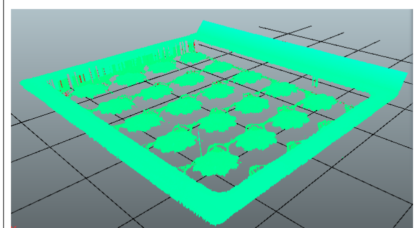
7.2.4 Image view options

The **Color** and **Options** buttons contain image view options.

Button	Name	Description
	Color	View image in color or grayscale
	Options	Contains options for **Color Range**, **Grid**, **Surface** and **Points**, as described below.

- Click **Color Range** to adjust the color interval of the displayed image. This option is only applicable for the **X**, **Y**, **Z** and **Reflectance** view modes.
  The available color spectrum covers the whole range between the minimum and the maximum value. Setting a narrow color range means that the color spectrum is spread over a smaller interval, which makes it easier to see small variations in depth. See table 6.
- Click **Grid** to show or hide the grid at the bottom of the field of view.
- Click **Surface** to view the image as a continuous surface, or **Points** to view the image as a point cloud.

Table 6: Narrow and broad color ranges

	Applied color range	Resulting image
Narrow color range		
Broad color range		

7.2.5 Image view controls

The image view controls are used to switch between the Live 3D image, the Job image, and the Sensor view. Different image view controls are available in different workflow steps.

Image view	Description
Job	Show the job image stored with the **Save job image** button.
Live 3D	Display the latest scanned image.
Sensor	Display the object as perceived by the image sensor. Use this view to adjust image acquisition settings.

7.3 Handling jobs and configurations

A set of image, task and result settings is referred to as a **job** in SOPAS ET, while a set of jobs and interface settings is referred to as a **configuration**.

Use SOPAS ET to manage jobs and to save the resulting configuration. A maximum of 32 jobs can be included in a configuration. Up to 32 tools can be applied to each job, but only one **Shape** tool.

Loading a configuration

When powering on the TriSpector1000, the configuration stored on the microSD memory card is automatically loaded into the device's working memory. If no microSD memory card is inserted, or if there is no configuration stored on the microSD memory card, the configuration is loaded from the device's flash memory to the working memory.

Managing jobs

Click the **Select job** menu in the SOPAS ET functions panel to manage and select jobs.

Notes:

- At the first use of the TriSpector1000, a new job ('Job 0') is created in the device's working memory.
- When an existing configuration is loaded into the device's working memory, the job that was active when the configuration was saved is automatically activated.

Saving a configuration

NOTICE
If a configuration is not saved before disconnecting the power to the device, all settings will be lost.

Select **Save permanent** from the **Parameter** menu in the SOPAS ET functions panel to save a configuration from the device's working memory to the flash memory. If the device has a microSD memory card inserted, the configuration is saved to both the flash memory and the microSD memory card.

Cloning a device using a microSD memory card

Follow the steps below to clone a device, which means copying the configuration and IP address from one device to another. The instruction requires that the first device has a microSD card inserted and is connected to SOPAS ET.

1. In SOPAS ET, select **Save permanent** from the **Parameter** menu to save the current configuration to the microSD memory card.
2. Move the microSD memory card from the first to the second device.
3. Power on the second device.
✓ The configuration on the microSD memory card is loaded into the working memory of the second device.
4. Select **Save permanent** from the **Parameter** menu in the SOPAS ET functions panel to save the configuration to the device's flash memory.

7.4 Workflow steps

To configure the TriSpector1000, click the workflow steps in the user interface (see figure 27). It is possible to change workflow step at any time during the configuration.

Figure 27: Workflow steps

Image

The **Image** workflow step is used to set up image acquisition for good image quality. Two different views are available: **Live 3D** and **Sensor**. See "Image workflow step", page 41 for more information.

Task

The **Task** workflow step contains tools for image analysis. Two different views are available: **Live 3D** and **Job**. The **Job** image is the reference image where the tools are applied and the tool configuration is done. See "Task workflow step", page 44 for more information.

Results

The **Results** workflow step contains settings for result processing and output handling. Note that a bold red underscore in the input fields indicates syntax error. See "Results workflow step", page 56 for more information.

Interfaces

The **Interfaces** workflow step contains settings for connections to external interfaces. See"Interfaces workflow step", page 66 for more information.

7.5 Image workflow step

7.5.1 Scanning an object

The TriSpector1000 builds the image by acquiring a number of laser line profiles of a moving object. Use an encoder if motion is not constant.

1. In the **Motion** section, select **Profile trigger:Free running** or **Incremental encoder** and specify the parameters accordingly. If applicable, click **Calculate** for assistance with encoder calculation. See "Encoder modes", page 41 for descriptions of the encoder modes.
2. Move the object under the TriSpector1000 laser line to make a scan.

7.5.1.1 Encoder modes

The TriSpector1000 has a two directional (up/down) encoder pulse counter. The forward (up) scanning direction is defined as clockwise encoder shaft movement, as seen from the tip of the shaft. See the **Current speed** parameter in the **Image** workflow step for traversing speed. There are five encoder pulse counter-modes: direction up/down, position up/down and motion.

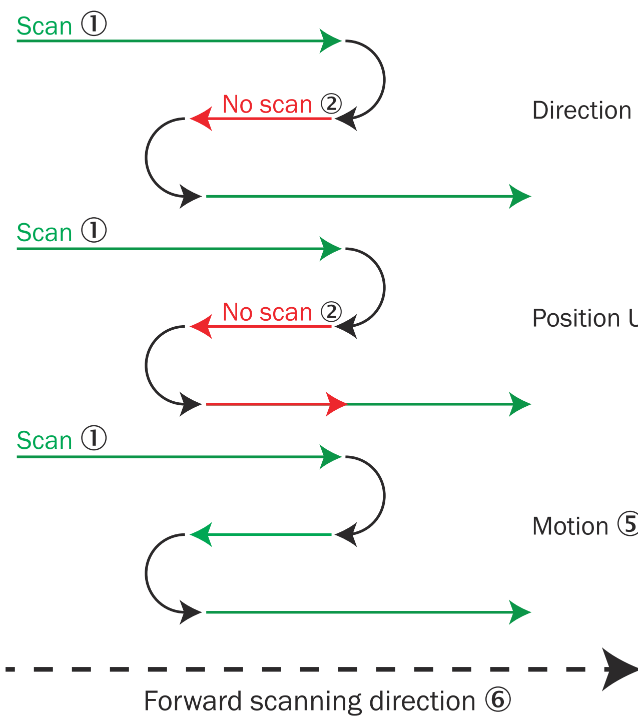


Figure 28: Encoder pulse counter-modes

①	Scan
②	No scan
③	Direction up
④	Position up
⑤	Motion
⑥	Forward scanning direction

7.5.2 Setting the field of view

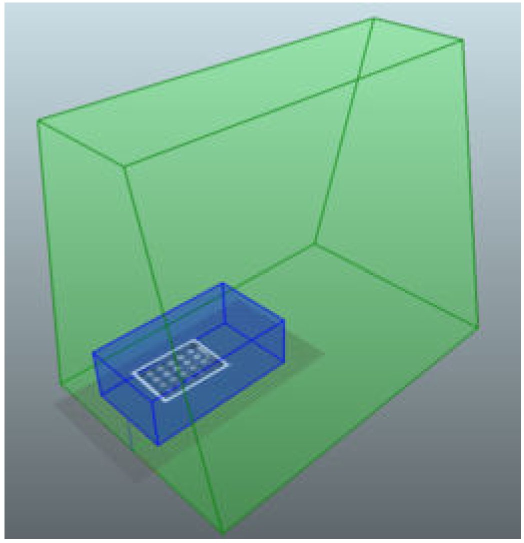


Figure 29: Field of view

The field of view consists of two regions displayed in the image area:

- Guaranteed field of view (green region): The field of view where image acquisition is possible. The guaranteed field of view for the TriSpector1000 consists of 2500 profiles, and the region length is determined by the **Profile distance** parameter.
- Selected field of view (blue region): The region in which the camera acquires range data for image analysis. The **Width** and **Height** of this region are set to maximum by default. The default **Length** depends on the device model (120-480 mm).

Adjust the field of view to optimize performance:

- Edit the **Profile distance** parameter in the **Motion** section in the parameter pane to change the length of the guaranteed field of view. A smaller profile distance allows a higher scan density but reduces the region length.
- Use the value boxes in the **Field of view** section in the parameter pane to resize the selected field of view. A smaller region allows faster processing and higher profile frequency rate.
- Make the pixels in the selected field of view more square by setting the **X-resolution** parameter to a value similar to the **Profile distance** parameter.

NOTE
If the X-resolution and the profile distance differ more than a factor 3, some tools may not work properly.

7.5.3 Adjusting the image settings

To adjust the image settings to get a good 3D image:

1. Click the **Sensor** button located above the image area to see the laser profile which can be used as reference when adjusting the exposure time and gain. See "Laser line exposure time", page 43 for examples.
2. Click the **Live 3D** button and perform a scan.
3. In the **Acquisition** section, adjust the **Exposure time** and **Gain** until the 3D live view looks good.
   Try to keep the gain low, and instead increase the exposure time. This will reduce noise in the image.
4. Adjust the **Laser threshold** parameter to determine which image sensor intensity values to include when locating laser peaks. A high threshold value results in less

noise in the range image, while a low threshold value makes it possible to detect weak laser peaks.

5. Select laser line acquisition criteria in the **Laser peak** list:
   - **Strongest** locates the point with the highest intensity.
   - **Top Most** and **Bottom Most** locate the highest or lowest point with an intensity higher than the value set by the **Laser threshold** parameter. See figure 30.

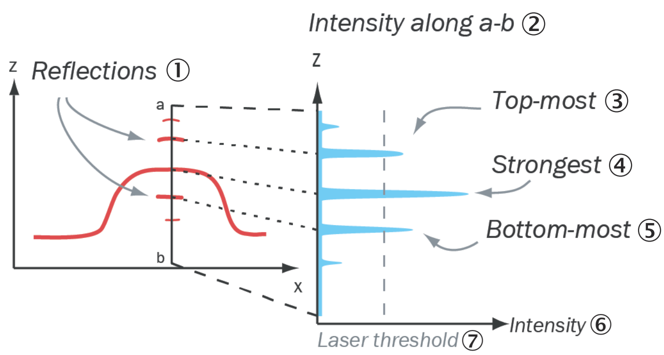


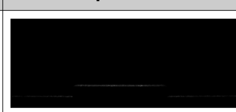
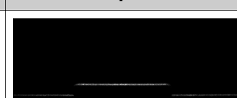
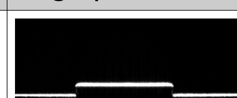
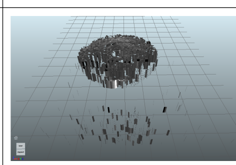
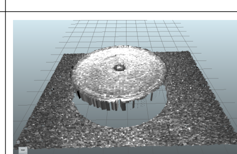
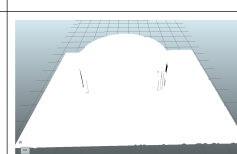
Figure 30: Laser line acquisition criteria.

① Reflections
② Intensity along a-b
③ Top-most
④ Strongest
⑤ Bottom-most
⑥ Intensity
⑦ Laser threshold

6. Repeat the procedure if necessary.

7.5.3.1 Laser line exposure time

The optimal exposure time yields a gray line in the 2D sensor image representation. A slightly brighter line is preferable to a slightly darker line. A solid white laser line indicates a too long exposure time and will result in an overexposed image. See the images below for examples of 2D sensor images and Live 3D images depicting short exposure time, normal exposure time, and long exposure time (from left to right).

View	Short exposure time	Normal exposure time	Long exposure time
Sensor view			
Live 3D view			

7.5.4 Configuring the trigger settings

The **Image Trigger** section is used to determine when image acquisition starts.

- Set the **Image trigger** parameter to **None (Continuous)** to acquire images continuously.
- Select **Object trigger** to define a plane on a certain height. The image acquisition starts when an object is above the specified plane.
- Select **Command Channel** to trigger the image acquisition from the command channel.

- Set the parameter to **Trigger on I/O 3** to trigger the TriSpector1000 by digital I/O via a photoelectric switch or a PLC. The image acquisition can be delayed by time (ms) or distance (mm) from the signal input.
  The TriSpector1000 triggers on a rising edge. The trigger pulse must be at least 50 µs. The TriSpector1000 ignores succeeding pulses during image acquisition.
- Click the **Force trigger** button above the image area to start an image acquisition. If an acquisition is already ongoing, it will be interrupted and re-started.

7.5.5 Recording images

Click **Image recording** (●) to save the acquired 3D images to disk. The saved images can be used in the emulator to configure tools offline. See "Using the SOPAS ET emulator", page 70 for information on how to use the emulator.

The saved images are Portable Network Graphics (.png) files which contain both range data and reflectance data. Each file name includes the name of the active job, the image number and the overall image decision.

To save images to disk:

1. Click the **Image recording** button.
✓ The **Image recording settings** window opens, and the **Image recording** button turns black.
2. Select a destination folder.
3. Select the **Max number of images** checkbox to specify the maximum number of images to save.
4. Select the **Reset image numbering** checkbox to overwrite existing files in the destination folder.
5. Click the **Start** button.
✓ The image recording starts. An image is saved each time a scan is made.
6. Click the **Image recording** button to stop the image recording. If the **Max number of images** checkbox was selected, the recording will stop automatically when the specified number of images have been saved.

7.6 Task workflow step

7.6.1 Using the tools

1. After the image configuration has been done in the **Image** workflow step, select the **Task** workflow step.
✓ The most recently acquired image is shown in the **Live 3D** view.
2. Click the **Save Job Image** button.
✓ The **Tools** buttons appear and the image view changes to **Job**.
3. Click on a tool button to apply the tool to the image. You can apply multiple tools to the image, but only one **Shape** tool.
4. Click the eye symbol (👁) in the parameter pane to toggle the visual representation on or off. If applicable, click one more time to also show the tool's region of interest.
   Selecting a tool in the parameter pane will always show the visual representation for the tool.

The **Job** image view shows the saved job image. Tools can only be applied in this view. The **Live 3D** view shows the latest scanned image.

For an example of how to use tools for image analysis, see "Application example: Counting chocolates", page 54.

7.6.2 Setting the region of interest

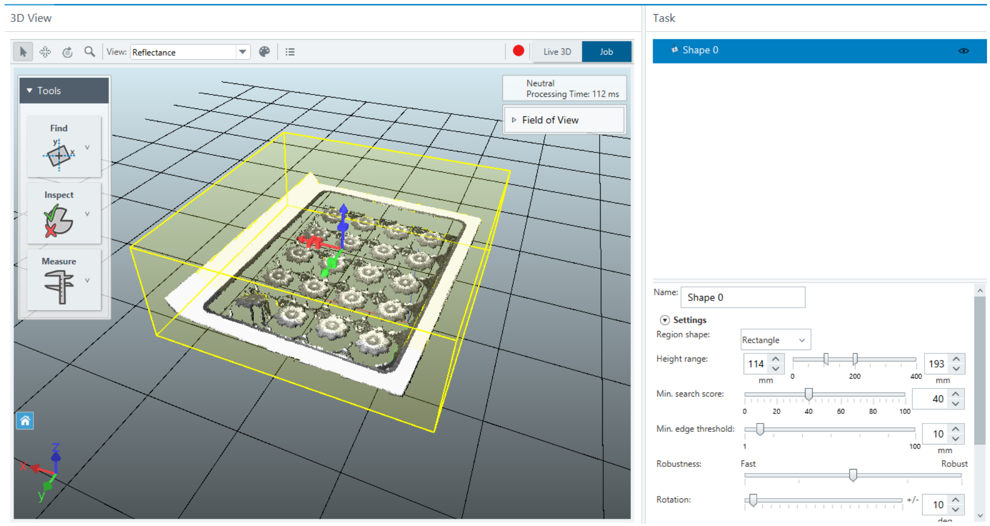


Figure 31: The region of interest (yellow)

The region of interest is the region that a tool is applied to. The region of interest is set for each tool and is displayed as a yellow box in the image area.

To set the region of interest for a tool:

1. Select a tool in the parameter pane.
2. Click the **Select** button in the image handling controls panel.
3. In the 3D navigation control in the lower left corner of the image area, click the green arrowhead for a front view.
4. Use the **Height range** slider in the **Settings** section in the parameter pane to adjust the height of the 3D region, so that it covers only the region of interest. This is important to only include relevant data, and to reduce the processing time.
5. In the 3D navigation control, click the blue arrowhead for a top view.
6. Select the most applicable **Region shape** in the parameter pane.
7. Use the region of interest handles (figure 32) or the region of interest manipulator (figure 33) in the image area to adjust the sides and position of the 3D region, so that it covers only the region of interest.
8. If needed, select **Masks** in the parameter pane to apply a mask that excludes non-relevant parts from the region.

Region of interest handles

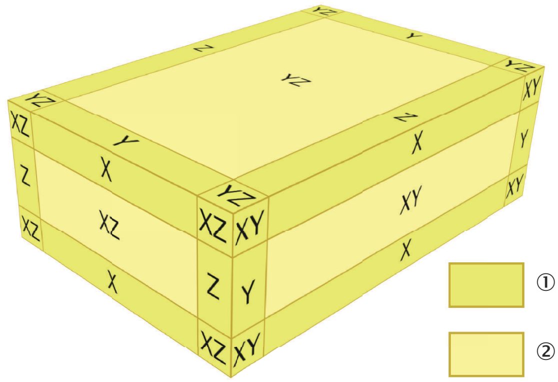


Figure 32: Region of interest handles.

①	Handles for scaling the region of interest
②	Handles for moving the region of interest

Hover over the region of interest to highlight its handles. Click and hold a handle to move or scale the region of interest, according to figure 32.

Region of interest manipulator

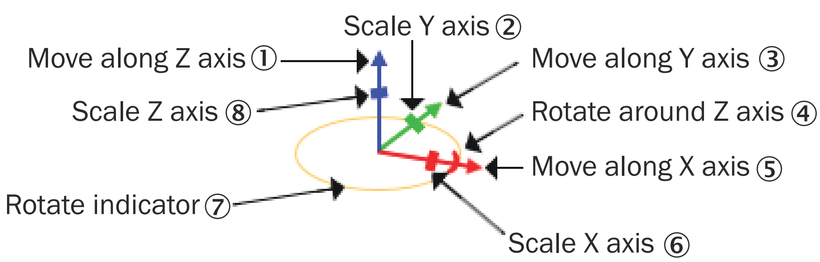


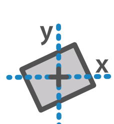
Figure 33: Region of interest manipulator.

①	Move along Z axis
②	Scale Y axis
③	Move along Y axis
④	Rotate around Z axis
⑤	Move along X axis
⑥	Scale X axis
⑦	Rotate indicator
⑧	Scale Z axis

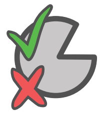
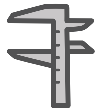
Click the arrows in the image area to move, scale or rotate the region of interest, according to figure 33.

7.6.3 Tool groups

The tools are divided into three groups:



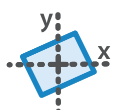
Find tools — Locate objects of different shapes, or features such as edges and planes.

	**Inspect** tools	Inspect properties on located objects or fixed regions in the image.
	**Measure** tools	Measure distance and angle between located objects or features.

Each tool group includes the following tools:

Find tools	**Shape** tool, **Blob** tool, **Edge** tool, **Plane** tool, **Fix Plane** tool, **Intersection** tool, **Point** tool, **Peak** tool
Inspect tools	**Area** tool
Measure tools	**Angle** tool, **Distance** tool

7.6.4 Shape tool



Shape The **Shape** tool locates a reference shape in the image and repositions other tools accordingly. Use the **Shape** tool when the position or rotation of the object varies.

NOTE
If the X-resolution and the profile distance of the image differ more than a factor 3, the **Shape** tool will not work properly. To adjust the values, see "Setting the field of view", page 42.

Shape tool settings

Setting	Description
Region shape	Set the shape of the region of interest to rectangular or elliptical.
Height range	Set upper and lower bounds of the region of interest on the Z-axis.
Min. search score	Lower search scores than the set value returns in a 'Not Found' result.
Min. edge threshold	Only include shapes of greater height than the set threshold.
Robustness	Set trade-off between speed and robustness. Low robustness will result in fewer data points considered to find shape.
Rotation	Maximum allowed shape rotation to search for. The greatest possible value is ±180 degrees.
Masks	Mask a part of the image to exclude it from the search.

Shape tool result output

Result in GUI (Result output parameter)	Description
Decision (Decision)	**OK/Not OK/Invalid/Neutral/Not found** For details, see see "Tool results", page 57.
Rotation (Rotation)	Rotation in degrees.
Score (Score)	Score value (0-100).
(T_x, T_y, T_z)	Translation (x, y, z) in mm.

Linking other tools to the Shape tool

If other tools are added after the **Shape** tool, they will be linked to the **Shape** tool and their regions will be repositioned according to the found shape. Drag and drop tools onto the **Shape** tool in the parameter pane to link them. Drag and drop tools outside the **Shape** tool to unlink them.

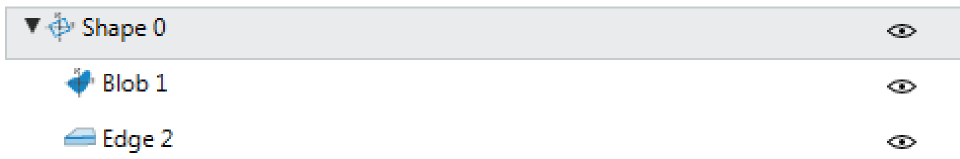
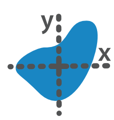


*Figure 34: Tools linked to **Shape** tool*

7.6.5 Blob tool



Blob

The **Blob** tool locates clusters of points within a defined height interval and cluster size. This makes it possible to measure volume, area, angle, and bounding box.

Blob tool settings

Setting	Description
Region shape	Set the shape of the region of interest to rectangular or elliptical.
Height range	Set upper and lower bounds of the region of interest on the X-axis.
Blob area	Only include blobs that have an area within a specified range.
Ignore data	Exclude all data above or below an assigned plane (defined by a **Plane** or **Fix Plane** tool) from the analysis.
Blob volume	Enable blob volume measurement, either from the bottom of the region of interest or from an assigned plane.
Sort by	Sort the list of found blobs by X-position, Y-position or area.
Invert blobs	Invert blobs to locate holes and gaps instead.
Exclude border blobs	Remove all blobs that touch the region of interest or image border.
Fill holes	Remove single pixel holes in blob.
Calculate angles	Calculate angles of blobs. The blob angle is the angle between the x-axis and an axis around which it would be easiest to rotate the blob.
Calculate bounds	Calculate bounding box position, width, and length.
Show blob markers	View a marker for each found blob in the image area.
Erode	Trim outer edge of blobs to separate discrete blobs when height map data overlaps.
Accuracy	Set trade-off between speed and accuracy, setting a high speed will downscale the image used for processing.
Masks	Mask a part of the image to exclude it from the search.

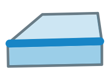
Blob tool result output

Result in GUI (Result output parameter)	Output
Decision (Decision)	**OK**/**Not OK**/**Invalid**/**Neutral** For details, see see "Tool results", page 57.
Blobs (NumBlobs)	Number of blobs detected. Select the **Blobs** checkbox to use the number of blobs as a pass/fail-condition.

Result in GUI (Result output parameter)	Output
Blob volume (OverallVolumeDecision)	**OK/Not OK.** Select the **Blob volume** checkbox to use the blob volume as a pass/fail-condition. The checkbox is only visible if blob volume measurements are enabled in the **Settings** section.
First blob/blobs[]: ■ index ■ cogX ■ cogY ■ cogZ ■ width* ■ length* ■ cx* ■ cy* ■ boundsAngle* ■ area ■ blobAngle* ■ volume* ■ volumeDecision	- ■ Index of blob in list. ■ X-center of blob. ■ Y-center of blob. ■ Z-height of X/Y-center. ■ Width of bounding box. ■ Length of bounding box. ■ X-center of bounding box. ■ Y-center of bounding box. ■ Bounding box angle (±90° from x-axis). ■ Blob area in mm^2. ■ Blob angle in degrees (±90° from x-axis). ■ Volume in cm^3. ■ **OK/Not OK**. Blob volume threshold per blob.

* If activated in the tool

7.6.6 Edge tool



Edge The **Edge** tool finds edges in the range data.

Edge tool settings

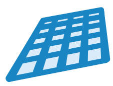
Setting	Description
Height range	Set upper and lower bounds of the region of interest on the X-axis.
Min. edge threshold	Only include edges of greater height than the minimum threshold. Raise the value to remove noise.
Edge side	Position line approximation on top or below of edge.
Polarity	Find rising edges, falling edges, or both.
Selection	Select which edge the **Edge** tool chooses: first edge, last edge or strongest edge found.
Edge sharpness	Adjust to find sharper or softer edges.
Probe spacing	Set the distance (in pixels) between edge locator probes, orthogonal to the search direction.
Outlier mode	Manually or automatically trim data points for line approximation. Manual mode enables the outlier distance setting.
Outlier distance	Only include data points within this distance (mm) for line approximation.
Masks	Mask a part of the image to exclude it from the search.

Edge tool result output

Result in GUI (Result output parameter)	Output
Decision (Decision)	**OK/Not OK/Invalid/Neutral** For details, see see "Tool results", page 57.
(MidPointX, MidPointY, MidPointZ)	Edge center point (x, y, z).
(DirectionX, DirectionY, DirectionZ)	Edge direction (x, y, z).

Result in GUI (Result output parameter)	Output
Score (Score)	Score value (0-100). Select the **Score** checkbox to use the score value as a pass/fail-condition.
Mean deviation (MeanDeviation)	Mean deviation from line approximation (mm). Select the **Mean deviation** checkbox to use the mean deviation as a pass/fail-condition.

7.6.7 Plane tool



Plane

The **Plane** tool uses pixels inside its region of interest to find a flat surface. Use the **Plane** tool to define plane features for use with other tools, or to use the tool output independently.

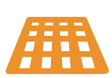
Plane tool settings

Setting	Description
Region shape	Set the shape of the region of interest to rectangular or elliptical.
Height range	Set upper and lower bounds of the region of interest on the Z-axis.
Robustness	Set trade-off between speed and robustness. Low robustness will result in fewer data points considered to find a plane.
Percentile	Set the span of height data points to include when calculating plane.
Show used points	Indicates the data points used to calculate plane.
Masks	Mask a part of the image to exclude it from the search.

Plane tool result output

Result in GUI (Result output parameter)	Output
Decision (Decision)	**OK/Not OK/Invalid/Neutral** For details, see see "Tool results", page 57.
Score (Score)	Score value (0-100). Select the **Score** checkbox to use the score value as a pass/fail-condition.
Tilt (Tilt)	Angle between the normal vector to the plane and the z-axis (in degrees). Select the **Tilt** checkbox to use the tilt value as a pass/fail-condition.
(C_x, C_y, C_z, N_x, N_y, N_z)	Equation of the plane through center point + normal.
(P_x, P_y, P_z, P_d)	Equation of the plane through P_x*x + P_y*y + P_z*z + P_d = 0.

7.6.8 Fix Plane tool



Fix Plane

The **Fix Plane** tool is used to manually set a reference plane in the field of view. The reference plane can be used as feature for other tools.

Fix Plane tool settings

Setting	Description
Position X	Center point of plane on X axis.
Position Y	Center point of plane on Y axis.
Position Z	Center point of plane on Z axis.

Setting	Description
XZ angle	Angle around the X axis.
YZ angle	Angle around the Y axis.

Fix Plane tool result output

Result in GUI (Result output parameter)	Output
Decision (Decision)	**Invalid/Neutral** For details, see see "Tool results", page 57.
(C_x, C_y, C_z, N_x, N_y, N_z)	Equation of the plane through center point + normal.
(P_x, P_y, P_z, P_d)	Equation of the plane through P_x*x + P_y*y + P_z*z + P_d = 0.

7.6.9 Intersection tool

Intersection

The **Intersection** tool finds the intersection point between two edges.

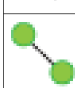
Intersection tool settings

Setting	Description
Feature A	First referenced **Edge** tool for finding the inter-section point
Feature B	Second referenced **Edge** tool for finding the intersection point
Projection	Projection of the intersection point: • **XY plane**: The intersection point is projected onto the XY-plane. The z-value is set based on the z-value for one of the edges, or the mean z value for both edges. • **Plane**: The intersection point is projected onto a selected **Plane** or **Fix Plane** tool.

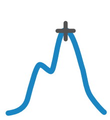
Intersection tool result output

Result in GUI (Result output parameter)	Output
Decision (Decision)	**Invalid/Neutral** For details, see see "Tool results", page 57.
(x, y, z)	Intersection point (x, y, z)

GUI operations

GUI button	Function
	Switch **Feature A** and **Feature B**.
	Enable or disable the selection of features in the image area.

7.6.10 Peak tool



Peak

The **Peak** tool finds the point with the minimum or maximum height value in its region of interest.

Peak tool settings

Setting	Description
Region shape	Set the shape of the region of interest to rectangular or elliptical.
Height range	Set upper and lower bounds of the region of interest on the X-axis.
Type	Find point with the minimum or maximum height value.
Use Median filter	Use a filter to remove noise spikes in the heightmap.
Kernel size	Radius of the median filter.
Masks	Mask a part of the image to exclude it from the search.

Peak tool result output

Result in GUI (Result output parameter)	Output
Decision (Decision)	**OK/Not OK/Invalid/Neutral** For details, see see "Tool results", page 57.
(x, y, z)	Peak position (x, y, z). Select the **Height** checkbox to use the height (z-position) as a pass/fail-condition.

7.6.11 Point tool

Point

The **Point** tool is used to manually set a reference point in the field of view. The tool can be used to define point features for use with other tools, or to use the tool output independently.

Point tool settings

Setting	Description
Position X	Point position on the X axis.
Position Y	Point position on the Y axis.
Position Z	Point position on the Z axis.
Snap to data	Place the point on the data height in the XY plane for the saved Job image (disables the Z-slider). This only applies when training the Point tool in **Job** view.

Point tool result output

Result in GUI (Result output parameter)	Output
Decision (Decision)	**Invalid/Neutral** For details, see see "Tool results", page 57.
(x, y, z)	User-defined position (x, y, z).

7.6.12 Area tool

Area

The **Area** tool calculates surface coverage by counting points within a defined 3D region, or within a specified intensity interval inside the region.

Area tool settings

Setting	Description
Region shape	Set the shape of the region of interest to rectangular or elliptical.
Height range	Set upper and lower bounds of the region of interest on the X-axis.

Setting	Description
Use intensity	Use grayscale intensity data in addition to height.
Intensity range	Limit the area tool to a specific range of image intensities.
Masks	Mask a part of the image to exclude it from the search.

Area tool result output

Result in GUI (Result output parameter)	Output
Decision (Decision)	**OK**/**Not OK**/**Invalid**/**Neutral** For details, see see "Tool results", page 57.
Coverage (Coverage)	Percentage of region coverage within the region of interest. Select the **Coverage** checkbox to use the coverage percentage as a pass/fail-condition.
(Area)	Area in mm^2.

7.6.13 Distance tool

Distance The **Distance** tool measures the distance between two tool key points.

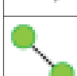
Distance tool settings

Setting	Description
Feature A	First referenced feature for measuring distance.
Feature B	Second referenced feature for measuring distance.
Measure type	■ **Right Angle** projects a point feature onto a line feature, and measures the distance between the projected point and the line feature. ■ **Point To Point** measures the distance between point features.
Measure distance in	Select dimensions to measure distance in.

Distance tool result output

Result in GUI (Result output parameter)	Output
Decision (Decision)	**OK**/**Not OK**/**Invalid**/**Neutral** For details, see see "Tool results", page 57.
Distance (Distance)	Distance in mm. Select the **Distance** checkbox to use the distance as a pass/fail condition.

GUI operations

GUI button	Function
	Switch **Feature A** and **Feature B**.
	Enable or disable selection of features in the image area.

7.6.14 Angle tool

Angle The **Angle** tool measures the angle between two planes or edges.

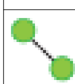
Angle tool settings

Setting	Description
Feature A	First referenced **Plane** or **Edge** tool for measuring angle.
Feature B	Second referenced **Plane** or **Edge** tool for measuring angle.
Projection	Projection of the angle to be measured. Only applicable if both features are edges. • **XY plane**: The angle is projected onto the XY-plane. • **Plane**: The angle is projected onto a selected **Plane** or **Fix Plane** tool.

Angle tool result output

Result in GUI (Result output parameter)	Output
Decision (Decision)	**OK**/**Not OK**/**Invalid**/**Neutral** For details, see see "Tool results", page 57.
Angle (Angle)	Angle in degrees. Select the **Angle** checkbox to use the angle as a pass/fail-condition.

GUI operations

GUI button	Function
	Switch **Feature A** and **Feature B**.
	Enable or disable selection of features in the image area.

7.6.15 Application example: Counting chocolates

The following steps describe how to use tools to count the number of chocolates in a box. Images from the TriSpector1030 emulator are used for the analysis.

1. Start the TriSpector1030 emulator in SOPAS ET. For instructions on how to start the emulator, see "Starting the emulator", page 71.

✓ The SOPAS ET device window opens.

2. Go to the **Image** workflow step and check that chocolate box images are displayed in the image area according to figure 35.

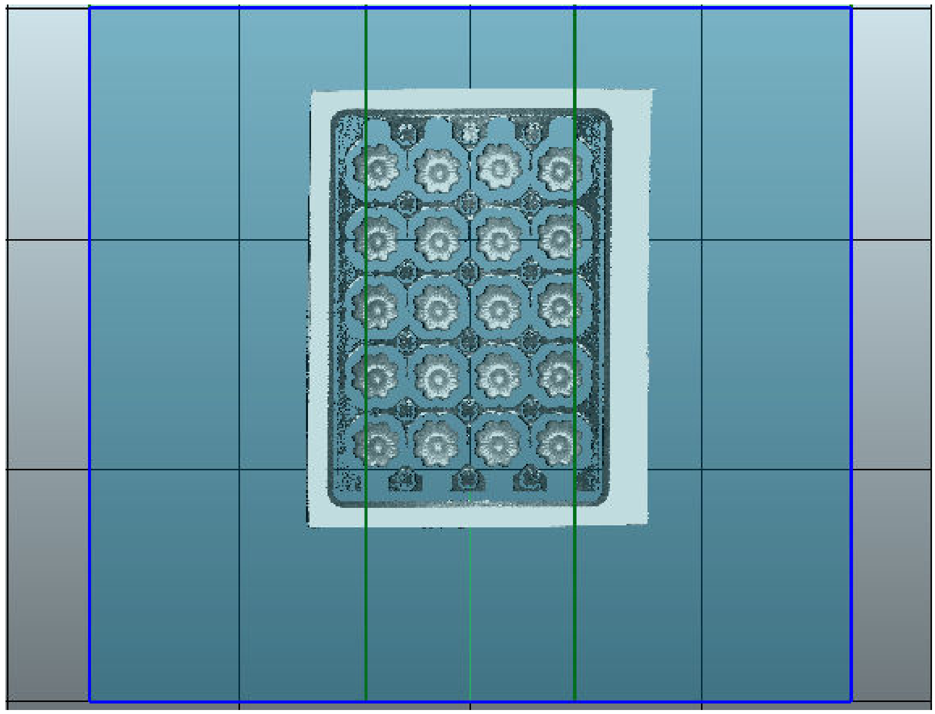


Figure 35: TriSpector1030 emulator image

3. Go to the **Task** workflow step.
4. Click **Save job image** to switch to the job image view where the tool groups are shown.
5. In the **Find** tool group, click **Shape** to apply a **Shape** tool to the job image.
6. Move and resize the region of interest for the **Shape** tool to cover the chocolate box. Leave some extra space between the object and the region borders. See figure 36.

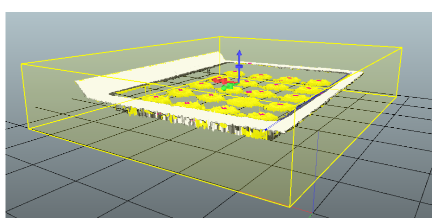


*Figure 36: Region of interest for **Shape** tool*

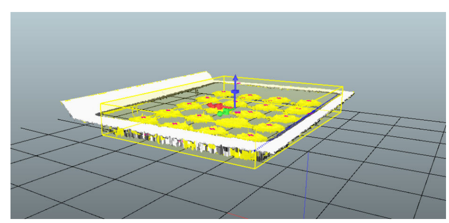


*Figure 37: Region of interest for **Blob** tool*

7. In the **Find** tool group, click **Blob** to apply a **Blob** tool to the job image.
8. Move and resize the region of interest for the **Blob** tool to cover only the chocolates. See figure 37.
9. Adjust the **Blob area** slider in the **Settings** section to only include the chocolates, by setting the minimum value to 300 mm^2 and the maximum value to 400 mm^2. See figure 38.

Figure 38: Blob area slider

10. Select the **Blobs** checkbox in the **Results** section to use the number of blobs as a Pass/Fail-condition. Adjust the slider so that both the minimum and the maximum value are set to 20. See figure 39.

Figure 39: Blobs checkbox

11. Click **Live 3D** and then **Run** (▶) to loop through the images.

✓ The tools are now applied to all images.

For images where the **Blob** tool finds 20 chocolates, an '**OK**' result is displayed in the image area and the region of interest turns green (see figure 40). If the number of found chocolates is not equal to 20, a '**Not OK**' result is displayed in the image area and the region of interest turns red (see figure 41).

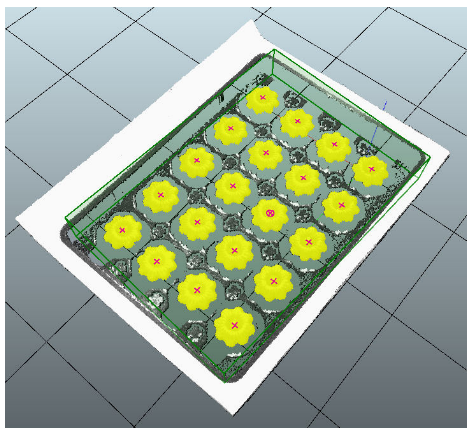


*Figure 40: Result '**OK**'*

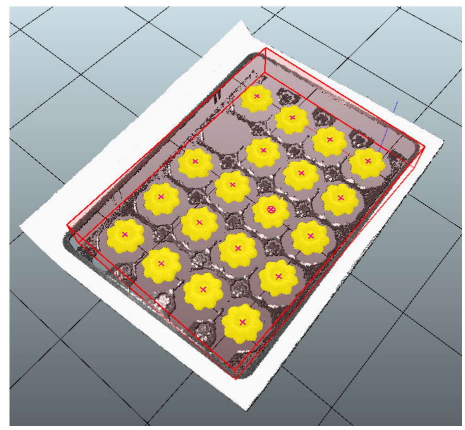


*Figure 41: Result '**Not OK**'*

7.7 Results workflow step

7.7.1 Result handling

Result handling consists of methods to define and format device output. Use cases range from activation of a reject mechanism to output of formatted string results.

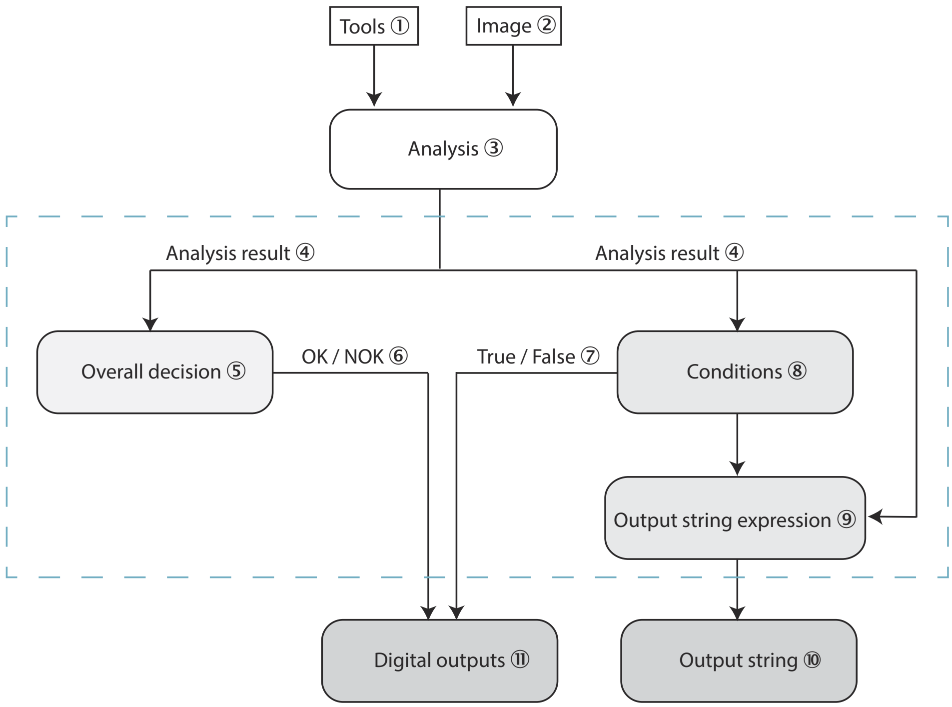


Figure 42: Overview of the result handling

① Tools
② Image
③ Analysis
④ Analysis result
⑤ Overall decision
⑥ OK / NOK
⑦ True / False
⑧ Conditions
⑨ Output string expression
⑩ Output string
⑪ Digital outputs

Binary results are based on tool tolerances set by the user. Conditions and expressions can be used for custom result handling without setting tolerances.

7.7.2 Tool result output

Tools return results as values (for example area: 42 mm^2, angle: 12 degrees) or binary results (**OK**/**Not OK**). The binary results are controlled by tolerances set by the user in the tool parameter pane in the **Task** workflow step, e.g. the number of blobs found by a **Blob** tool. If no tolerances are set, the results are returned as neutral.

7.7.2.1 Tool results

The table below lists the available result types and their meanings for the different tools.

NOTE

Notes:

- An empty cell means that the result type is not available for the tool.
- If a **Shape** tool result is **Invalid** or **Not found**, all its linked tools are assigned an **Invalid** result (not true for the **Angle**, **Distance**, and **Intersection** tools).

Tool	Invalid	Not found	Neutral	OK	Not OK
Shape	Data processing error (teach of reference object failed)	Match score below **Min. Search score.**	No pass/fail-conditions set	One or more pass/fail-conditions set, result passed	One or more pass/fail-conditions set, result failed
Blob	Data processing error		No pass/fail-conditions set	One or more pass/fail-conditions set, result passed	One or more pass/fail-conditions set, result failed
Edge	Data processing error (no edges found)		No pass/fail-conditions set	One or more pass/fail-conditions set, result passed	One or more pass/fail-conditions set, result failed
Plane	Data processing error (no plane found)		No pass/fail-conditions set	One or more pass/fail-conditions set, result passed	One or more pass/fail-conditions set, result failed
Fix Plane	Data processing error		OK (no errors)		
Intersection	Data processing error (one or more selected features are invalid, or the selected edges are parallel)		OK (no errors)		
Peak	Data processing error (no data in the region of interest)		No pass/fail-conditions set	One or more pass/fail-conditions set, result passed	One or more pass/fail-conditions set, result failed
Point	Data processing error		OK (no errors)		
Area	Data processing error		No pass/fail-conditions set	One or more pass/fail-conditions set, result passed	One or more pass/fail-conditions set, result failed
Distance	Data processing error (one or more selected features are invalid)		No pass/fail-conditions set	One or more pass/fail-conditions set, result passed	One or more pass/fail-conditions set, result failed

Tool	Invalid	Not found	Neutral	OK	Not OK
Angle	Data processing error (one or more selected features are invalid, the projection plane is invalid, or the projection plane is not set)		No pass/fail-conditions set	One or more pass/fail-conditions set, result passed	One or more pass/fail-conditions set, result failed

7.7.2.2 Tool result symbols

The overall image result is displayed in the image area. Individual tool results are displayed in the parameter pane in the **Task** workflow step.

Image area		Tool symbol (parameter pane)	Tool result section (parameter pane)	Description
	Not OK			Result not OK.
	OK			Result OK.
	Invalid			Result invalid.
	Not found			Result not found.
	Neutral			Result neutral (no tolerances set).

7.7.2.3 Decision model

The SOPAS ET decision model applies tool tolerances to define binary results. If multiple tools are used, the tool results are combined for the overall image decision (AND logic).

The priority for a decision is:

1 **Not Found** (one or more tool results are '**Not Found**')
2 **Invalid** (one or more tool results are '**Invalid**')
3 **Not OK** (one or more tool results are '**Not OK**')
4 **OK** (all tool results are '**OK**')
5 **Neutral** (no tool tolerances are set)

NOTE
Only '**OK**' and '**Not OK**' can be sent as overall decisions to the digital outputs. Other result types can be handled by conditions, see "Condition examples", page 60.

7.7.3 Conditions

In the **Conditions** section, the results from the **Task** workflow step are used to create true/false expressions.

"Condition examples", page 60 and "Handling arrays by conditions", page 61 contain examples of conditions for mathematical operations, result checks and array handling. Use the results as digital output, for creation of result strings or as input to other conditions.

Creating a condition

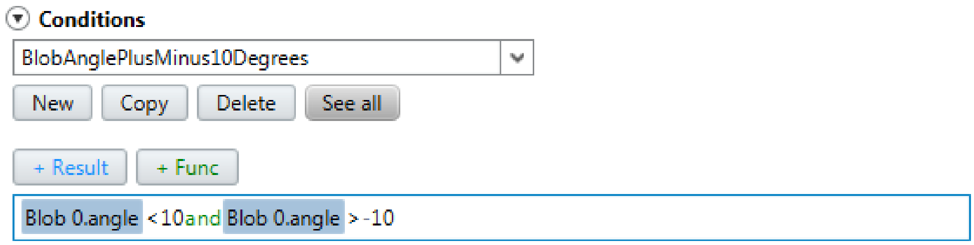


Figure 43: Creating a condition

1. Click **New** and name the condition.
2. Edit the condition in the condition editor, either by typing or by inserting parameters from the **+Result** and **+Func** lists.
   - The **+Result** list contains image analysis results and tool result output.
   - The **+Func** list contains supported functions and operators.

7.7.3.1 Condition examples

Example 1: Rotation

```
Shape 0.Rotation < 10 and Shape 0.Rotation > 5
```

This condition returns true if the **Rotation** output from the **Shape** tool "Shape 0" is between 5 and 10 degrees.

Example 2: Decision

```
Tools.OverallDecision = "Not found" or Blob 1.NumBlobs ~=20
```

This condition returns true if the overall decision (for all tools) is **Not found** or if the number of blobs from the **Blob** tool "Blob 1" is not equal to 20.

Example 3: Calculate angle

```
deg(atan(Plane 1.Nx / Plane 1.Nz)) < 20
```

This condition returns true if the angle from the **Plane** tool "Plane 1" in the XZ plane is less than 20 degrees.

Plane 1.Nx returns the X component of the normal vector for the plane, while **Plane 1.Nz** returns the Z component. The `atan()` function returns the value of the angle from the Z axis, in radians. The `deg()` function converts the value to degrees and returns a positive value in the interval [0, 360].

Example 4: Condition inside condition

```
NewCondition2 or Distance 1.Distance > 50
```

This condition returns true if the condition **NewCondition2** returns true or if the **Distance** output from the **Distance** tool "Distance 1" is less than 50 mm.

NOTE
When using conditions inside conditions, make sure not to create a circular dependency.

7.7.3.2 Handling arrays by conditions

In the examples below, conditions are used to create and handle arrays. Array handling is useful for the **Blob** tool, which returns most results as arrays with one element for each found blob.

Example 1: Maximum value

```
max( Blob 0.area[] ) < 50
```

This condition returns true if the maximum value in the **Blob 0.area**[] array is smaller than 50 mm^2.

Example 2: Single element

```
element( Blob 0.blobAngle[] , 3) > 0
```

This condition returns true if the fourth element (the element with index 3) in the **Blob 0.blobAngle**[] array has a positive value.

The `element()` function accesses a specified element in the array. Note that the first element in the array has index 0.

NOTE
The condition returns an error if the array contains less than 4 blobs, which means that neither a true or false value is returned.

Example 3: Check each element

```
all( Blob 0.cogX[]  < 50 and  Blob 0.cogX[]  > -50)
```

This condition returns true if the X-center of each blob in the **Blob 0.cogX[]** array is in the interval [-50, 50].

Example 4: Create array

```
max(array( Peak 0.z , Peak 1.z , Peak 2.z , Peak 3.z )) < min(array( Peak 0.z , Peak 1.z , Peak 2.z , Peak 3.z )) + 1
```

This condition returns true if the distance between the Z coordinates of four **Peak** tools is greater than 1 mm.

The `array()` function creates an array containing the Z values for four **Peak** tools. The minimum and maximum values in the array are then compared.

7.7.4 Digital outputs

The **Digital outputs** section specifies which results to send to the available digital outputs.

NOTE
Enable the digital outputs in the **I/O Definitions** section in the **Interfaces** workflow step.

Start of signal

The **Type** column defines the starting point for each output signal. There are two available options:

- **On Result**: The signal is sent to the assigned output when the image analysis result is ready.
- **Delayed**: The signal is sent to the assigned output after a delay, which starts when the first image profile is triggered.

The length of the delay is set by the user in the **Delay from image start** field. The delay must be longer than the total time for image acquisition and analysis, otherwise the output is discarded. Because the time for image acquisition and analysis may vary, the delay starts when the first image profile is triggered.

Set the delay either as a distance or as a time interval. Because the distance measurements are based on encoder ticks, an encoder must be connected to get a correct output.

Signal criteria

The **Activate when** column defines the criteria for sending output signals to the assigned outputs. The following options are available:

- **Device ready**: The device is ready for operation. This option is only available for non-delayed signals (where **Type** = **On Result**).
- **Result ready**: The image analysis result is ready.
- **Overall result OK/Not OK**: The overall image result is OK/Not OK.
- Created conditions: True/False.

Signal duration

The **Deactivate when** column defines the duration of the output signal. The following options are available:

- **Hold until next result**: Hold signal until a new result is ready.
- **Duration complete**: The signal duration is set manually by the user.
  For delayed signals (**Type** = **Delayed**), the signal duration can be set as a distance. The distance measurements are based on encoder ticks. If a distance measurement fails, a duration timeout is used as a fallback option.
- **Overall result OK/Not OK**: Hold signal until result is OK/Not OK.

7.7.5 Ethernet output string

Use the **Ethernet output string** section to define a result string, based on e.g. a boolean tool decision or a condition.

- The **+Prog** list contains statements.
- The **+Char** list contains ASCII characters.
- The **+Result** list contains available image analysis results and tool decisions.
- The **+Func** list contains supported functions and operators.

Figure 44: Example of result string

In figure 44, the coordinates of the first blob are returned if the **BlobAnglePlusMinus10Degrees** condition evaluates as true. If the condition evaluates as false, zero coordinates are returned.

The result string can be sent as output via Ethernet/IP, TCP/IP or the serial interface. To enable output via TCP/IP or the serial interface, select the applicable **Output string** checkbox in the **Interfaces** workflow step.

7.7.6 Image logging

Edit the **Image logging** section to set the log conditions to use for logging images to an FTP server or to a microSD memory card.

The lists of log conditions are only available when image logging is active. To activate image logging, go to the **Interfaces** workflow step and select the **FTP** checkbox or the **SD Card** checkbox.

NOTE
The **Image logging** section does not show the status of the image transfer. To check the connection to the FTP server, see "Logging images to an FTP server", page 67.

Log conditions

The following log conditions are available:

1. **None** - Log no images.
2. **All** - Log all images.
3. **Overall result** - Log images for which the overall result is '**OK**' or '**Not OK**'.
4. Conditions - Log images based on a created condition (see "Conditions", page 59).

Naming logged files

Use the checkboxes in the **Image logging** section to select which variables to include in the file names of the logged images.

The **Date** and **Time** variables are based on the set device time in the **Interfaces** workflow step, see "Setting the device time", page 69.

7.7.7 Functions and operators

The TriSpector1000 supports a set of operators and functions for result handling. The categories of operators and functions are mathematical, logical and strings.

7.7.7.1 Logical operators

Table 7: Logical operators

Operator	Syntax	Result type	Description
<	x < y	BOOL	Smaller than
>	x > y	BOOL	Larger than
<=	x <= y	BOOL	Smaller than or equal to
>=	x >= y	BOOL	Greater than or equal to
=	x = y	BOOL	Equal to
~=	x ~= y	BOOL	Not equal to
and	x and y	BOOL	Logical and
or	x or y	BOOL	Logical or
not	not(x)	BOOL	Logical not of a boolean x

7.7.7.2 Logical functions

Table 8: Logical functions

Function	Syntax	Result type	Description
any	any(array)	BOOL	Check if any array element is true.
all	all(array)	BOOL	Check if all array elements are true.
count	count(array)	NUM	Count all true values in a boolean array.
contains	contains(array, x)	BOOL	Check if a value x is present in an array.
element	element(array, index)	Any	Get an element out of an array by index.
array	array(x1, ...)	Any	Create an array of list elements, array index starts at zero.
size	size(array)	NUM	Calculate the size of an array.

7.7.7.3 Mathematical operators

Table 9: Mathematical operators

Operator	Syntax	Result type	Description
+	x + y	NUM	Addition
-	x - y	NUM	Subtraction
-	-x	NUM	Negation
*	x * y	NUM	Multiplication
/	x / y	NUM	Division
^	x^y	NUM	Exponentiation
.	x.y	NUM	Decimal

7.7.7.4 Mathematical functions

Table 10: Mathematical functions

Functions	Syntax	Result type	Description
min	min(x1, ...)	NUM	Find the minimum value in a list of elements or an array.

Functions	Syntax	Result type	Description
max	max(x1, ...)	NUM	Find the maximum value in a list of elements or an array.
sum	sum(x1, ...)	NUM	Sum of all elements in a list or an array.
abs	abs(x)	NUM	Absolute value of x.
round	round(x, dec)	NUM	Round x to an integer or to a fixed number of decimals.
mod	mod(x,y)	NUM	Calculate x modulo y.
sqrt	sqrt(x)	NUM	Calculate the square root of x.
sin	sin(x)	NUM	Calculate the sine of x in radians.
cos	cos(x)	NUM	Calculate the cosine of x in radians.
tan	tan(x)	NUM	Calculate the tangent of x in radians.
asin	asin(x)	NUM	Calculate the arcsine of x in radians.
acos	acos(x)	NUM	Calculate the arccosine of x in radians.
atan	atan(x)	NUM	Calculate the arctangent of x in radians.
deg	deg(rad)	NUM	Convert radians to degrees.
rad	rad(deg)	NUM	Convert degrees to radians.
pow	pow(x,y)	NUM	Calculate a to the power of b (x^y).
exp	exp(x)	NUM	Calculate e to the power of x (e^x).
ln	ln(x)	NUM	Calculate the natural logarithm of x.
log10	log10(x)	NUM	Calculate the logarithm of x (base 10).

7.7.7.5 String operators

Table 11: String operators

Operator	Syntax	Result type	Description
""	"x"	STR	String representation.
..	x .. y	STR	Concatenation of strings x and y.
=	x = y	BOOL	Check if two strings x and y are equal, with wildcard functionality.

7.7.7.6 String functions

Table 12: String functions

Function	Syntax	Result type	Description
matches	matches(str,pattern,mode)	BOOL	Check if a string matches a regular expression. The mode variable can be set to "unicode" or "binary", the default setting is "unicode".
hasSub-String	hasSub-string(str,sub-str)	BOOL	Check if a string contains a substring. Case sensitive.
find	find(str,sub-str)	NUM	Check if a string contains a substring. Returns index of the first character of a found sub-string. Case sensitive.
length	length(str)	NUM	Calculate the length of a string in bytes.
substr	sub-str(str,start,end)	STR	Get a substring from a string.

Function	Syntax	Result type	Description
`token`	`token(str,d)`	Array	Create an array of a string of tokens with delimiter d.
`prefix`	`prefix(str,filler, len)`	STR	Fill the beginning of a string with a filler character up to a specified length.
`postfix`	`postfix(str,filler, len)`	STR	Fill the end of a string with a filler character up to a specified length.
`roundToStr`	`roundToStr(number,dec)`	STR	Convert a number to a string, with a given number of decimals,

7.7.7.7 String wildcards

Table 13: String wildcards

Wildcard	Syntax	Description
`*`	`*S`	Any string precedes a certain string. Equal to regular expression ".*"
`?`	`?S`	Any character precedes a certain string. Equal to regular expression ".?"

7.8 Interfaces workflow step

7.8.1 Configuring the digital in- and outputs

The digital I/O interfaces are available on the Power I/O connector, see "Pin assignment", page 32. I/O 4, 5, 6 and 7 are configurable and can be used for either input or output signals.

NOTE
When an I/O is configured as output, define its output signal in the **Digital outputs** section in the **Results** workflow step.

Configure I/O as input

1. Set the **Type** parameter in the **I/O Definitions** table to **In**.
2. Set the input signal level (**Active** parameter) to **High** (24 V) or **Low** (0 V).
3. Set the minimum input signal length by the **Debounce** parameter (1 ms is default).

Configure I/O as output

1. Set the **Type** parameter in the **I/O Definitions** table to **Out**.
2. Set the output signal level (**Active** parameter) to **High** (24 V) or **Low** (0 V).

7.8.2 Using the command channel

A command channel is a camera interface which is used for external modification of image and tool parameter settings. The available command channel interfaces for the TriSpector1000 are described below.

See "Available commands for command channel", page 82 for a full list of available commands.

TCP/IP

To use TCP/IP as command channel interface:

1. Select the **Command channel** checkbox in the **Ethernet** section in the parameter pane.
2. In the **Server/client** drop-down menu, select to act as server or client (depending on the settings of the connected device).
3. Set port number in the **TCP Port** text field. 2115 is the default port.

Serial interface

To use the serial interface as command channel interface:

1. Select the **Command channel** checkbox in the **Serial** section in the parameter pane.

EtherNet/IP

To use EtherNet/IP as command channel interface, see "Setting up a TriSpector1000 to communicate via EtherNet/IP", page 92.

7.8.3 Sending an output string

The available interfaces for sending an output string from the TriSpector1000 to a connected device are described below.

Define the output string in the **Ethernet output string** section in the **Results** workflow step, see "Ethernet output string", page 62 for details.

TCP/IP

To send the output string via TCP/IP:

1. Select the **Output string** checkbox in the **Ethernet** section in the parameter pane.
2. In the **Server/client** drop-down menu, select to act as server or client (depending on the settings of the connected device).
3. Set the port number in the **TCP Port** text field. 2114 is the default port.

Serial interface

To send the output string via the serial interface:

1. Select the **Output string** checkbox in the **Serial** section in the parameter pane.

EtherNet/IP

To send the output string via EtherNet/IP, see "Setting up a TriSpector1000 to communicate via EtherNet/IP", page 92.

7.8.4 Logging images to an FTP server

To log images to an FTP server:

1. Select the **FTP image logging** checkbox in the **Ethernet** section.
2. Specify the IP address, port, username and password to connect to an FTP server.
3. Test the connection to the server by clicking **Test connection**.

✓ A message with the connection status appears in the GUI.

4. Specify the log condition in the **Image logging** section in the **Results** workflow step, see "Image logging", page 63.

The logged images are saved in Portable Network Graphics (.png) format, and contain both height data and reflectance data.

Destination folder

By default, the logged images are saved to the specified folder on the FTP server. To create a sub-folder and save the images there, define the path to the sub-folder in the **Path** field in the **FTP image logging** section.

7.8.5 Enabling the web interface

Enable the **Web server** option to view the images acquired by the TriSpector1000 in a web browser. Google Chrome is recommended as web browser due to advanced support of WebGL and web sockets.

NOTE
In order to access the web interface, the computer and the TriSpector1000 must be on the same subnet.

To view images in a web browser:

1. Select the **Web server** checkbox in the **Ethernet** section.
2. Open a web browser window.
3. Type the IP address of the TriSpector1000 in the address field.

The page displayed in the web browser views the live 3D image, the name of the current job, the image number, the results for all applied tools and the overall image decision. Use the eye symbol above the image area to show the applied tools as overlay to the image.

NOTE
If an error message is displayed in the web browser, it means that the **Web server** checkbox in SOPAS ET is not selected.

7.8.6 Setting parameters of the serial interface

The serial interface is available on the Power I/O connector, see "Pin assignment", page 32. The following parameters can be set:

- **Baud rate**: Data rate, in bits per second (bps).
- **Data bits**: The number of data bits.
- **Stop bits**: The number of stop bits.
- **Parity**: Whether to use parity (odd or even).

Command channel

Select the **Command channel** checkbox to use the serial interface as command channel. see "Using the command channel", page 66 for more information.

Output string

Select the **Output string** checkbox to send a result string via the serial interface. see "Sending an output string", page 67 for more information.

7.8.7 Logging images to a microSD memory card

Follow the instructions below to log images to a microSD memory card:

NOTE
If no microSD memory card is inserted or if there is not enough memory available, no images are logged. The images currently stored on the SD card are not overwritten.

1. Select the **SD card image logging** checkbox in the **SD Card** section in the **Interfaces** workflow step to activate the image logging.
2. Specify the log condition in the **Image logging** section in the **Results** workflow step, see "Image logging", page 63.

The logged images are saved in Portable Network Graphics (.png) format, and contain both height data and reflectance data.

Destination folder

The first time an image is logged, a default folder named `TriSpectorImages` is created on the microSD memory card. To create a sub-folder and save the images there, define the path to the sub-folder in the **Path** field in the **SD Card** section. If the **Path** field is empty, all images are saved to the default folder.

Options

The following options are available in the **SD Card** section:

- Click **SD Card Info** to view the currently available memory on the microSD memory card.
- Click **Export images** to export all images in the default folder and the underlying sub-folders from the microSD memory card to the PC.
- Click **Delete all images** to delete all images from the default folder and the underlying sub-folders on the microSD memory card.
- Click **Eject** to stop all access to the data on the microSD memory card before removing the card from the device.

Monitoring the log status

Three parameters related to the image logging are available in the **Conditions** and **Ethernet output string** sections in the **Result** workflow step:

- **ImageQueueLength** (the number of images placed in the log queue)
- **MaxImageQueueLength** (the maximum number of images which can be placed in log queue)
- **AvailableMemory** (the available memory on the microSD memory card)

Use the above parameters to create conditions for monitoring the image logging, see figure 45 and figure 46 for examples. For further information on how to create and use conditions, see "Conditions", page 59.

Figure 45: Example of a condition which returns true if the available memory on the microSD card is less than 10 MB

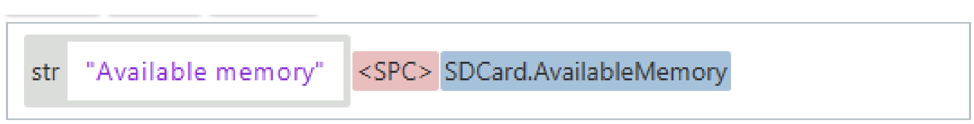


Figure 46: Example of a result output string which contains the available memory on the microSD card (in bytes)

7.8.8 Setting the device time

Edit the **Device time** section in the **Interfaces** workflow step to set the date and time on the device. Use the set device time to create timestamps for logged images, created conditions and output strings.

Time source

Set the device time based on user input or based on information from a time server.

NOTE
If the device time is based on user input, it will be reset to default after a device reboot.

To set the device time based on user input:
1. In the **Time source** list, select **User input**.
2. Edit the date and time in the assigned fields. The date and time for the connected PC are entered by default.
3. Click **Set time on device** to set the device time based on the entered values.

To set the device time based on information from a time server:
1. In the **Time source** list, select **Time server**.
2. Enter the IP address and port for connecting to the time server.
3. Click **Update time from server** to establish a connection between the device and the time server.

Default device time

At the first use of the device, the device time is set to 1970-01-01 00:00 by default. If no time source has been entered, the default value is reset when restarting or rebooting the device.

Click **Get current time from device** to display the currently set device time.

Using the device time as a result parameter

To include the device time in a condition or in an output string:
1. Click the **Result** workflow step.
2. In the **Conditions** section or the **Ethernet output string** section, click **+Result**.
3. Point to the **Datetime** row and click **Date** to include the device date or **Time** to include the device time.

7.8.9 Selecting jobs via digital inputs

The **Job Selection Input** section enables job selection via digital input signals. The number of available inputs depend on the number of I/O configured as inputs in the **I/O Definitions** section, see "Configuring the digital in- and outputs", page 66.

NOTE
The input signals must be kept high during the whole session, not only when the job is initiated.

- Each row in the list represents a unique combination of available inputs.
- Select a job in the **Job** drop-down list to match it with a set of inputs.
- Click the **Auto-fill** button to include all available jobs in the list (one job per row).
- Click the **Clear** button to remove all jobs from the list.

7.8.10 Fieldbuses

The TriSpector1000 has native EtherNet/IP support. To connect the TriSpector1000 to a PROFINET network, use the fieldbus module CDF600-2200 from SICK. See "Connecting the TriSpector1000 to CDF600-22xx", page 89 for details.

7.9 Using the SOPAS ET emulator

Use the SOPAS ET emulator to simulate and test device settings off-line without access to a real TriSpector1000. The main differences when using the emulator compared to a real device are:

- Images will not be affected by changes made in the **Image** workflow step.
- Processing times may differ, since the emulator uses the PC hardware instead of the device hardware.
- The digital I/O connections are disabled.

7.9.1 Starting the emulator

NOTE
The emulators are available via the SOPAS Device Drivers (SDD). An SDD is installed at the first use of the TriSpector1000. If no SDD is installed, the Emulators tab in SOPAS ET is empty and the SDD must be installed manually, as described below.

Starting the emulator when connected to a TriSpector1000

1. In the SOPAS ET device window, click **Switch to simulated device** on the **Device** menu.

Starting the emulator when starting SOPAS ET

1. In the SOPAS ET main window, in the device pane, click the **Emulators** tab.

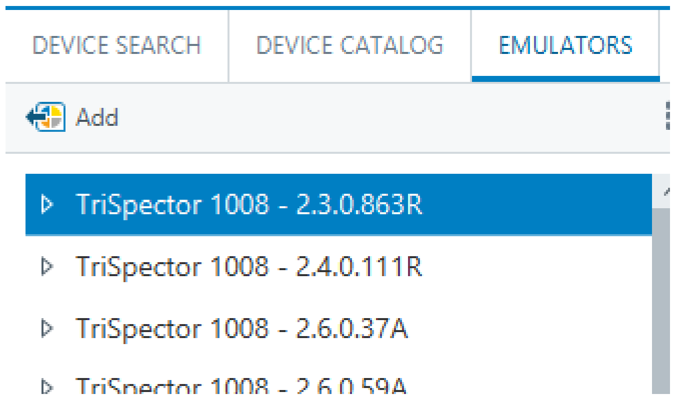


Figure 47: Emulators tab

2. Select the emulator corresponding to the suitable TriSpector1000 variant and click **Add**.
3. Double-click the emulator icon in the project pane to start the configuration.

Installing an SDD

If no SDD is installed, follow the steps below for manual installation.

1. In the SOPAS ET main window, in the device pane, click the **Device catalog** tab.
2. Click the device driver management button ().
✓ The **Device driver management** window opens.
3. Select **Install** and click Next.
4. Select www.mysick.com as source and click Next.
5. In the search field, type `TriSpector`.
✓ All available TriSpector1000 SDDs appear in the list of drivers.
6. Select the TriSpector1000 variant and the version to install (the latest version is recommended) and click Next.
✓ The installation starts.
7. When the installation is complete, click **Finish**.
✓ The emulators are now listed under the **Emulators** tab.

7.9.2 Controlling the emulator

The emulator buttons above the image area control the emulator. These buttons are only visible when using the emulator.

Loop through the images in the selected folder.

Stop the playback.

Step to the next image.

Step to the previous image.

Step to the first image in the series.

Open a dialog window for file selection

View options for the image series.

7.9.3 Selecting emulator images

Click the **Choose image folder** button to select images to use in the emulator. The images must be Portable Network Graphics (.png) files acquired by a TriSpector1000.

NOTE
Select any image in the **Choose image folder** window to include all images in the folder in the emulator.

To save images from the TriSpector1000 to PC, see "Recording images", page 44.

7.9.4 Copying a configuration from the emulator

When connected to a TriSpector1000

If the emulator was started by using the **Switch to simulated device** option:

1. In the SOPAS ET device window, click **Switch to physical device** on the **Device** menu.

When not connected to a TriSpector1000

If the emulator was started from the SOPAS ET main window:

1. In the SOPAS ET device window, click **Export...** on the **Device** menu.
✓ The **Device Export** window opens.
2. Save the configuration as a SOPAS ET (.sopas) file.
3. Connect to the TriSpector1000 from the SOPAS ET main window.
4. Load the configuration by clicking **Import...** on the **Device** menu.

NOTE
To save an imported configuration in the device´s flash memory, click **Save permanent** on the **Parameter** menu.

7.10 Exporting and importing a configuration

Exporting a configuration from the device to a PC

1. In the SOPAS ET functions panel, on the **Device** menu, click **Export...**
✓ The **Device Export** window opens.
2. Select file name and destination folder on the PC.
3. In the **Save as type** list, select **SOPAS (*.sopas) SOPAS ET file.**
4. Click **Save.**

Importing a configuration to the device from a PC

1. In the SOPAS ET functions panel, on the **Device** menu, click **Import...**
✓ The **Import** window opens.
2. Select a configuration (.sopas) file and click **Open.**

3. To save the configuration, click **Save permanent** on the **Parameter** menu.
✓ The configuration is saved to the device flash memory. If the device has a microSD memory card inserted, the configuration is saved to both the flash memory and the microSD memory card.

8 Maintenance

8.1 Maintenance plan

During operation, the device works maintenance-free.

NOTE
No maintenance is required to ensure compliance with the laser class.

Depending on the assignment location, the following preventive maintenance tasks may be required for the device at regular intervals:

Table 14: Maintenance plan

Maintenance work	Interval	To be carried out by
Check device and connecting cables for damage at regular intervals.	Depends on ambient conditions and climate.	Specialist
Clean housing.	Depends on ambient conditions and climate.	Specialist
Check the screw connections and plug connectors.	Depends on the place of use, ambient conditions or operating requirements. Recommended: At least every 6 months.	Specialist
Check that all unused connections are sealed with protective caps.	Depends on ambient conditions and climate. Recommended: At least every 6 months.	Specialist

8.2 Cleaning

NOTICE
Equipment damage due to improper cleaning.
Improper cleaning may result in equipment damage.

- Only use recommended cleaning agents and tools.
- Never use sharp objects for cleaning.

▶ Keep the windows clean to maintain the read performance of the device. Regularly check the windows for contamination such as dust or humidity. If there is contamination, gently clean the window with a soft, damp cloth. Use a mild cleaning agent if needed.

▶ The windows in the PMMA variants can attract dust particles due to static charging. To minimize this effect, use the anti-static plastic cleaner from SICK (no. 560000006) together with the SICK lens cloth (no. 4003353).

8.3 Upgrading the firmware

The latest firmware for the device can be downloaded from the SICK Support Portal, supportportal.sick.com.

To upgrade the firmware:

1. Log in to the SICK Support Portal and navigate to the **TriSpector1000** page.
2. Click the latest firmware release.
3. Download the firmware package file (FWP file format) and the firmware key file (KEY file format).
4. Follow the instructions on the page to install the firmware.

NOTICE
Old firmware for the TriSpector1000 is available on the SICK Support Portal and can be used to downgrade the device. Before downgrading, make sure all application parameters are reset to default:

1. Click **Load application defaults** on the **Functions** menu in SOPAS ET.
2. Click **Save permanent** on the **Parameter** menu in SOPAS ET.

Failure to reset the parameters might result in a device that does not boot and needs re-flashing.

8.4 Data back-up and restoration

Creating a back-up of a configuration

For instructions on how to copy a TriSpector1000 configuration to a PC, see "Exporting and importing a configuration", page 72.

Resetting parameters to default

To reset all SOPAS ET parameters including IP address:

1 On the **Functions** menu, click **Load factory defaults.**
2 On the **Parameter** menu, click **Save permanently.**

To reset all SOPAS ET parameters except IP address:

1 On the **Functions** menu, click **Load application defaults.**
2 On the **Parameter** menu, click **Save permanently.**

9 Troubleshooting

9.1 General faults, warnings, and errors

Possible errors and corrective actions are described in the table below. In the case of errors that cannot be corrected using the information below, please contact the SICK Service department. To find your agency, see the back page of this document.

NOTE
Before calling, make a note of all type label data such as type designation, serial number, etc., to ensure faster telephone processing.

Table 15: Troubleshooting questions and answers

Question	Possible cause and resolution
The result is invalid.	The **Shape**, **Plane**, **Edge** and **Peak** tools require image data inside their region of interest to work. ▶ In Job mode, make sure that the regions of interest are placed correctly in 3D space. The **Shape** tool is unable to find any edges inside its region of interest. ▶ Try increase the robustness slider, or include a larger portion of the image for the **Shape** tool. For a **Shape** tool, the result might become invalid if the Profile distance parameter is changed after the job image is saved. If a **Shape** tool result is invalid, the results for all tools linked to it will also be invalid.
The laser LED is on, but there is no laser.	The laser module has a protection circuit and will shut itself down if it detects a voltage spike. Such a voltage spike is common if the TriSpector1000 is powered on by mechanically plugging in an already powered 24 V contact into the device. ▶ When all cables are connected, make sure that the power source is stable and that a power switch is used to power on the TriSpector1000.
The **Shape** tool does not find the object, it returns "Not Found".	The object contains areas with noise, or areas which change from sample to sample. ▶ Only use the **Shape** tool on edges that are present in all live images. ▶ In the tool parameter pane, in the **Masks** section, apply a mask to exclude unwanted parts of the object. ▶ Increase the robustness of the **Shape** tool. ▶ Decrease the edge threshold of the **Shape** tool to include more edges for matching. Do not include noise or areas that can change.
A reflective surface reflects the laser light, so it is difficult to get a good image.	Tilt the TriSpector1000 to adjust the camera-laser angle. ▶ To get more data from reflective surfaces, tilt the TriSpector1000 forward. A more specular camera-laser angle will result in more light entering the camera. ▶ To avoid noise due to a reflective surface, tilt the TriSpector1000 backwards for a more look-away camera-laser angle. The TriSpector1000 coordinate system also tilts when tilting the device, so height measurements in mm are no longer valid.

Question	Possible cause and resolution
It is difficult to get data at the bottom of a hole	By tilting the TriSpector1000 forward or backwards, different parts of the hole can be seen. The TriSpector1000 coordinate system also tilts when tilting the device, so height measurements in mm are no longer valid.
The TriSpector1000 is only receiving every 2nd (3rd, 4th...) image	Triggers which arrive while the image acquisition is still ongoing will be discarded as overtrigs and no new image acquisition will start. ▶ Reduce the image acquisition time by decreasing the **Length** parameter in the **Image** workflow step.

9.2 Repairs

Repair work on the device may only be performed by qualified and authorized personnel from SICK AG. Interruptions or modifications to the device by the customer will invalidate any warranty claims against SICK AG.

9.3 Returns

▶ Do not dispatch devices to the SICK Service department without consultation.
▶ The device must be sent in the original packaging or an equivalent padded packaging.

NOTE
To enable efficient processing and allow us to determine the cause quickly, please include the following when making a return:

- Details of the contact person
- Description of the application
- Description of the fault that occurred

9.4 Disposal

If a device can no longer be used, dispose of it in an environmentally friendly manner in accordance with the applicable country-specific waste disposal regulations. Do not dispose of the product along with household waste.

NOTICE
Danger to the environment due to improper disposal of the device.

Disposing of devices improperly may cause damage to the environment.

Therefore, observe the following information:

- Always observe the valid regulations on environmental protection.
- Separate the recyclable materials by type and place them in recycling containers.

10 Technical data

NOTE
The relevant online data sheet for your product, including technical data, dimensional drawing, and connection diagrams can be downloaded, saved, and printed from the Internet:

- www.sick.com/TriSpector1000

Please note: This documentation may contain further technical data.

10.1 Features

Attribute	Value
Tasks	Positioning, inspection, measurement
Technology	3D, line scan, image analysis
Working distance (measured from front win-dow)	1008: 56...116 mm range 1030: 141...541 mm range 1060: 291...1091 mm range
Width at minimum working distance	1008: 40 mm 1030: 90 mm 1060: 180 mm
Width at maximum working distance	1008: 75 mm 1030: 330 mm 1060: 660 mm
Maximum height range	1008: 60 mm 1030: 400 mm 1060: 800 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
Laser fan angle	45° ±2°
Imaging angle	1008/1030: 65° 1060: 67°
Offline support	Emulator
Toolset	Angle, Area, Blob, Distance, Edge, Fix Plane, Plane, Peak, Point, Shape

10.2 Ambient data

Attribute	Value
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	0 °C ... +40 °C (adequate heat dissipation is required, see "Mounting the device", page 22)
Ambient storage temperature	–20 °C ... +70 °C
Permissable relative air humidity	0% ... 90%, non-condensing

10.3 Interfaces

Attribute	Value
Operator interface	SOPAS
Configuration software	SOPAS
Communication interfaces	Gigabit Ethernet (TCP/IP), serial (RS-232), configurable digital I/O
Digital inputs	3 x, non-isolated
Digital in-/ outputs	4 x, non-isolated, configurable
Encoder interface	RS-422/TTL (DBS36E-BBCP02048)
Maximum encoder frequency	300 kHz
Fieldbuses	Ethernet/IP via Gigabit Ethernet port ProfiNet via serial interface connected to SICK module CDF600-2200

10.4 Mechanics and electronics

Attribute	Value
Power I/O	M12 male connector, 12-pin A-coded
Gigabit Ethernet	M12 female connector, 8-pin X-coded
Encoder	M12 female connector, 8-pin A-coded
Connector material	Brass, nickel-plated
Supply voltage	24 V DC, ± 20 % SELV + LPS (EN 60950-1) or ES1 + PS2 (EN 62368-1) or Class II (UL1310)
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 62368-1
Protection class	III
EMC	Immunity: EN 61000-6-2:2005 Emission: EN 61000-6-3:2007
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

10.5 Input switching levels

Attribute	Value
Input levels	Up to 30 V Stresses beyond this over-voltage level can cause permanent damage to the device.

Attribute	Value
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

10.6 Output switching levels

Attribute	Value
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 at 24 °C
Overcurrent protection	< 200 mA
Capacitive load	≤ 100 nF
Inductive load	≤ 1H (with use of external free-wheeling diode, otherwise permanent damage to the device can occur)

10.7 Performance

Attribute	Value
Typical height resolution (Near field/Far field)	1008: 20/50 µm 1030: 40/280 µm 1060: 80/670 µm
Scan rate	5000 profiles/s
Maximum number of profiles	2500 per image
3D profile resolution	1008: 0.049 mm/px 1030: 0.215 mm/px 1060: 0.43 mm/px

11 Accessories

NOTE
Accessories and where applicable mounting information can be found online at:

- www.sick.com/TriSpector1000

12 Annex

12.1 EU declaration of conformity/Certificates

The EU declaration of conformity and other certificates can be downloaded from the Internet at:

- www.sick.com/TriSpector1000

12.2 Licenses

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.

12.3 Cybersecurity

Protection against cybersecurity threats requires a comprehensive and holistic cybersecurity concept that must be continuously monitored and maintained. Such a concept consists of organizational, technical, process-related, electronic and physical defense levels and sets up appropriate measures for the different types of risk. SICK's products and solutions must be regarded as an integral part of this concept.

Information on Cybersecurity can be found at: www.sick.com/psirt .

12.4 Available commands for command channel

12.4.1 Sending a command

- Add the command delimiter bytes `stx` (0x02) and `etx` (0x03) to mark the beginning and end of a command.
- Use single spacing between a command and an argument, and between arguments.
- Use quoted text for string arguments and unquoted text for all other argument types.

12.4.2 Command examples

Select a job:

```
<stx>set job "job0"<etx>
```

Enable object triggering:

```
<stx>set imageTriggerMode Object<etx>
```

Get all settings for a tool named "Fix Plane 1":

```
<stx>get tool "Fix Plane 1"<etx>
```

Get the XZ Angle for a fixed plane tool named "Fix Plane 1":

```
<stx>get tool "Fix Plane 1" XZAngle<etx>
```

Set the XZAngle for a fixed plane tool to 42.0:

```
<stx>set tool "Fix Plane 1" XZAngle 42.0<etx>
```

12.4.3 General commands

Command	Arguments	Returns	Description
`set overtriggerQueueSize`	int: queue-size	result	Set queue size of over triggered image acquisitions. Currently only 0 and 1 are supported.
`get overtriggerQueueSize`		int: queue-size	Get queue size of over triggered image acquisitions.
`set job`	string: job name	result	Select active job, by name.
`get job`		string: job-name	Get the name of the active job.
`set exposure`	int: exposure	result	Set the sensor exposure time, in microseconds.
`get exposure`		int: exposure	Get the sensor exposure time, in microseconds.
`set gain`	real: gain	result	Set the sensor gain factor.
`get gain`		real: gain	Get the sensor gain factor.
`set fieldOfView`	real: width, real: height, real: x offset, real: z offset	result	Set the field of view.
`get fieldOfView`		real: width, real: height, real: x offset, real: z offset	Get the field of view.
`set heightmapLength`	real: length	result	Set the heightmap length.
`get heightmapLength`		real: length	Get the heightmap length.
`set imageTriggerMode`	enum: imageTriggerMode `[Input, None, Object, Command]`	result	Set the image trigger mode.
`get imageTriggerMode`		enum: imageTriggerMode `[Input, None, Object, Command]`	Get the image trigger mode.
`set laser`	enum: enable `[on, off]`	result	Turn the laser on or off.
`set laserThreshold`	int: threshold	result	Set the intensity threshold for detecting the laser.
`get laserThreshold`		int: threshold	Get the intensity threshold for detecting the laser.
`set profileDistance`	real: distance	result	Set the distance between profile acquisitions, in mm.
`get profileDistance`		real: distance	Get the distance between profile acquisitions, in mm.
`set profileTriggerMode`	enum: mode `[Encoder, FreeRunning]`	result	Set the profile trigger mode.

Command	Arguments	Returns	Description
get profileTrigger-Mode		enum: mode [Encoder, FreeRunning]	Get the profile trigger mode.
set encoderMode	enum: mode [PositionUp, Position-Down, Direc-tionUp, Direction-Down, Motion]	result	Set the encoder mode
get encoderMode		enum: mode [PositionUp, Position-Down, Direc-tionUp, Direction-Down, Motion]	Get the encoder mode
set pulsesPerMM	real: pulses	result	Set the number of encoder pulses per mm. Relevant only when profile Trigger-Mode is set to Encoder.
get pulsesPerMM		real: pulses	Get the number of encoder pulses per mm. Relevant only when profile Trigger-Mode is set to Encoder.
set speed	real: speed optional enum: option [Instant]	result	Specify the object move-ment speed, in mm/s. Rel-evant only when profileTrig-gerMode is set to FreeRun-ning. If the optional enum Instant is added, the speed is updated during the ongoing image acquisi-tion.
get speed		real: speed	Get the specified object movement speed, in mm/s. Relevant only when pro-fileTriggerMode is set to FreeRunning.
set triggerDelay-Time	int: delay	result	Specify the trigger delay in ms. Relevant only when triggerDelayMode is set to ms.
get triggerDelay-Time		int: delay	Get the trigger delay in ms. Relevant only when triggerDelayMode is set to ms.
set triggerDelay-Track	int: delay	result	Specify the trigger delay in mm. Relevant only when triggerDelayMode is set to mm.
get triggerDelay-Track		int: delay	Get the trigger delay in mm. Relevant only when triggerDelayMode is set to mm.

Command	Arguments	Returns	Description
`set triggerDelay-Mode`	enum: mode [ms, mm]	result	Specify whether triggerDe-layTrack (mm) or triggerDe-layTime (ms) is used.
`get triggerDelay-Mode`		enum: mode [ms, mm]	Get whether triggerDelay-Track (mm) or triggerDelay-Time (ms) is used.
`echo`	string: message	string: message	Returns the input message.
`trigger`			Trigger an image acquisi-tion. Usable only when imageTriggerMode is set to Command.
`save-to-flash`		result	Save the current configura-tion persistently.
`help`		string	Retrieve command channel help text.
`set ftpImageLogging`	enum: enable [on, off]	result	Enables/disables FTP image logging.
`get ftpImageLogging`		enum: enable [on, off]	Get whether FTP image log-ging is enabled or not

12.4.4 Tool commands

The tool settings in chapter 12.4.4.1 are used as identifiers for the `get tool` and `set tool` commands.

Command	Arguments	Returns	Available for EtherNet/IP	Description
`get tool`	string: tool name identifier: setting name (optional)	tool info	Yes	Get either all settings and tolerances for a tool or a specified setting and toler-ance for a tool.
`set tool`	string: tool name identifier: setting name parameter value	boolean: [OK, Not OK]	Yes	Set the value of a setting or a tolerance for a tool.
`get tools`		tool info	No	Get the settings and toler-ances for all tools added to the current job
`get jobs`		list of jobs	No	Get a list of all jobs in the configuration

12.4.4.1 Tool settings

Find tools

*Table 16: **Shape** tool*

Setting	Data type	Value range
`MinEdgeThreshold`	real	0.1, 100
`MinScoreTolerance`	int	0, 100
`MinSearchScore`	int	0, 100
`Robustness`	int	0, 100
`Rotation`	int	0, 180
`RotationTolerance`	realrange	-180, 180

*Table 17: **Blob** tool*

Setting	Data type	Value range
`Accuracy`	int	0, 3
`Area`	realrange	0.04, 10000
`BoundingBoxType`	enum	`SmallestEnclosing, Using-BlobAngle`
`CalculateAngles`	boolean	`true, false`
`CalculateBounds`	boolean	`true, false`
`CutOffDirection`	enum	`NotSet, Above, Below`
`CutOffPlaneTool`	PlaneRef	-
`Erode`	int	0, 10
`ExcludeBorderBlobs`	boolean	`true, false`
`FillHoles`	boolean	`true, false`
`InvertBlobs`	boolean	`true, false`
`NumBlobsTolerance`	intrange	0, 250
`ShowBlobMarkers`	boolean	`true, false`
`SortBy`	enum	`Area, X, Y`
`SortDirection`	enum	`Ascending, Descending`
`VolumeBound`	enum	`Disabled, Bottom, Plane`
`VolumePlaneTool`	PlaneRef	-
`VolumeTolerance`	realrange	0, 3000

*Table 18: **Edge** tool*

Setting	Data type	Value range
`EdgeSide`	enum	`High, Low`
`EdgeThreshold`	real	0.1, 100
`MeanDeviationTolerance`	realrange	0, 10
`MinScoreTolerance`	int	0, 100
`OutlierDistance`	real	0.01, 20
`OutlierMode`	enum	`Auto, Manual`
`Polarity`	enum	`Any, Positive, Negative`
`ProbeSpacing`	int	1, 20
`Selection`	enum	`Strongest, First, Last`
`Sharpness`	int	1, 10

*Table 19: **Plane** tool*

Setting	Data type	Value range
`MinScoreTolerance`	int	0, 100
`Offset`	real	-60, 60
`Percentile`	intrange	0, 100
`Robustness`	int	10, 100
`ShowUsedPoints`	boolean	`true, false`
`TiltTolerance`	realrange	0.0, 90.0

*Table 20: **Fix Plane** tool*

Setting	Data type	Value range
`PositionX`	real	-40,40

Setting	Data type	Value range
PositionY	real	0, 400
PositionZ	real	-20, 80
XZAngle	real	-89, 89
YZAngle	real	-89, 89

*Table 21: **Intersection** tool*

Setting	Data type	Value range
FeatureA	FeatureRef	-
FeatureB	FeatureRef	-
IntersectionProjection	enum	XYPlane, UserPlane
ProjectionPlane	PlaneRef	-
ZFrom	enum	FeatureA, FeatureB, Average

*Table 22: **Peak** tool*

Setting	Data type	Value range
HeightTolerance	realrange	-
KernelSize	int	2, 6
Type	enum	Max, Min
UseMedianFilter	boolean	true, false

*Table 23: **Point** tool*

Setting	Data type	Value range
PositionX	real	-1000, 1000
PositionY	real	-1000, 1000
PositionZ	real	-1000, 1000

Inspect tools

*Table 24: **Area** tool*

Setting	Data type	Value range
CoverageTolerance	realrange	0, 100
IntensityRange	intrange	0, 255
UseIntensity	boolean	true, false

Measure tools

*Table 25: **Distance** tool*

Setting	Data type	Value range
DistanceTolerance	realrange	0, 700
FeatureA	FeatureRef	-
FeatureB	FeatureRef	-
MeasureInX	boolean	true, false
MeasureInY	boolean	true, false
MeasureInZ	boolean	true, false
MeasureType	enum	RightAngle, PointToPoint
Offset	real	-

*Table 26: **Angle** tool*

Setting	Data type	Value range
`AngleProjection`	enum	XYPlane, UserPlane
`AngleTolerance`	realdeviation	0, 359.99; 0, 180
`FeatureA`	FeatureRef	-
`FeatureB`	FeatureRef	-
`ProjectionPlane`	PlaneRef	-

12.5 Connecting the TriSpector1000 to CDF600-22xx

This is an instruction for connecting the TriSpector1000 to the SICK fieldbus module CDF600-22xx.

Setting up the connection

Use either pre-assembled cables or open-ended cables and connection terminals to connect the TriSpector1000 to the CDF600-22xx.

NOTICE
If pre-assembled cables are used for the connection, the CDF600-22xx cannot access the digital I/Os on the TriSpector1000.

To connect using pre-assembled cables:

1. Connect the TriSpector1000 to the CDF600-22xx by a connection cable (e.g. part no 2055419) and a cable adapter (part no 2086398) according to figure 48.

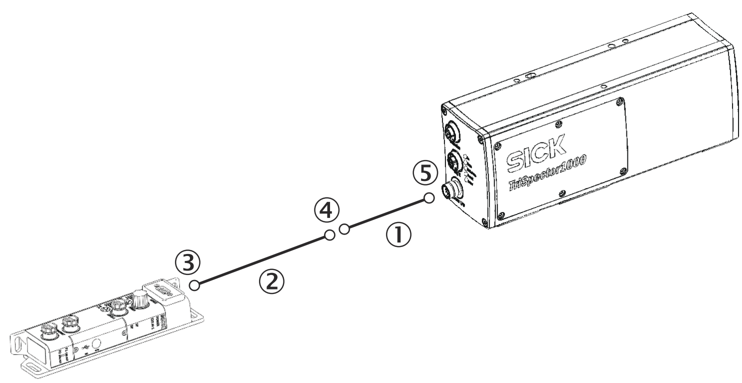


Figure 48: Connecting the TriSpector1000 to the CDF600-22xx by pre-assembled cables

① Adapter cable (2086398)
② Connection cable (e.g. 2055419)
③ D-Sub HD, 15-pin
④ M12, 17-pin
⑤ M12, 12-pin

2. Connect the CDF600-22xx to a 24 V power supply.
3. Set the rotary encoding switch found on the side of the CDF600-22xx to "2".

To connect using open-ended cables and connection terminals:

1. Connect the pins according to the table below.

Pin on TriSpector1000 (Power I/O connector)	Pin on CDF600-2200 (DEVICE connector)	Connection type
Pin 11	Pin 2	Serial (RX to TX)
Pin 12	Pin 3	Serial (TX to RX)
Pin 2	Pin 5	GND to GND
Pin 1	Pin 1	24 V to 24 V (optional)[1]

1 The 24 V to 24 V connection allows the TriSpector1000 to be powered by the CDF600-22xx.

2. Set the rotary encoding switch found on the side of the CDF600-22xx to "2".

Output data transmission - Output string

The TriSpector1000 sends data to the CDF600-22xx as an output string. The CDF600-22xx creates an array based on the string received from the TriSpector1000 and adds an overhead of seven additional characters.

To configure the TriSpector1000 for the data transmission:

1. Go to the **Interfaces** workflow step in SOPAS ET.
2. In the **Serial** section, set **Baud rate** to 57600, **Data bits** to 8, **Stop bits** to 1 and **Parity** to NONE.
3. Select the **Output string** checkbox in the **Serial** section to enable data transmission from the TriSpector1000 to the CDF600-22xx.

To create the output string:

1. Define the output string in the **Ethernet output string** section in the **Result** workflow step in SOPAS ET.

Figure 49: Example string

2. From the **+Char** list, add `<STX>` to the beginning of the output string and `<ETX>` to the end. For an example string, "Example", the output in the editor should look like figure 49.

Input data transmission - Command channel

Data from the CDF600-22xx to the TriSpector1000 is received via the command channel.

To configure the TriSpector1000 for the data transmission:

1. Go to the **Interfaces** workflow step in SOPAS ET.
2. In the **Serial** section, set **Baud rate** to 57600, **Data bits** to 8, **Stop bits** to 1 and **Parity** to NONE.
3. Select the **Command channel** checkbox in the **Serial** section to enable data transmission from the CDF600-22xx to the TriSpector1000.

The command channel requires no further configuration on the TriSpector1000 side. Commands, for example "trigger", can be sent from the PLC via the CDF600-22xx.

Further information

For information about configuration and operation of the CDF600-22xx, see the CDF600-2xxx product page, www.sick.com/CDF600-2. The following information is available there:

- Technical information
- Function blocks (including documentation) for sending strings to and from the CDF600-22xx using Profinet and Profibus

12.6 Data streaming using the TriSpector1000

In addition to internal processing of acquired image data, the TriSpector1000 can be used as a data streamer. When set up for data streaming, the TriSpector1000 provides data which can be received according to the GenICam GenTL 1.5 standard. For more information about the GenICam GenTL 1.5 standard, see www.emva.org/standards-technology/genicam/.

During data streaming, the TriSpector1000 sends out 8-bit reflectance data and 16-bit range data together with the metadata needed for converting range data into millimetres. The data is sent to the host PC via a GenTL interface. A software application which can access the GenTL interface is required for receiving and processing the streamed image data.

Overview of interfaces

See figure 50 for an overview of the interfaces required for data streaming. Explanations of the interfaces and concepts are included below the image.

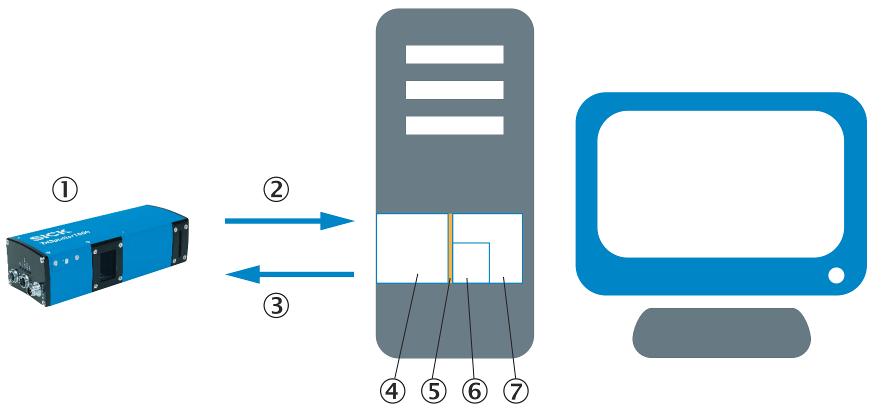


Figure 50: Interfaces for data streaming according to the GeniCam GenTL 1.5 standard

① TriSpector1000 (Remote device)
② Data stream
③ Camera control
④ GenTL Producer software
⑤ GenTL interface
⑥ GenAPI module
⑦ GenTL Consumer software

Camera control means external configuration of the camera via the GenTL interface.

The **GenTL interface** is an interface which provides a port for accessing the device. The GenTL interface does not include any device-specific functionality, except for the functionality needed to establish the communication with the device.

The **GenTL Producer** is a software module which implements a GenTL interface. The GenTL Producer allows applications and software libraries to access and stream data from the device.

The **GenAPI module** is an application programming interface (API) which is required for accessing and controlling device features.

The **GenTL Consumer** is the receiving software, which implements a GenAPI module to access and stream data from the device via the GenTL interface.

Setting up the TriSpector1000 for data streaming

The software needed to set up the TriSpector1000 for data streaming is available from the SICK Support Portal, supportportal.sick.com/downloads/trispector-streamer/. The following components are available there:

- A complete tutorial including description of application examples
- TriSpectorGenTL Producer software (32-bit and 64-bit) for streaming data from the TriSpector1000 to a Windows PC via a GenTL interface.
- An example program for integrating the TriSpector1000 with HALCON
- An example application, including source code for creating your own receiving software

12.7 Setting up a TriSpector1000 to communicate via EtherNet/IP

12.7.1 Interfacing TriSpector1000 with an Allen Bradley/Rockwell Programmable Controller

This is a procedure used to set up a TriSpector1000 to a Rockwell Automation/Allen Bradley Controller. The hardware used:

- SICK TriSpector1000 sensor
- Rockwell Automation/Allen Bradley 1756-L75 ControlLogix5561 Controller
- Rockwell Automation/Allen Bradley 1756-ENBT/A, 10/100Mbps Ethernet Bridge
- Rockwell Automation/Allen Bradley 1756-A4 Chassis
- Intel PC running Windows 7

The hardware is set up in a simple network as shown in figure 51.

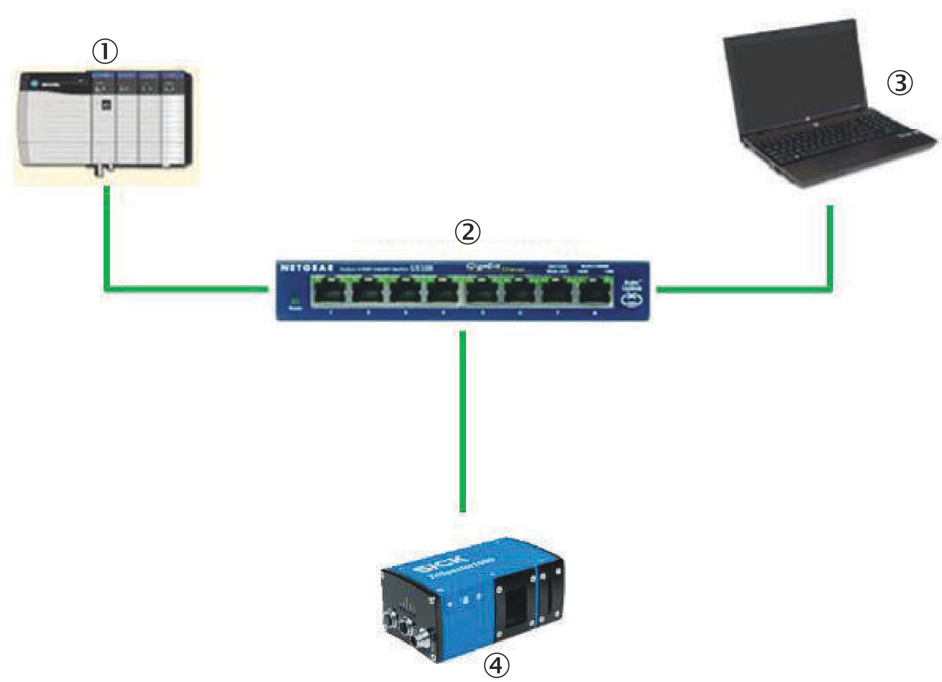


Figure 51: Hardware network

① PLC

② Ethernet switch

③ Configuration PC

④ TriSpector1000

The software used to configure the TriSpector1000 to the Rockwell Automation/Allen Bradley controller is:

- SOPAS Engineering Tool (ET), version 3.3.1 or newer
- RSLinx Classic Lite Version 3.71
- Studio 5000 PLC programming software version 24

Use SOPAS ET to modify the TriSpector1000's IP address. Once a connection is made and the device is discovered on the network, the IP address can be modified through the **TCP/IP Settings** menu (see figure 52).

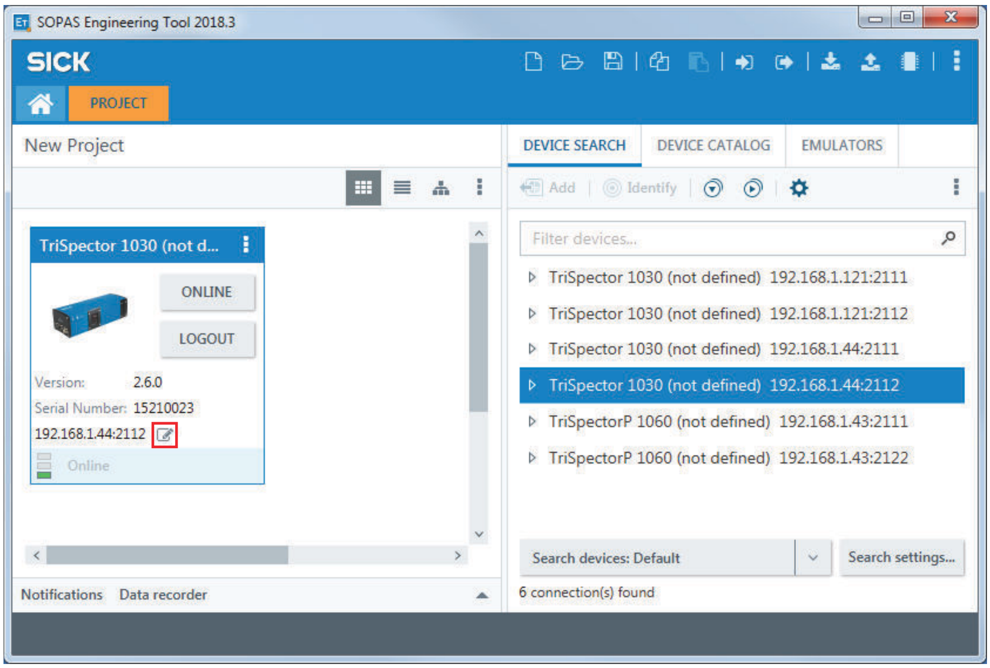


*Figure 52: **TCP/IP Settings** menu (red square)*

A static IP Address should be assigned to the TriSpector1000. If a DHCP server exists, the TriSpector1000 can be configured to receive its IP Address.

After an IP Address has been assigned to the TriSpector1000, it can be configured for the end user's application by following the workflow in the TriSpector1000 device window, see figure 53.

Figure 53: Workflow bar

EtherNet/IP is active in the TriSpector1000. The data generated by the TriSpector1000 is configured in the **Ethernet output string** section in the **Results** workflow step (see figure 54).

Figure 54: Ethernet output string

The result string appears within the programmable controller's '**Controller Tags**'.

12.7.2 Input and Output data assemblies - TriSpector1000

The TriSpector1000 connects to the programmable controller over EtherNet/IP. This connection is configured in the programmable controller's 'Generic Ethernet Module' profile. For details, see "Inside the Programmable Controller", page 96.

The data exchanged between the TriSpector1000 and the programmable controller includes three data assembly instances:

- **Input** contains the data from the TriSpector1000 to the programmable controller.
- **Output** contains the data from the programmable controller to the TriSpector1000.
- **Configuration** is a dummy which is not actually used.

12.7.2.1 Input data format

The input data format described in table 27 shows the byte allocation as well as the bit description for the entire input data format for the TriSpector1000.

Table 27: Input data format

Parameter	Byte	Bit	Description	Comment
Status Word	BYTE 0	D0	Device Ready	The state of the TriSpector1000 is set to 0 while booting up or configuring, and set to 1 when the device is ready. “Ready” indicates that the internal micro-processor is ready. It turns on along with the “On" LED.
		D1	Input Status 1	The state of the digital inputs.
		D2	Input Status 2	
		D3	Input Status 3	
		D4	Input Status 4	
		D5	Input Status 5	
		D6	Input Status 6	
		D7	Input Status 7	
	BYTE 1	D0	Output Status 1	The state of the digital inputs.
		D1	Output Status 2	
		D2	Output Status 3	
		D3	Output Status 4	
		D4	Heartbeat	The TriSpector1000 provides this heartbeat bit to the fieldbus master to show its presence and availability on the connection. The heartbeat will toggle between high (1) and low (0) in a fixed interval.
		D5	-	Reserved
		D6	-	Reserved
		D7	-	Reserved
Error Code	BYTE 2			Command response error code
ID	BYTE 3			Command response ID
Cmd Length	BYTE 4			Command response string length
Cmd Response	BYTE 5 through BYTE 36			Command response string
Result Status	BYTE 37			Result string status
Result Length	BYTE 38	LSB		Result string length
	BYTE 39	MSB		
Result	BYTE 40 through BYTE 493			Result string

12.7.2.2 Output data format

The output data format is described in table 28. This table shows the byte allocation for the entire output data format for the TriSpector1000.

Table 28: Output data format

Parameter	Byte	Comment
Cmd ID	BYTE 0	Command channel ID
Cmd Length	BYTE 1	Command string length

Parameter	Byte	Comment
Command	BYTE 2 through BYTE 129	Command channel string

12.7.3 Inside the Programmable Controller

The TriSpector1000 is set up in the programmable controller as a '**Generic Ethernet Module**'.

To set up a new configuration (starting with hardware right out of the box), start a new project in Studio 5000, configure all the hardware used in the programmable controller (i.e. local I/O cards and communication modules), and save the configuration to the memory of the controller.

Follow the steps below to set up of the TriSpector1000 as a '**Generic Ethernet Module**':

1. Select **Ethernet** under the communication interface 1756-ENBT/A in Studio 5000 and click **New Module** to open the module selection window, see figure 55.

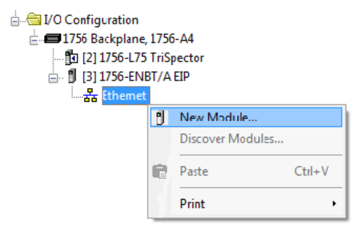


Figure 55: Open New Module window.

2. In the **Select Module Type** window, double-click the '**Generic Ethernet Module**' in the list of catalogs (see figure 56).

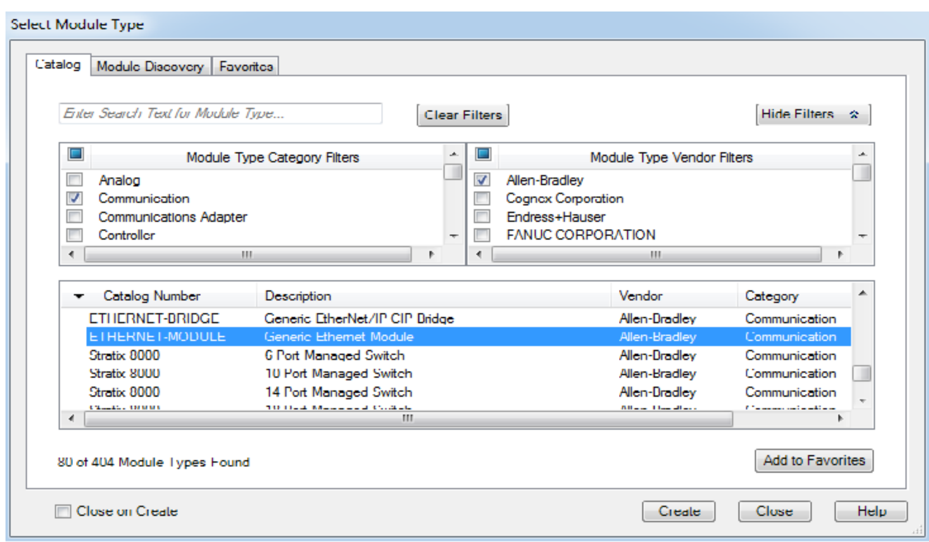


*Figure 56: '**Generic Ethernet Module**' selected in list*

✓ A 'New Module' window opens, see figure 57. Here, the TriSpector1000 data can be entered.

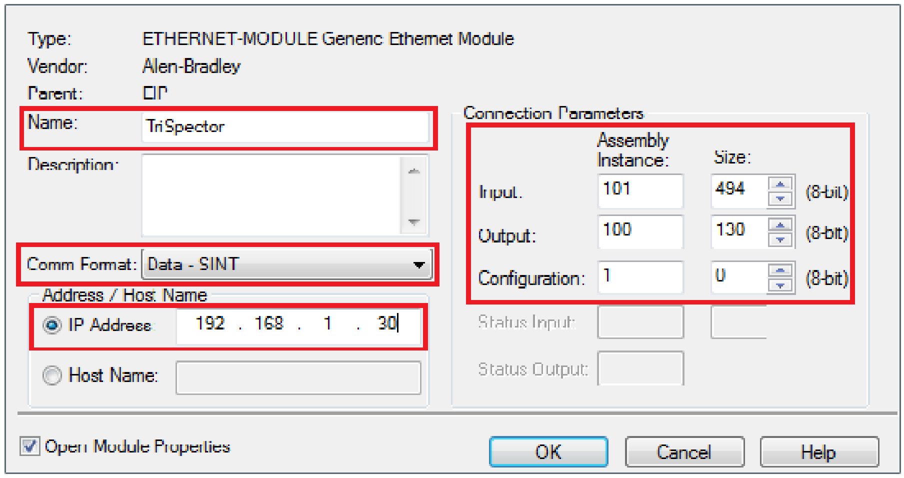


Figure 57: 'New Module' window

The following parameter values are required to set up the TriSpector1000 in the programmable controller:

Name	TriSpector
Comm Format	Data - SINT
IP Address	192.168.1.30

	Assembly Instance	Size
Input	101	494
Output	100	130
Configuration	1	0

The connection between the TriSpector1000 and the programmable controller uses Unicast messaging, see figure 58.

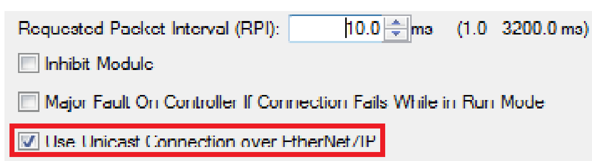
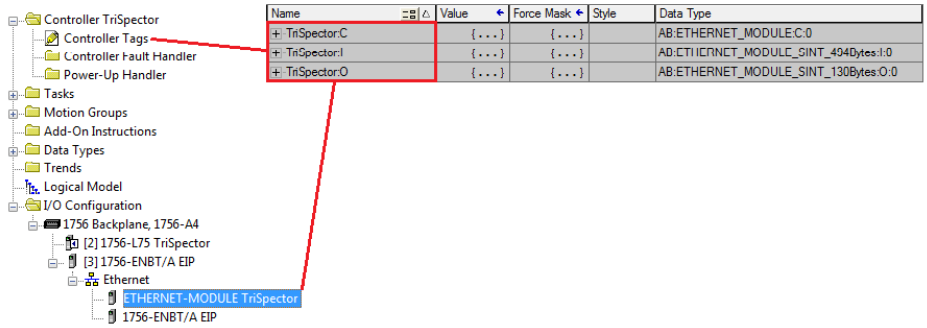


Figure 58: Enabling Unicast messaging

3. When the TriSpector1000 is configured, its data appears in the tag database as shown in the image below. The TriSpector1000 occupies three tags:

Tag	Used for
TriSpector:C	**Configuration** data
TriSpector:I	**Input** data
TriSpector:O	**Output** data



The programmable controller's software has three default windows, see figure 59:

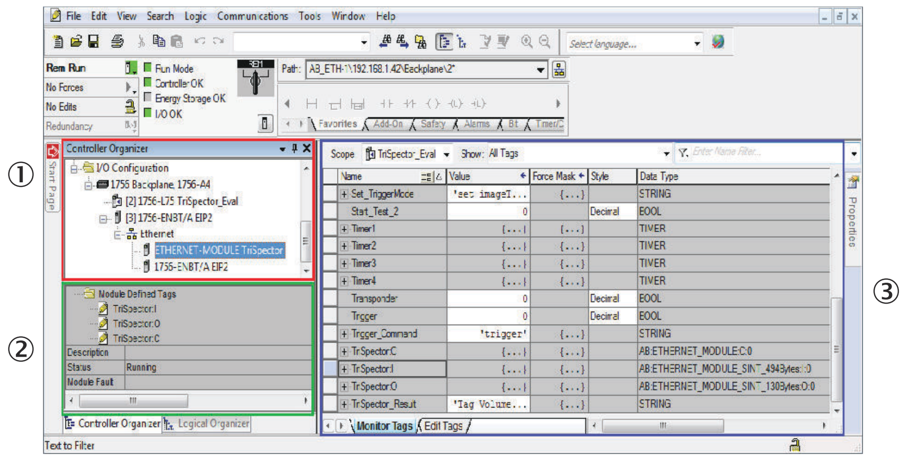


Figure 59: Software windows

① Controller Organizer

② Tag/Main routine

③ Diagnostic

Select a device under **I/O Configuration** in the **Controller Organizer** window to display its current status in the **Diagnostic** window. The following parameters are displayed:

Parameter	Description
Names of the Tags	The **Input**, **Output** and **Configuration** data which the device generates.
Description	User description of the device (entered during module setup).
Status	The current condition of the module (such as 'Running', 'Connecting' or 'Shutting Down').
Module Fault	The fault code and a description of what the controller is attempting to do to correct the problem.

Select and right-click the device name in the **I/O Configuration** section to access to the device properties.

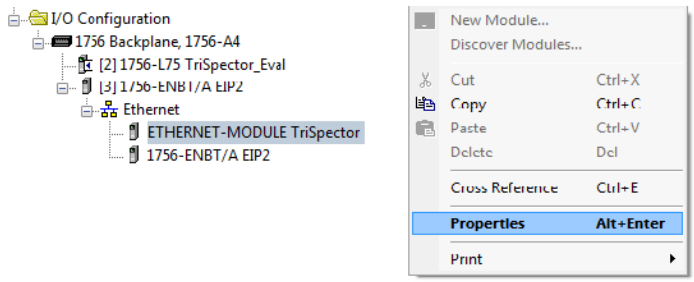


Figure 60: Accessing the device properties

In the properties window for the selected device, general module configuration (figure 61) and module information (figure 62) is available. The module information tab provides the current status of the module including faults, hardware revision, serial number and product name.

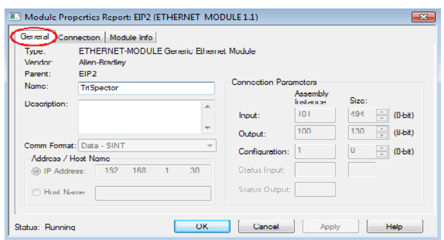


Figure 61: 'General' module configuration

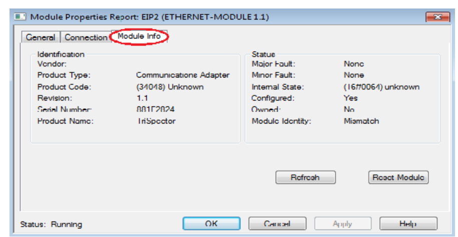


Figure 62: 'Module info'

12.7.4 Triggering a TriSpector1000

To trigger the TriSpector1000 over EtherNet/IP, a 'trigger' command is used. To configure the TriSpector1000 to receive its trigger by a command, go to the **Image** workflow step in SOPAS ET and set the **Image trigger** to **Command channel.**

With the TriSpector1000 configured to receive its trigger from the command channel, the programmable controller can be set up to write the ‘trigger’ command into the command channel string of the **Output** tag database.

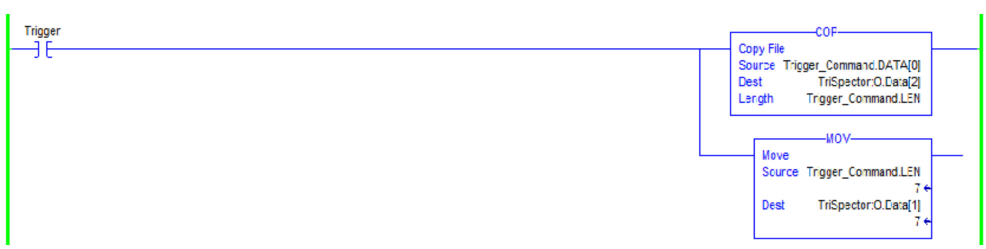


Figure 63: Write trigger command

This logic will copy the content of the User Defined Data Type (UDT) to the command channel string:

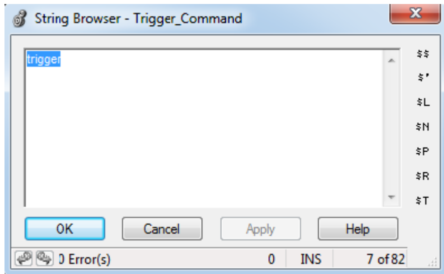


Figure 64: Content of UDT

Name	Value	Force Mask	Style	Data Type
TriSpector:O.Data	{...}	{...}	ASCII	SINT[130]
TriSpector:O.Data[0]	0		Decimal	SINT
TriSpector:O.Data[1]	7		Decimal	SINT
TriSpector:O.Data[2]	't'		ASCII	SINT
TriSpector:O.Data[3]	'r'		ASCII	SINT
TriSpector:O.Data[4]	'i'		ASCII	SINT
TriSpector:O.Data[5]	'g'		ASCII	SINT
TriSpector:O.Data[6]	'g'		ASCII	SINT
TriSpector:O.Data[7]	'e'		ASCII	SINT
TriSpector:O.Data[8]	'r'		ASCII	SINT
TriSpector:O.Data[9]	'$00'		ASCII	SINT
TriSpector:O.Data[10]	'$00'		ASCII	SINT

Figure 65: Command channel string

The results appear within the result string in the **Input** tag database, see figure 66.

Name	Value	Force Mask	Style	Data Type
TriSpector:I.Data[37]	'$00'		ASCII	SINT
TriSpector:I.Data[38]	4		Decimal	SINT
TriSpector:I.Data[39]	1		Decimal	SINT
TriSpector:I.Data[40]	'T'		ASCII	SINT
TriSpector:I.Data[41]	'a'		ASCII	SINT
TriSpector:I.Data[42]	'g'		ASCII	SINT
TriSpector:I.Data[43]	' '		ASCII	SINT
TriSpector:I.Data[44]	'V'		ASCII	SINT
TriSpector:I.Data[45]	'o'		ASCII	SINT
TriSpector:I.Data[46]	'l'		ASCII	SINT
TriSpector:I.Data[47]	'u'		ASCII	SINT
TriSpector:I.Data[48]	'm'		ASCII	SINT
TriSpector:I.Data[49]	'e'		ASCII	SINT
TriSpector:I.Data[50]	'='		ASCII	SINT
TriSpector:I.Data[51]	'2'		ASCII	SINT
TriSpector:I.Data[52]	'6'		ASCII	SINT
TriSpector:I.Data[53]	'.'		ASCII	SINT
TriSpector:I.Data[54]	'8'		ASCII	SINT
TriSpector:I.Data[55]	'5'		ASCII	SINT
TriSpector:I.Data[56]	'9'		ASCII	SINT
TriSpector:I.Data[57]	'1'		ASCII	SINT
TriSpector:I.Data[58]	'2'		ASCII	SINT
TriSpector:I.Data[59]	'9'		ASCII	SINT
TriSpector:I.Data[60]	','		ASCII	SINT
TriSpector:I.Data[61]	' '		ASCII	SINT
TriSpector:I.Data[62]	'T'		ASCII	SINT

Figure 66: Result string

12.7.5 Switching Jobs

The TriSpector1000 is capable of storing up to 32 jobs. To add jobs to the TriSpector1000 through SOPAS ET, select **Manage jobs** from the **Select jobs** list in the SOPAS ET functions panel.

Once jobs are added to the TriSpector1000, instructions to the programmable controller can be used to switch jobs in the TriSpector1000. The command used to switch jobs is `set job`. To select the correct job to switch to, the name of the job can also be used.

In this example there are two jobs in the TriSpector 1000: `Transponder` and `Coin`. To switch jobs in the TriSpector1000, two commands are required. For each command, two UDTs are created in the programmable controller (figure 67).

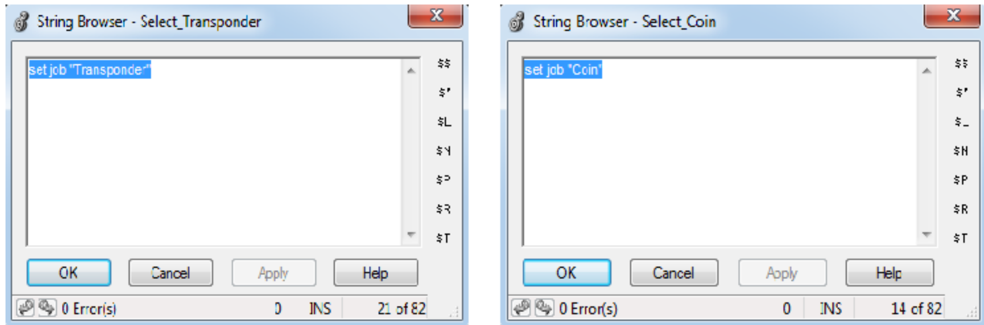


Figure 67: Commands for 'Transponder' job (left) and 'Coin' job (right)

The logic to execute the job switch is the same as the trigger.

To select the job named `Coin`:

Figure 68: Select the job named 'Coin'

To select the job named `Transponder`:

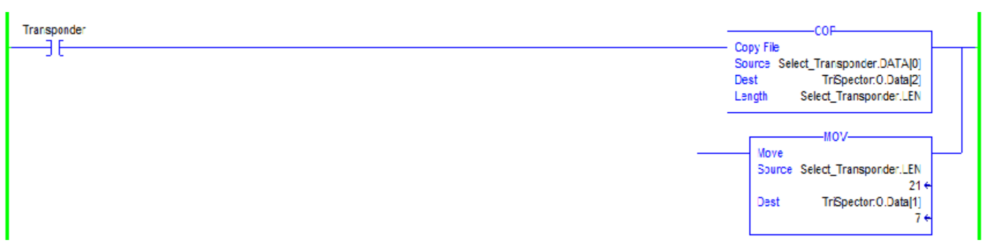


Figure 69: Select the job named 'Transponder'

When the command to select the job named `Coin` is sent from the programmable controller to the TriSpector1000 as a command channel string (figure 70), a successful response will be available in the command response string (figure 71).

TriSpector:O.Data	{...}	{...}	ASCII	SINT[130]
TriSpector:O.Data[0]	0		Decimal	SINT
TriSpector:O.Data[1]	14		Decimal	SINT
TriSpector:O.Data[2]	's'		ASCII	SINT
TriSpector:O.Data[3]	'e'		ASCII	SINT
TriSpector:O.Data[4]	't'		ASCII	SINT
TriSpector:O.Data[5]	' '		ASCII	SINT
TriSpector:O.Data[6]	'j'		ASCII	SINT
TriSpector:O.Data[7]	'o'		ASCII	SINT
TriSpector:O.Data[8]	'b'		ASCII	SINT
TriSpector:O.Data[9]	' '		ASCII	SINT
TriSpector:O.Data[10]	'"'		ASCII	SINT
TriSpector:O.Data[11]	'C'		ASCII	SINT
TriSpector:O.Data[12]	'o'		ASCII	SINT
TriSpector:O.Data[13]	'i'		ASCII	SINT
TriSpector:O.Data[14]	'n'		ASCII	SINT
TriSpector:O.Data[15]	'"'		ASCII	SINT
TriSpector:O.Data[16]	'$00'		ASCII	SINT
TriSpector:O.Data[17]	'$00'		ASCII	SINT

Figure 70: Command channel string

TriSpector:I.Data	{...}	{...}	ASCII	SINT[494]
TriSpector:I.Data[0]	-1		Decimal	SINT
TriSpector:I.Data[1]	16		Decimal	SINT
TriSpector:I.Data[2]	0		Decimal	SINT
TriSpector:I.Data[3]	0		Decimal	SINT
TriSpector:I.Data[4]	2		Decimal	SINT
TriSpector:I.Data[5]	'O'		ASCII	SINT
TriSpector:I.Data[6]	'K'		ASCII	SINT
TriSpector:I.Data[7]	'$00'		ASCII	SINT
TriSpector:I.Data[8]	'$00'		ASCII	SINT

Figure 71: 'Command response string'

12.7.6 TriSpector1000 EtherNet/IP Compatibility

Table 29: Programmable automation controllers

RA Bulletin Number	Product	Interface
1769	CompactLogix Controllers, 1769-L2 and 1769-L3 Series	Built-in EtherNet/IP Port
1768	CompactLogix Controllers, 1768-L4 Series	1768-ENBT Scanner
1756	ControlLogix, 1756-L6 Series	1756-EN2T Interface 1756-EN2F Interface 1756-ENBT Interface
1789	SoftLogix 5800 Controller	Computer NIC

Currently these controllers and the associated interface will provide EtherNet/IP network capabilities.

Online information on Rockwell Automation Programmable Logic Controllers, network interfaces, and software can be found at www.rockwellautomation.com.

12.8 TriSpector1000 Result Output function block for Siemens Simatic S7 PLC

12.8.1 About this document

This document will guide you through the process of connecting the TriSpector1000 to a Siemens Simatic S7 PLC, using the TriSpector1000 function block to parse an example string and how to modify the function block to suit your needs.

NOTE
This document does not provide a plug-and-play solution that will fit any result output string. This document is a guide for setting up the PLC for a specific example output string from the TriSpector1000. This document will also guide you how to modify the function block by editing the structured text implementation of the function block.

12.8.1.1 Target group

This instruction addresses planning engineers, developers and operators of machines and systems who use a TriSpector1000 vision sensor and who are familiar with Simatic S7 PLC programming using the TIA Portal.

12.8.1.2 Scope

This document applies to the following variants of TriSpector1000:

- TriSpector1008
- TriSpector1030
- TriSpector1060

12.8.1.3 Supported PLCs

Download the function blocks from the SICK Support Portal:

1. Log on to the SICK Support Portal.
2. From the **TriSpector1000** page, navigate to the **TriSpector PLC function blocks for Siemens S7** page.
3. Download the .zip file with the function blocks.

The zip file contains function blocks for the following controllers:

- S7-300
- S7-400
- S7-1200
- S7-1500

The table below tells you what function block to choose depending on your PLC version. The function blocks are different and not compatible with each other. The differences are explained in more detail in chapter 12.8.7.

Folder name	Compatible with Siemens S7 version
Siemens S7-1500,1200	S7-1500 and S7-1200 firmware version[1] 4.x
Siemens S7-1200	S7-1200 firmware version[1] > 2 and < 4
Siemens S7-300,400,1200	S7-300, S7-400 and S7-1200 firmware version[1] 1.x

1 "Firmware version" here refers to the firmware for the TRCV function block, not for the PLC. See "Function block differences", page 121 for more information.

NOTE
You must use the TIA Portal software to be able to use the function blocks. If you use the Simatic STEP7 version 5.5 software, it is not possible to use the function blocks.

12.8.2 Overview

The Result Output function block simplifies the use of a TriSpector1000 vision sensor on Simatic S7-1500/1200/400/300 controls. It is designed for TCP/IP communication. The function block supports the task to read and normalize an example output string from the TriSpector1000.

figure 72 shows the concept behind the TriSpector1000 PLC integration.

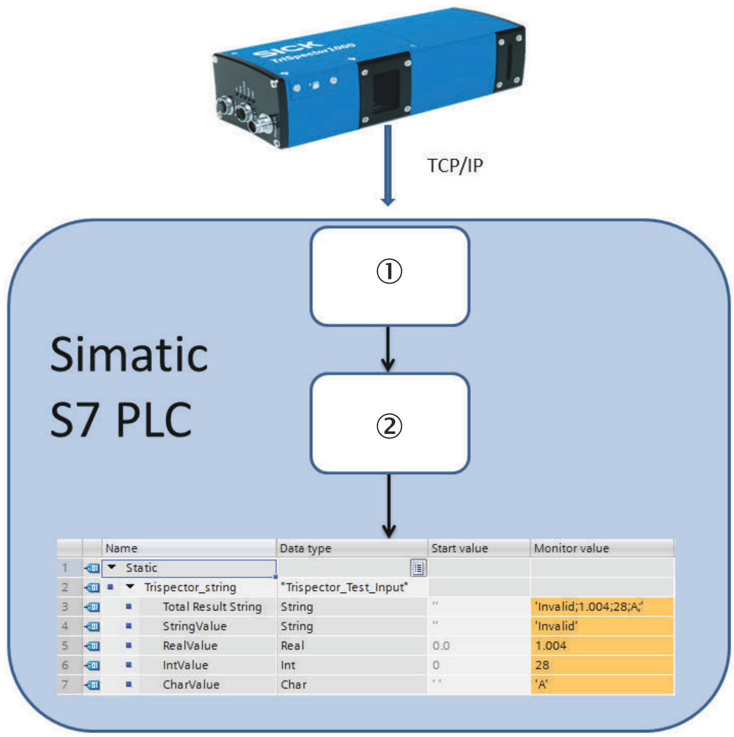


Figure 72: TriSpector PLC integration concept

① Result Output function block

② Result Output structure (described via user defined data type)

12.8.3 TriSpector1000 configuration

To integrate the TriSpector1000 into the TIA PLC environment, the current active result output configuration must be known in the PLC program. This function block is written to handle a specific example string. To follow this example, this string must be configured in the TriSpector1000.

12.8.3.1 Defining the output string

The user can define the TriSpector1000 Ethernet output string in SOPAS ET in any way she wants, but the function block can not handle any type of result string. This function block will only handle an example of a result string. It is up to the PLC programmer to make changes in the function block to make it work with the current output.

<STX>Invalid;1.004;28;A;<ETX>

Figure 73: Example of Ethernet output string

The output string in figure 73 is an example string for this function block. This string contains a string, a real number, an integer and a character. In the TriSpector1000 this corresponds to different data types but when sent over TCP/IP it is sent as a string. No other data types are transmitted. In the example above, the numeric value 1.004 will be a string of 5 characters that needs to be parsed and converted to a real number by the function block in the PLC. To know when a value starts and stops a delimiter is used; in the example above we use a semicolon between every value. The string must start with a <STX> character and end with a <ETX> character.

NOTE
To be able to use this function block, configure the above output string in the TriSpector1000.

12.8.3.2 Enabling output string over TCP

To enable the output string on Ethernet:

1. In SOPAS ET, go to the **Ethernet** section in the **Interfaces** workflow step.
2. Enable **Output string.**
3. Choose to act as server on default TCP port 2114 (see figure 74).

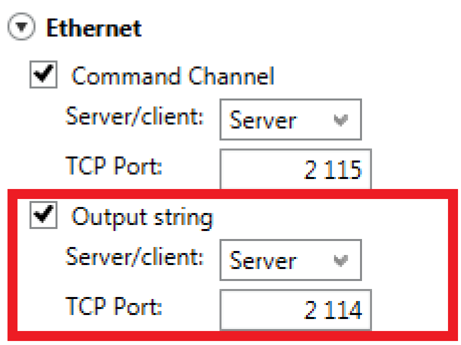


Figure 74: Ethernet settings

12.8.4 PLC Configuration

12.8.4.1 Establishing a TCP connection

The TriSpector1000 Result Output function block uses a TCP connection between the sensor and the S7 PLC to get the result output data. The TCP communication must be established by using the Siemens function block **TCON**.

Use the Siemens TCON function block

To establish an open TCP connection, the following communication blocks are available in the Siemens standard library (see figure 75):

- **TCON** for establishing the connection
- **TDISCON** for terminating the connection
- **T_CONFIG** for configuring the interface

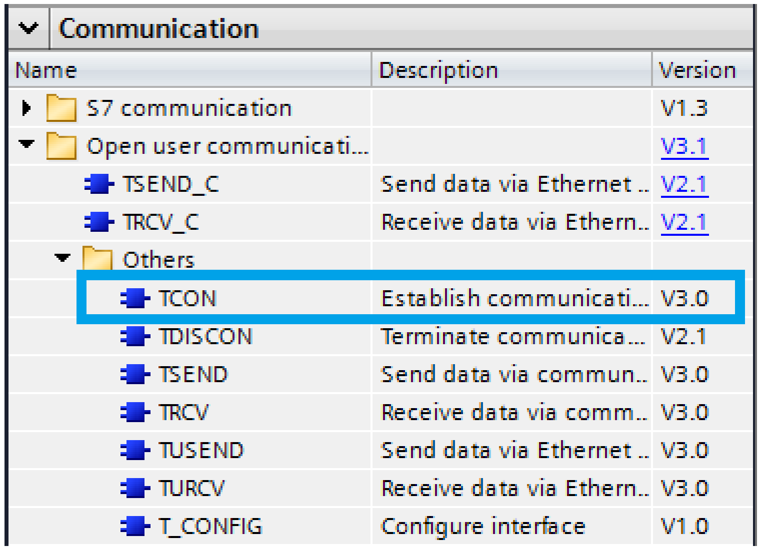


Figure 75: Siemens standard library

In this example, the **TCON** function block (see figure 76) is called in OB1 with the instance data block **TCON_DB**. In the case of a reboot, the "**#Initial_Call**" flag triggers the the **TCON** function block to open the TCP communication. The **ID** parameter of the TriSpector1000 function block input must have the same value as the **ID** input of the **TCON** function block (will be set up in next step).

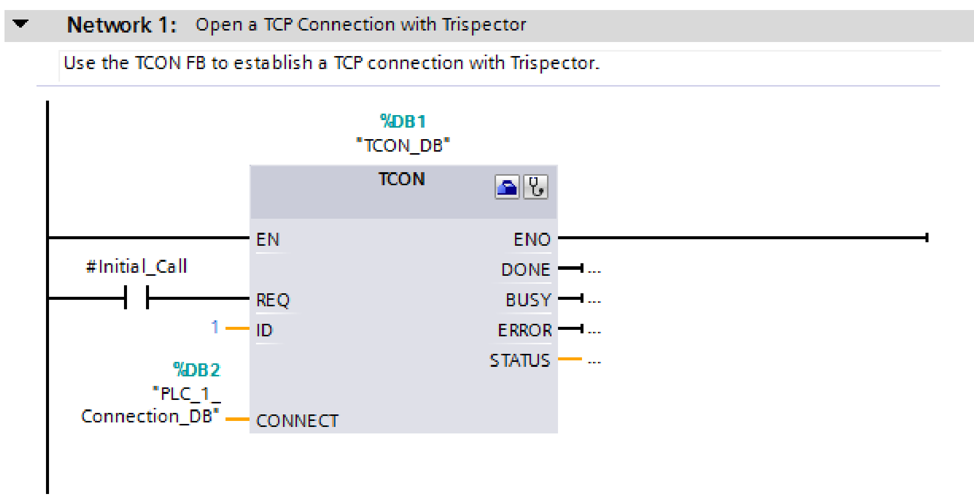


Figure 76: Establish TCP connection

The connection parameters for setting up the TCP connection are stored in an automatically generated data block. Open the **Properties** tab for the **TCON** function block to make the settings and generate the data block, see figure 77.

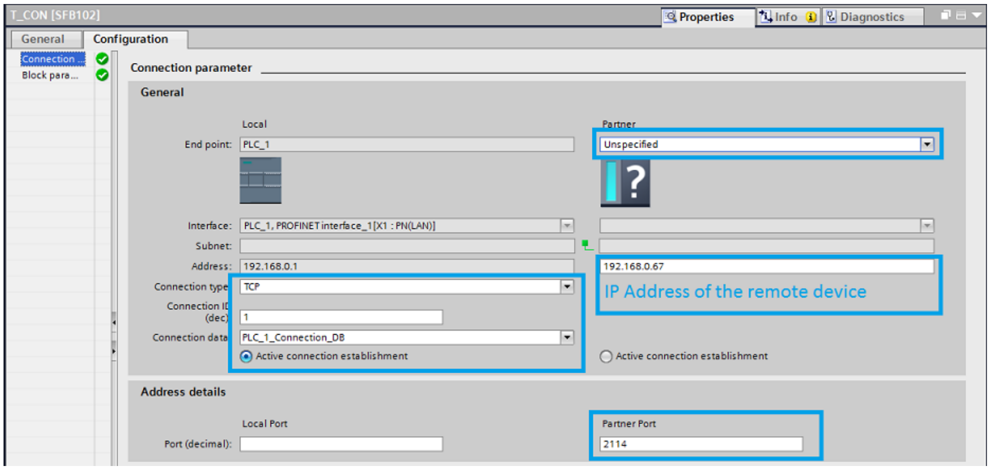


Figure 77: Connection parameters

For more information, see the TIA Portal help system.

12.8.4.2 Importing the TriSpector1000 Test Function Block

To get the generated blocks into the PLC project, the **TrispectorTest_FB.scl** file must be imported as an external source file, see figure 78.

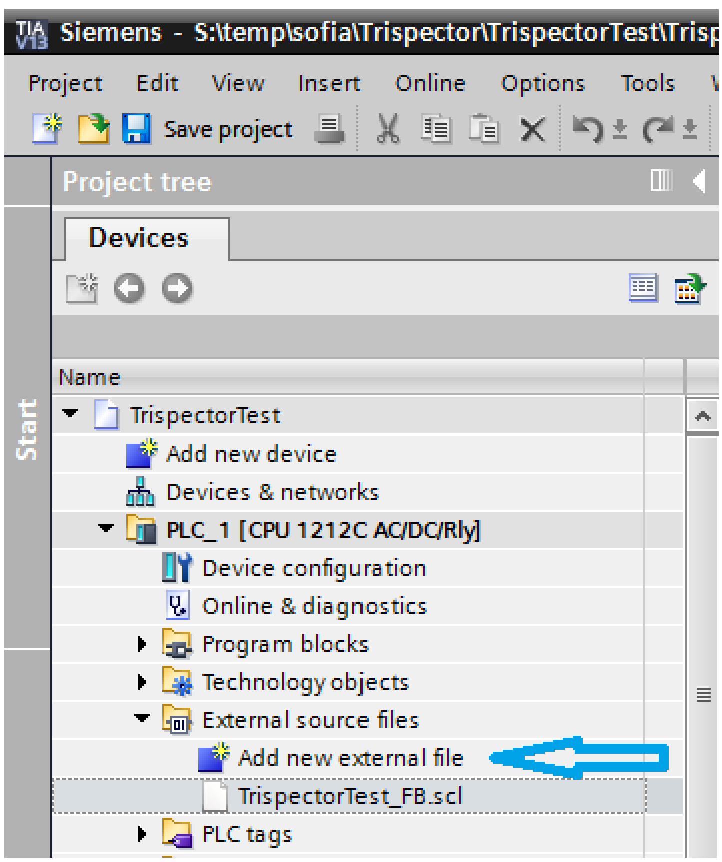


Figure 78: Adding an external file

To use the block, it is necessary to generate it from the source file. Right-click the source file and click **Generate blocks from source**, see figure 79.

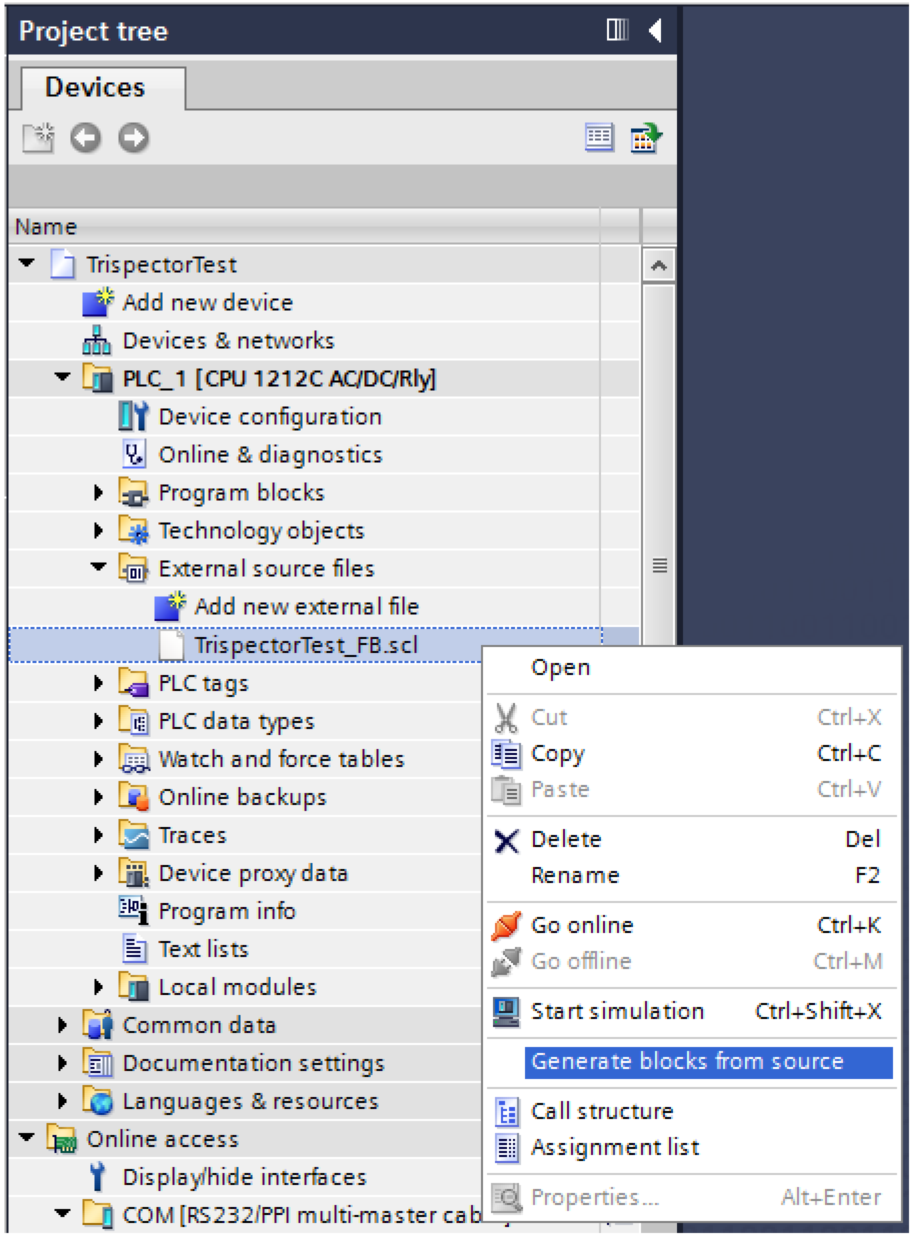


Figure 79: Generate blocks from source

Now the TIA Portal automatically generates the PLC blocks and adds it to the program. After a successful generation, the blocks are created in the **Program blocks** and **PLC data types** folders in the TIA project tree, see figure 80.

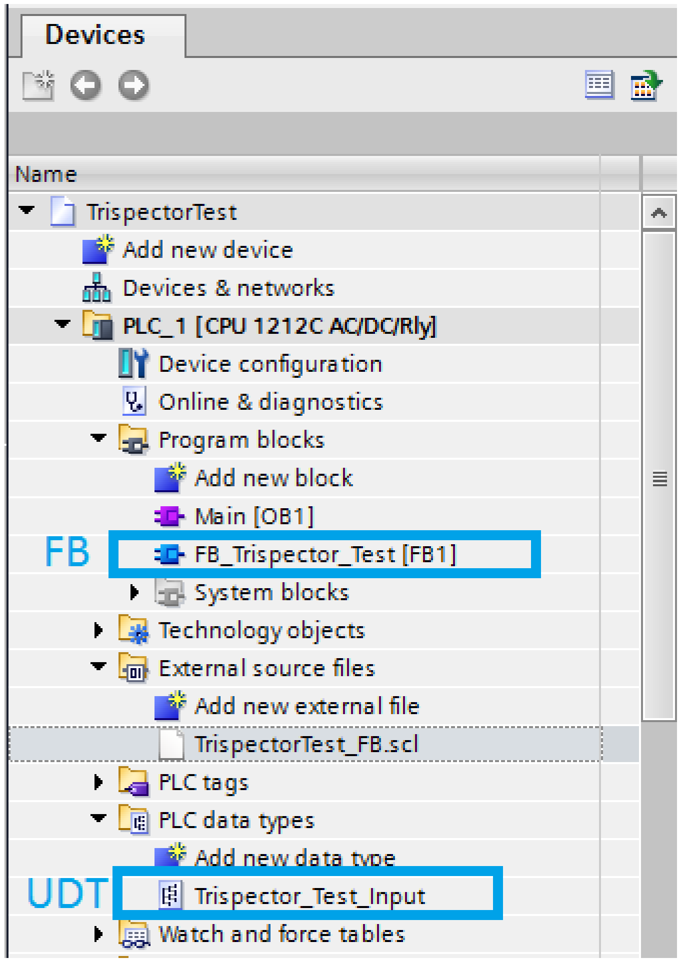


Figure 80: TIA project tree

NOTICE
If a PLC block with the same name as the generated block located in the project exists, the existing block will be overwritten.

12.8.4.3 Function block usage

The picture below shows the generated function block which can interpret an incoming string with format `string;real;int;char;` for example “`Invalid;1.004;28;A;`” from the TriSpector1000.

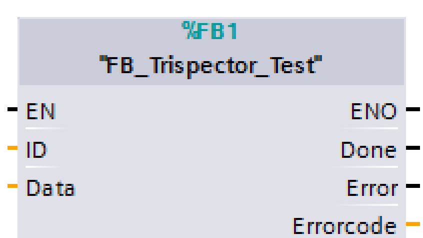


Figure 81: Generated function block

Function block parameters

Parameter	Declaration	Data type	Description
ID	Input	CONN_OUC	TCP Connection ID getting from the **TCON** Function Block
Data	Input	User Defined Data Type (UDT)	Reference to the generated data structure instance
Done	Output	BOOL	The rising edge indicates that new data is available FALSE: no new data available TRUE: new data available
Error	Output	BOOL	Error flag FALSE: no error detected TRUE: error detected
Errorcode	Output	DWORD	Error code, see error description in chapter 12.8.6.2

The **ID** parameter must be the same as the **ID** parameter of the TCON function block. The **Data** parameter needs a reference to the data structure instance. Create this reference by adding a data block that contains a variable of data type **Trispector_Test_Input**. Here, this block is called **DB_Trispector_Data** and contains a variable called **Trispector_Data**. The data block is created by double-clicking **Add new block** (see figure 82).

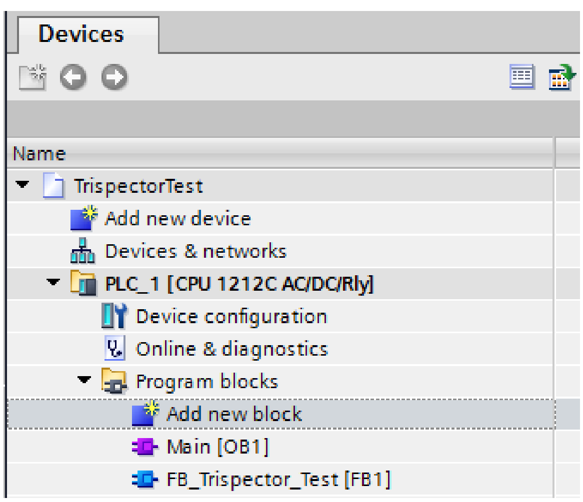


Figure 82: Adding a new data block

In the added block, create a variable named **Trispector_Data** that is of type **Trispector_Test_Input**, see figure 83.

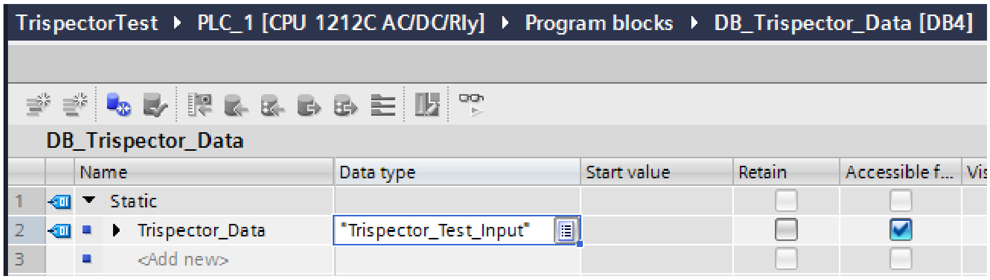


Figure 83: Creation of `TriSpector_Data` variable

Use the **Trispector_Data** variable as the **Data** input parameter to the function block. Now you should have two networks according to figure 84.

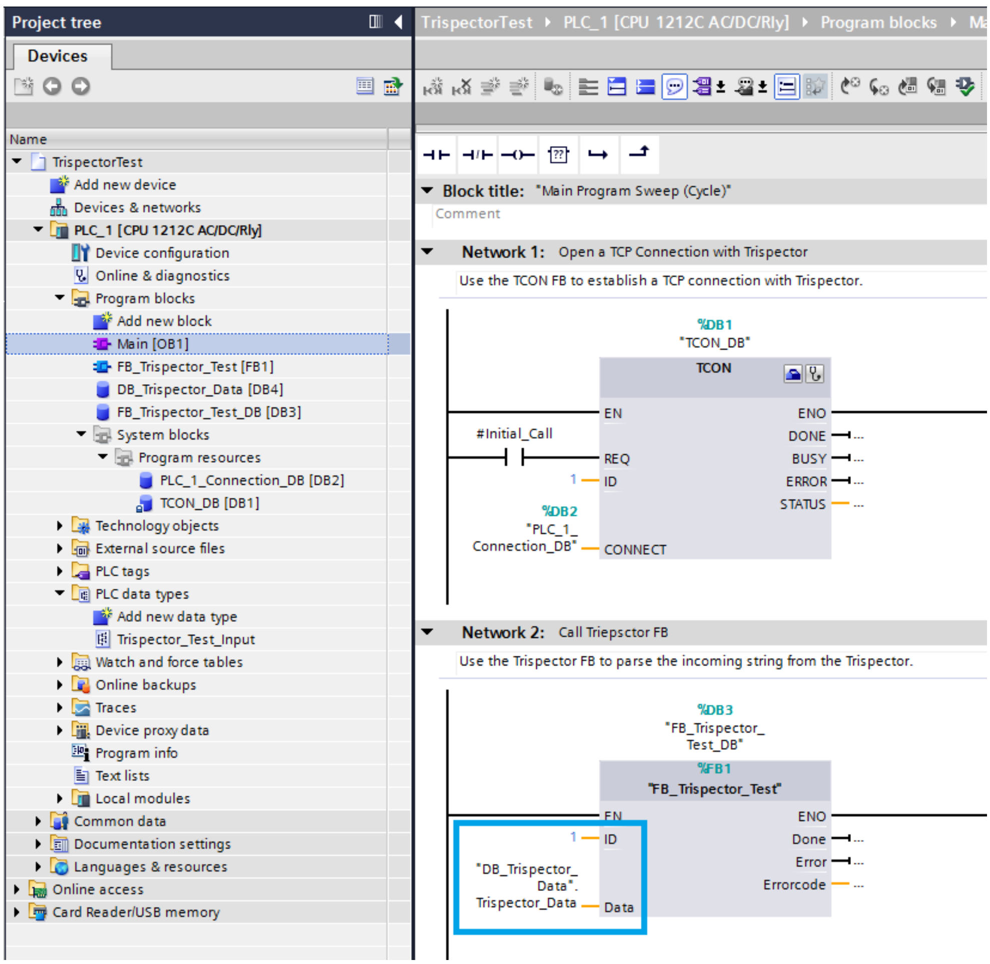


Figure 84: Networks

12.8.4.4 Observing the inspection results

When the configuration is activated and the PLC is in run mode, the inspection results can be observed in the '**DB**' instance of the generated structure, in this case **DB_Trispector_Data**. The structure is updated each time the TriSpector1000 sends new data (**Done** flag).

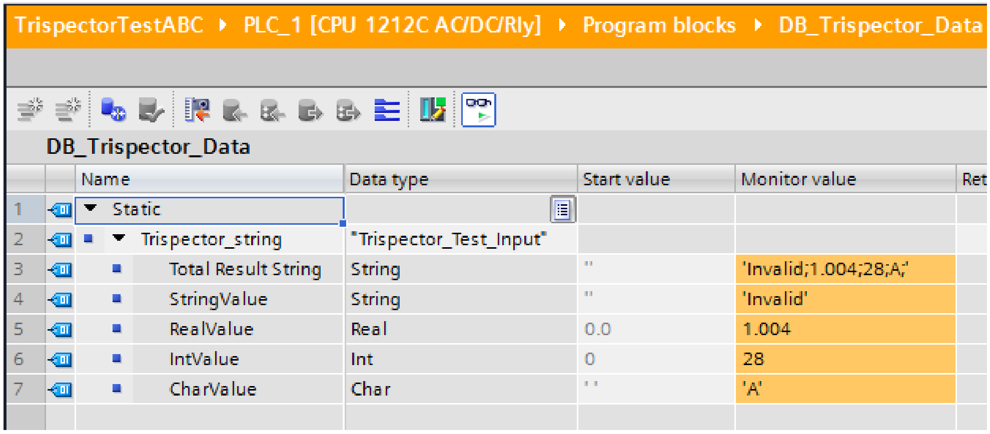


Figure 85: ***DB_TriSpector_Data*** *variable*

12.8.5 Modifying the Function Block

12.8.5.1 Modifying the example string

The example we have used is for parsing a string with four variables: `string;real;int;char;`. They do not need to have the values we have tested, try to modify the values but still keeping the data types and see that it is still working. For example:

Figure 86: Modified example string

The values appear in the **DB_Trispector_Data** structure (figure 87).

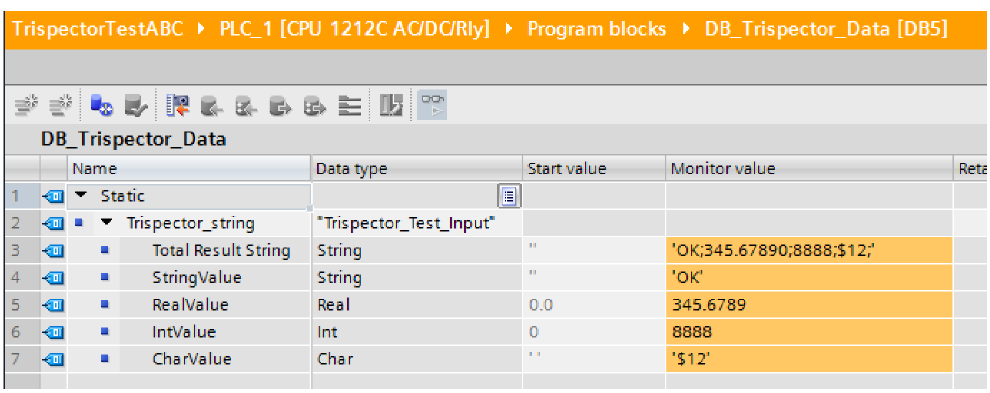


Figure 87: ***DB_TriSpector_Data*** *structure*

12.8.5.2 Using your desired output string

If you want to send some other data in your result output from the TriSpector1000, you must configure the function block and the data type (UDT) to fit your desired result. An example is presented below.

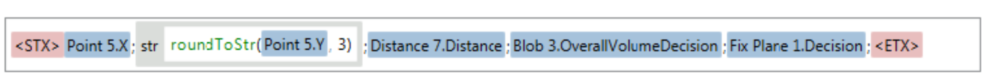


Figure 88: Output string with three real values and two strings

In the output string in figure 88 there are three real values followed by two string values. The UDT must be modified to match this. In the following examples, we will use the new output string to show how to modify your UDT and function block.

12.8.5.3 Modifying your UDT

The UDT in this example is **Trispector_Test_Input** ,which is found in the **PLC data types** folder (see figure 89). By right-clicking in the UDT view, you can insert rows and delete rows. Make sure that the UDT matches your current output. It is a good idea to keep the **Total Result String** variable for debugging purpose. If you want to remove it later to save processing time, see chapter 12.8.5.4.1.

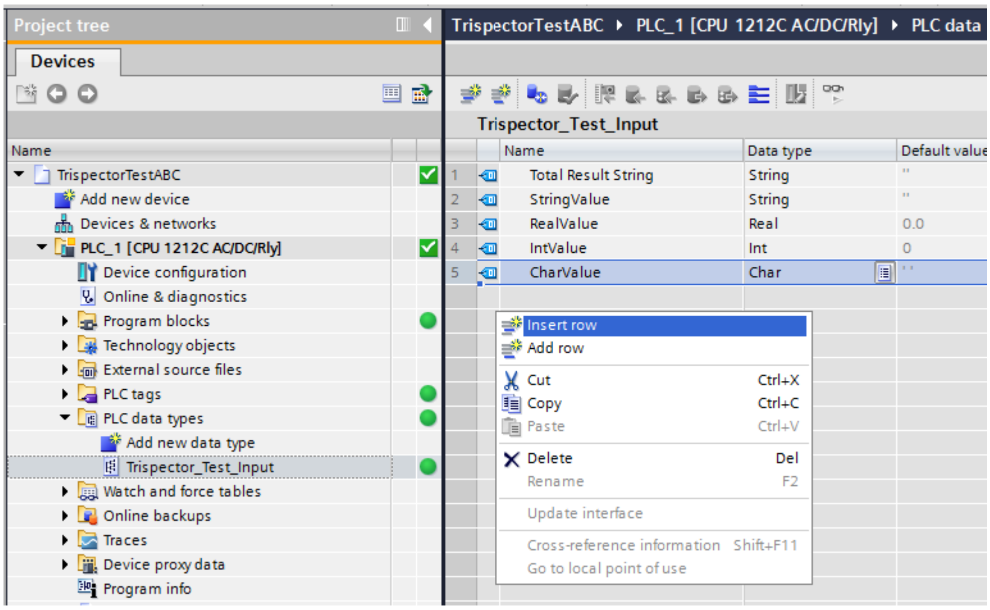


*Figure 89: **TriSpector_Test_input** data type*

If we use the new example string defined in section 6.2, the UDT should look like figure 90:

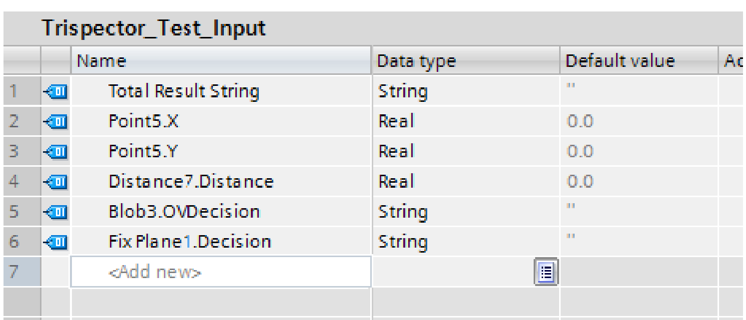


Figure 90: UDT for new example string

After changing the UDT, compile the data block using the UDT **DB_Trispector_Data**. Right-click on the data block **DB_Trispector_Data** and click **Compile** -> **Software (only changes)**, see figure 91.

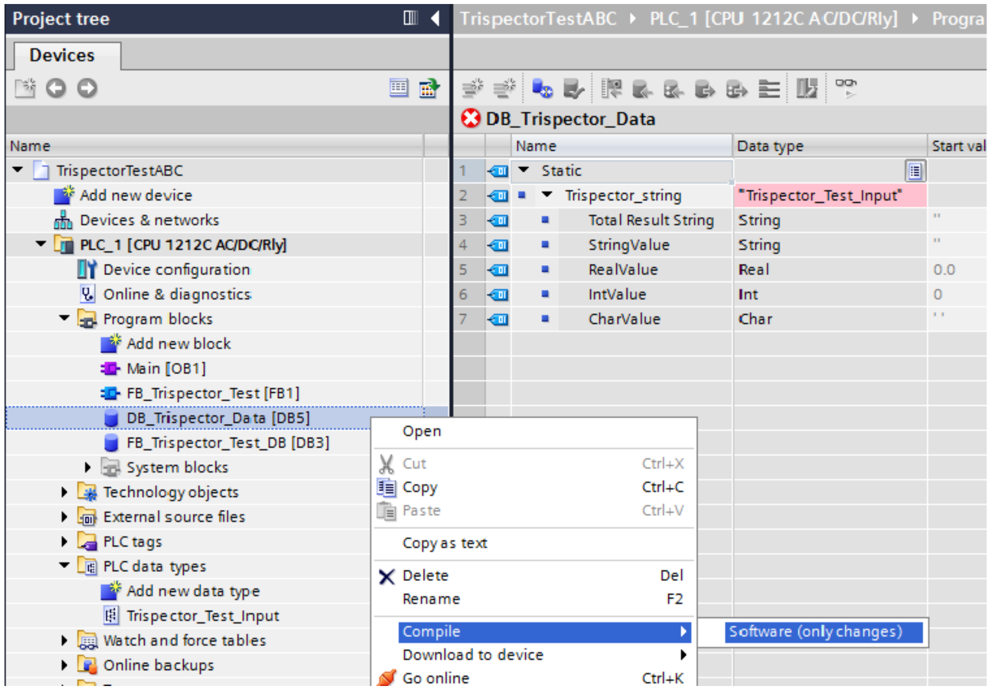


Figure 91: Compile data block

12.8.5.4 Modifying the Function Block

Secondly, modify the function block **FB_Trispector_Test** to parse the incoming data correctly. Open the function block and scroll down to the comment:

```
//********************START THE PARSING*********************
```

This is where the parsing of the string into different data types starts. Since the function block expects four values (`string;real;int;char`), the parsing is divided into four sections:

//1: String of unknown size

```
***********************************************************
WHILE (BYTE_TO_CHAR(#arrData[#i]) <> ';' & #i <>
(#iReceivedLength-1)) DO
#s_temp := CONCAT(IN1 := #s_temp, IN2 :=
CHAR_TO_STRING(BYTE_TO_CHAR(#arrData[#i])));
#i := #i + 1;
END_WHILE;
#Data.StringValue := #s_temp;
#s_temp := '';
#i := #i + 1;
//**********************************************************
```

//2: Real of unknown size

```
***********************************************************
WHILE (BYTE_TO_CHAR(#arrData[#i]) <> ';' & #i <>
(#iReceivedLength-1)) DO
#s_temp := CONCAT(IN1 := #s_temp, IN2 :=
CHAR_TO_STRING(BYTE_TO_CHAR(#arrData[#i])));
#i := #i + 1;
END_WHILE;
#Data.RealValue := STRING_TO_REAL(#s_temp);
#s_temp := '';
#i := #i + 1;
//**********************************************************
```

//3: Int of unknown size

```
************************************************************
WHILE (BYTE_TO_CHAR(#arrData[#i]) <> ';' & #i <>
(#iReceivedLength-1)) DO
#s_temp := CONCAT(IN1 := #s_temp, IN2 :=
CHAR_TO_STRING(BYTE_TO_CHAR(#arrData[#i])));
#i := #i + 1;
END_WHILE;
#Data.IntValue := STRING_TO_INT(#s_temp);
#s_temp := '';
#i := #i + 1;
//************************************************************
```

//4: Char of known size 1 byte

```
************************************************************
#Data.CharValue := BYTE_TO_CHAR(#arrData[#i]);
//************************************************************
```

These four sections are possible to copy and paste and combine in any way to match your output string, as long as the data types are strings, integers, real values or characters.

NOTE
If the output string contains an unknown number of variables, which depend on the number of found blobs for example, this approach will not work. You must rewrite the code block.

Since we now have a UDT with three real values followed by two string values, the code must be modified to fit this. For the real values, reuse section 2 (Real of unknown size). For the string values, reuse section 1 (String of unknown size).

Remove all the code above and insert three 'section 2' snippets followed by two 'section 1' snippets. Finally, assign the parsed values to the correct name in the UDT since this has changed (see the **Data** variables in the code below). The result should look like this:

//1: Real of unknown size

```
************************************************************
WHILE (BYTE_TO_CHAR(#arrData[#i]) <> ';' & #i <>
(#iReceivedLength-1)) DO
#s_temp := CONCAT(IN1 := #s_temp, IN2 :=
CHAR_TO_STRING(BYTE_TO_CHAR(#arrData[#i])));
#i := #i + 1;
END_WHILE;
#Data."Point5.X" := STRING_TO_REAL(#s_temp);
#s_temp := '';
#i := #i + 1;
//************************************************************
```

//2: Real of unknown size

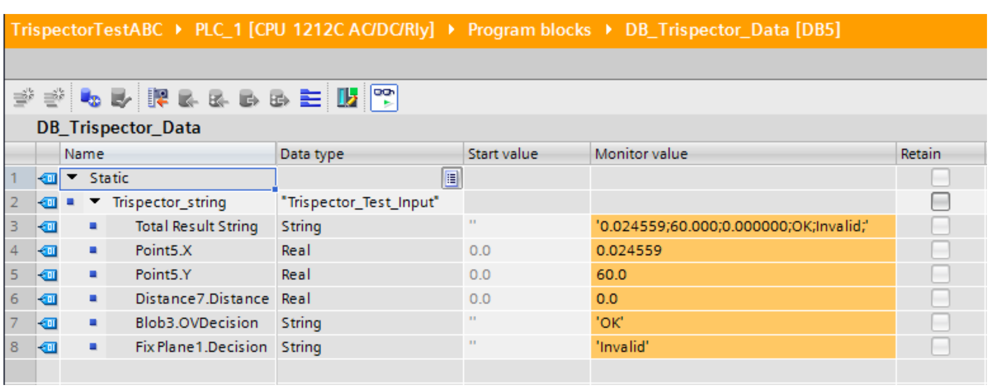
```
************************************************************
WHILE (BYTE_TO_CHAR(#arrData[#i]) <> ';' & #i <>
(#iReceivedLength-1)) DO
#s_temp := CONCAT(IN1 := #s_temp, IN2 :=
CHAR_TO_STRING(BYTE_TO_CHAR(#arrData[#i])));
#i := #i + 1;
END_WHILE;
#Data."Point5.Y" := STRING_TO_REAL(#s_temp);
#s_temp := '';
#i := #i + 1;
//************************************************************
```

```
//3: Real of unknown size
**********************************************************
WHILE (BYTE_TO_CHAR(#arrData[#i]) <> ';' & #i <>
(#iReceivedLength-1)) DO
#s_temp := CONCAT(IN1 := #s_temp, IN2 :=
CHAR_TO_STRING(BYTE_TO_CHAR(#arrData[#i])));
#i := #i + 1;
END_WHILE;
#Data."Distance7.Distance" := STRING_TO_REAL(#s_temp);
#s_temp := '';
#i := #i + 1;
//**********************************************************
```

```
//4: String of unknown size
**********************************************************
WHILE (BYTE_TO_CHAR(#arrData[#i]) <> ';' & #i <>
(#iReceivedLength-1)) DO
#s_temp := CONCAT(IN1 := #s_temp, IN2 :=
CHAR_TO_STRING(BYTE_TO_CHAR(#arrData[#i])));
#i := #i + 1;
END_WHILE;
#Data."Blob3.OVDecision" := #s_temp;
 #s_temp := '';
#i := #i + 1;
//**********************************************************
```

```
//5: String of unknown size
**********************************************************
WHILE (BYTE_TO_CHAR(#arrData[#i]) <> ';' & #i <>
(#iReceivedLength-1)) DO
#s_temp := CONCAT(IN1 := #s_temp, IN2 :=
CHAR_TO_STRING(BYTE_TO_CHAR(#arrData[#i])));
#i := #i + 1;
END_WHILE;
#Data."Fix Plane1.Decision" := #s_temp;
 #s_temp := '';
#i := #i + 1;
//**********************************************************
```

Now compile and have a look at the inspection result (figure 92).


TrispectorTestABC ▸ PLC_1 [CPU 1212C AC/DC/Rly] ▸ Program blocks ▸ DB_Trispector_Data [DB5]

DB_Trispector_Data

	Name	Data type	Start value	Monitor value	Retain
1	▼ Static				
2	▼ Trispector_string	"Trispector_Test_Input"			
3	Total Result String	String	''	'0.024559;60.000;0.000000;OK;Invalid;'	
4	Point5.X	Real	0.0	0.024559	
5	Point5.Y	Real	0.0	60.0	
6	Distance7.Distance	Real	0.0	0.0	
7	Blob3.OVDecision	String	''	'OK'	
8	Fix Plane1.Decision	String	''	'Invalid'	

Figure 92: New inspection result

12.8.5.4.1 Removing the Total Result String

If you do not want to keep the **Total Result String** variable, which is mostly used for debugging purposes, you can simply remove it. Remove it from your UDT and find the code snippet below in the function block and delete this snippet. The snippet loops through the whole string, waiting for the `<ETX>` character that determines the end of the string. Since the string is looped through once again to parse all the variables it is a good idea to remove this snippet if the time is critical.

```
//While we don't have <ETX>, save bytes to result string
//Saving the total result string is mostly for testing, this could
be removed to save time!
WHILE (BYTE_TO_CHAR(#arrData[#i]) <> '$03') DO
//Save every byte to the total result string
#s_total := CONCAT(IN1 := #s_total, IN2 :=
CHAR_TO_STRING(BYTE_TO_CHAR(#arrData[#i])));
#i := #i + 1;
END_WHILE;
#Data."Total Result String" := #s_total
//Start over and loop through incoming data again to cast every
single data type sent
#i := 6;
```

12.8.5.5 Increasing the data array size

In the current function block, the data received from the TriSpector1000 is saved in an array of 100 bytes. If the string sent from the TriSpector1000 contains more than 100 bytes (characters), you must extend this array. This is done by editing the data type in the function block **FB_Trispector_Test**, see figure 93.

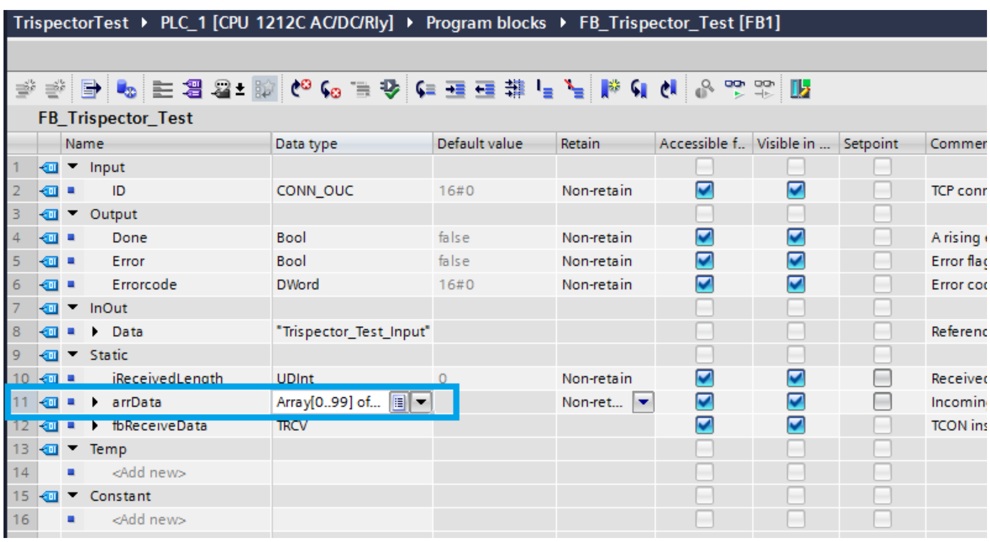


Figure 93: Editing the data type

After changing the size of the array, compile the data block corresponding to this function block to update it. Right-click the data block **FB_Trispector_Test_DB** and click **Compile** -> **Software (only changes)**, see figure 94

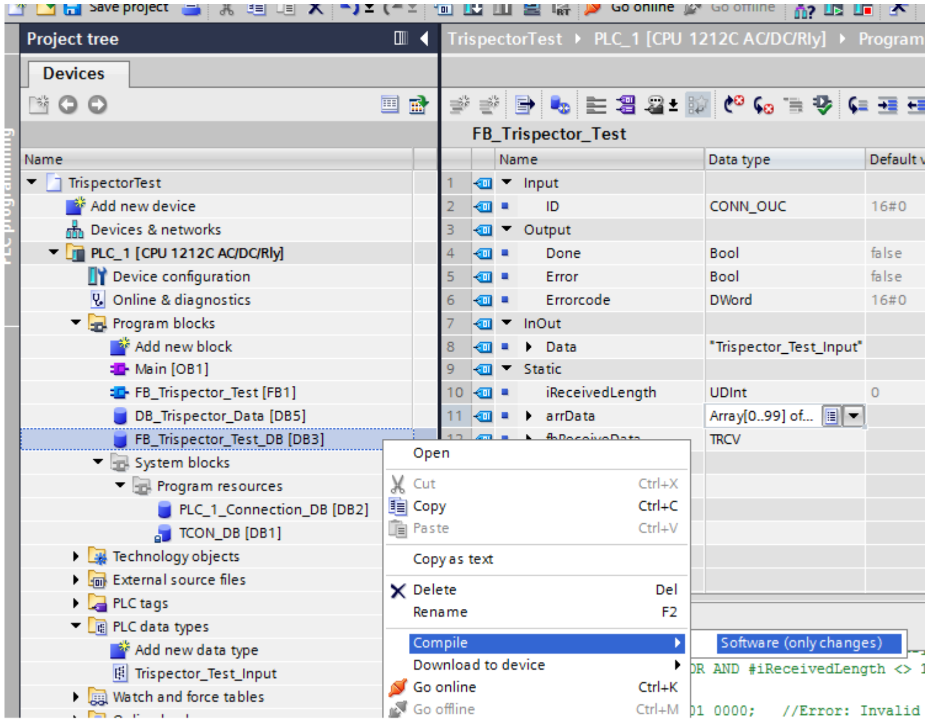


Figure 94: Compiling the data block

12.8.6 Troubleshooting

12.8.6.1 No data appears

If no data appears, there are two common reasons:

- The string you are sending is too long. If the string is more than 100 characters long, increase the data array size according to "Increasing the data array size", page 118.
- The <STX> and the <ETX> characters at the start and end of the string are missing. They are needed for the PLC to retrieve the string.

If the error remains, see "Error codes", page 119.

12.8.6.2 Error codes

The monitored values in **FB_Trispector_Test_DB** include an **Errorcode** parameter (see figure 95) for identification of errors. The **Errorcode** parameter contains error information generated by the Siemens **TRCV** function block that is forwarded through the TriSpector1000 Result Output function block.

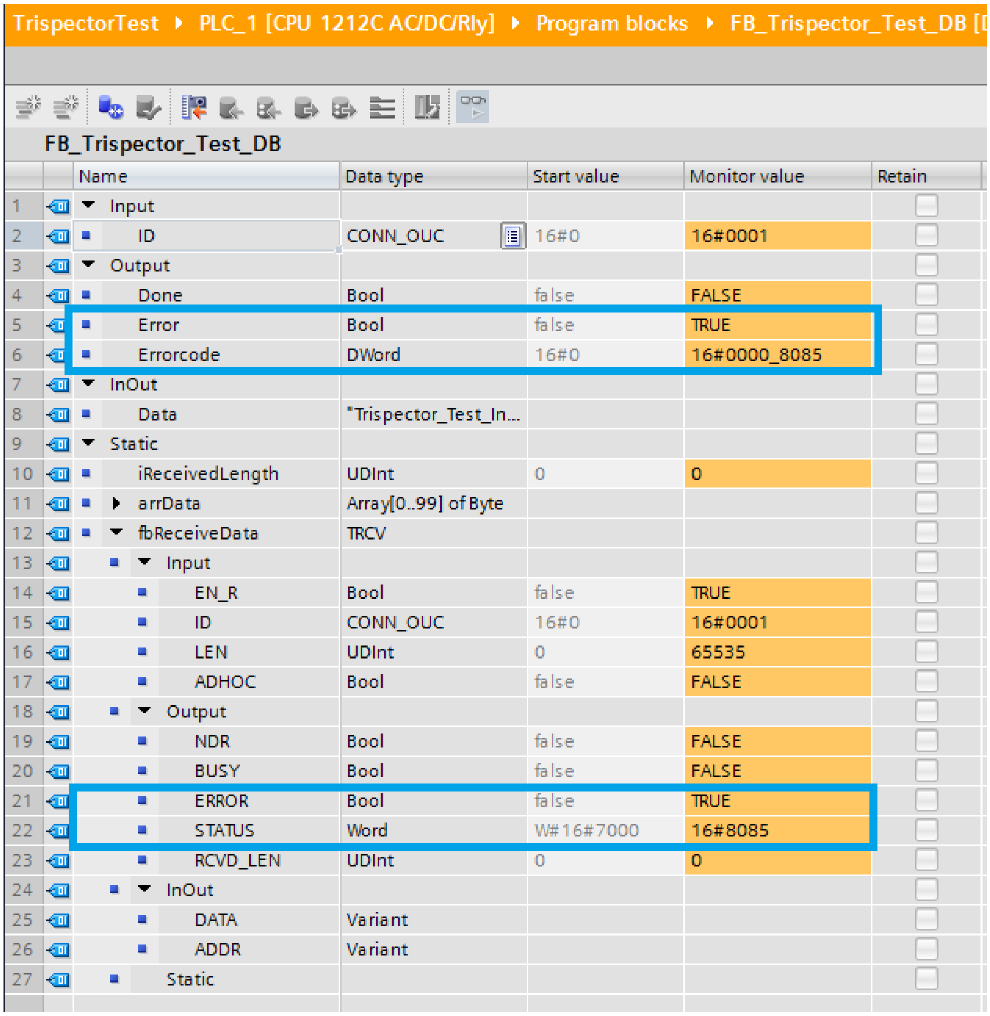


*Figure 95: **Errorcode** and related parameters*

See table 30 for a list of error codes of the TriSpector1000 Result Output function block. For interpretation of the **TRCV** error codes, refer to the **TCON** help in the Siemens help system.

Table 30: Error codes

Error code	Short description	Description
16#0000_0000	No Error	No Error.
16#0001_0000	Invalid telegram length (<STX> or <ETX> is missing)	Data sent from the Trispector contains too many bytes. Default is maximum 100 bytes, if sending more data the array size needs to be increased, see chapter 6.5. Could also be that <STX> or <ETX> char is missing in start/end of string.
16#0000_XXXX	**TRCV** Error	Use the TIA Portal help system to interpret the error code.

In figure 95, the error code 8085 comes from the **TRCV** function block. According to the TIA Portal help system, the trouble is due to the **LEN** or **DATA** parameter in the **TRCV** function block. This problem typically appears if you have used the wrong function block. The same yields for error 8088. Look in the table in "Supported PLCs", page 104 to see which function block to choose.

12.8.6.3 Parsing goes wrong

If there is data in the **Total Result String** variable but nothing in the other variables, the data is coming through but something goes wrong in the parsing of the data. The delimiters in the string might be missing; you must separate the variables with a semicolon.

12.8.7 Function block differences

As mentioned in chapter 12.8.1.3, this document is applicable for three different function blocks. The difference lay in how the ad-hoc functionality in the **TRCV** function block is handled. The **TRCV** function block is used by the TriSpector1000 Output Result function block to receive the data from the TCP/IP connection. The data is only available when the data length specified at the **LEN** parameter has been completely received. For the data to be available immediately, even if the data length specified at the **LEN** parameter has not yet been completely received, you must use the ad-hoc mode. How to enable this ad-hoc mode differs from the different PLC versions. In the latest version, S7-1500 and S7-1200 with firmware 4 and higher, there is a specific ad-hoc parameter. In earlier versions, you use the **LEN** parameter to enable the ad-hoc mode. The following table shows how to enable the ad-hoc mode in the different versions.

Simatic S7 version	TRCV values
S7-1500 and S7-1200 firmware version 4.x	**ADHOC** = TRUE, **LEN** = 0
S7-1200 firmware version > 2 and < 4	**LEN** = 65535
S7-300, S7-400 and S7-1200 version 1.x	**LEN** = 0

When using S7-1200, the selection of function block depends on the firmware version. It is not the firmware version of the PLC but the firmware version of the TRCV FB as could be seen in image below.

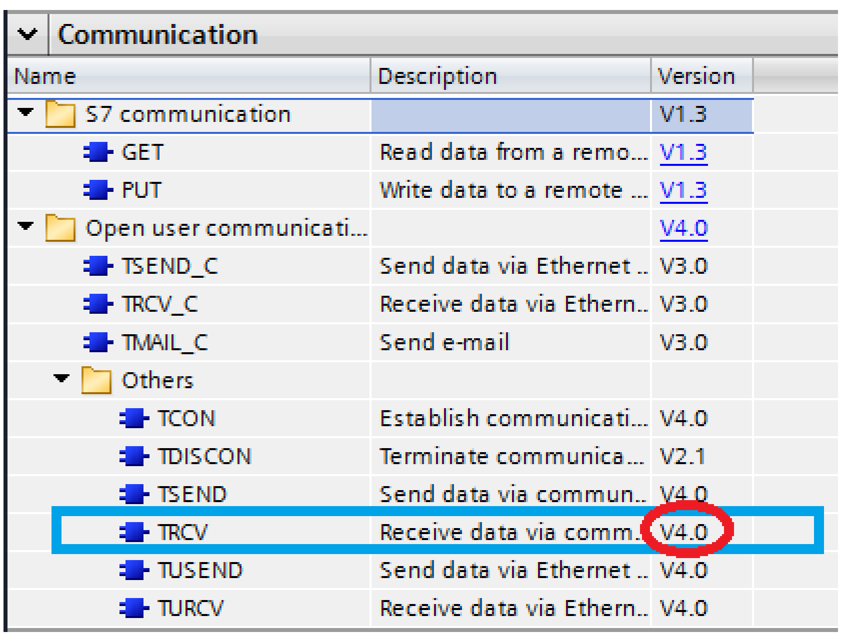


Figure 96: Firmware version of TRCV function block

8021808/199P/2020-09/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 2 57 91 18 50
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 01
E-Mail info@sick.de

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 +972-4-6881000
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 mario.garcia@sick.com

Netherlands
Phone +31 (0) 30 229 25 44
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

Russia
Phone +7 495 283 09 90
E-Mail info@sick.ru

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 (0)11 472 3733
E-Mail info@sickautomation.co.za

South Korea
Phone +82 2 786 6321
E-Mail info@sickkorea.net

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 info@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

Further locations at **www.sick.com**

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

SICK
Sensor Intelligence.