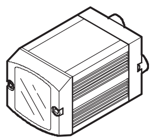
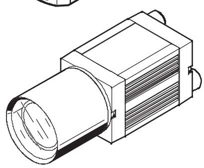
Compact Vision System

Manual
Electronics

Manual
Compact Vision
System SBO...-Q





Manual
548319
en 1103d
[757656]

Original .... de

Edition .... en 1103d

Designation .... P.BE-SBO-Q-EN

Order no. 548319

Internet: http://www.festo.com
E-mail: service_international@festo.com

Table of contents

Intended use VI
Range of application and certification VII
Safety instructions VIII
Service IX
Target group IX
Important user instructions X
About this manual XII
Product-specific terms and abbreviations XIII

1. System overview 1-1
1.1 Design of the SBO Compact Vision System 1-3
1.2 Variants 1-5
1.2.1 Mode of operation 1-8
1.2.2 Display and connecting elements 1-9
1.3 Software packages 1-12
1.4 Accessories 1-13
1.5 Selecting a lens for type SBOC 1-14
1.6 Selecting lighting 1-18

2. Mounting 2-1
2.1 Mounting 2-3
2.1.1 Mounting the Compact Vision System 2-4
2.1.2 Dimensions of SBO Compact Vision Systems 2-5
2.1.3 Mounting with adapter kit type SBOA-HMSV-39 2-6
2.1.4 Mounting/removing the lens and shield tube on type SBOC 2-7
2.1.5 Removing the protective foil on type SBOI 2-8

3. Installation 3-1
3.1 General instructions on installation 3-3
3.1.1 Selecting a power supply unit 3-5
3.2 Electrical connections 3-6
3.2.1 Connecting the operating voltage supply and I/Os 3-6
3.2.2 Connecting the Ethernet interface 3-11
3.2.3 Connecting the CAN interface 3-15

4.	**Commissioning**		**4-1**
4.1	Notes on commissioning		4-3
4.2	Installing the software packages		4-4
4.3	Network settings on the PC		4-5
4.4	Firewall settings on the PC		4-8
4.5	Network settings on the Compact Vision System		4-11
4.6	CheckKon connection to the Compact Vision System		4-14
4.7	Settings on the Compact Vision System		4-16
4.8	System parameters for preprocessing		4-19
4.9	Selection of the evaluation mode		4-21
	4.9.1	I/O process in “Triggered” evaluation mode	4-22
	4.9.2	I/O process with the “Free run” evaluation mode	4-28
	4.9.3	I/O sequence with the “Fixed frame rate” evaluation mode	4-39
4.10	Connection to higher-level controller (PLC/IPC)		4-44
	4.10.1	General information on use of inputs	4-46
	4.10.2	General information on use of outputs	4-50
	4.10.3	Use of internal I/Os	4-50
	4.10.4	Use of the I/O expansion	4-52
	4.10.5	Use of the device as CPI module at CP nodes	4-57
	4.10.6	Use of the device as CANopen Master	4-63
	4.10.7	Use of the Ethernet interface with EasyIP	4-65
	4.10.8	Use of the Ethernet interface with Telnet	4-66
	4.10.9	Use of the Ethernet interface with Telnet streaming	4-77
	4.10.10	Use of the Ethernet interface with Telnet XML	4-79
	4.10.11	Use of the Ethernet interface with Modbus	4-90
	4.10.12	Use of the Ethernet interface with EasyIP	4-92
	4.10.13	Use of CoDeSys	4-101
	4.10.14	Display of inspection results with the “SBO...-Q WebViewer”	4-107
4.11	Creation of the check programs		4-110
4.12	Checking the system settings		4-111
4.13	Instructions on operation		4-112

5. Diagnostics and error handling 5-1
5.1 General diagnostic options 5-3
5.1.1 Status display 5-3
5.1.2 Error handling 5-6

A. Technical appendix A-1
A.1 Cleaning and care A-3
A.2 Addressing in the Ethernet (basics) A-4
A.3 Siemens star A-7
A.4 Technical data A-8
A.5 Error messages A-11
A.6 Address table for EasyIP, Modbus, Telnet and CoDeSys embedded A-14
A.6.1 Input register A-14
A.6.2 Output register A-15
A.6.3 Rapid access to the input and output registers A-16
A.6.4 Further system status/system information A-17
A.6.5 System time of the device A-18
A.6.6 Total tolerance of the type in the current check program A-18
A.6.7 Basic results of the last check A-19
A.6.8 Free flag words (non-remanent) A-21
A.6.9 Features – results of the last check A-21
A.6.10 System parameters A-24
A.6.11 String address table A-25
A.7 Address table for EtherNet/IP A-26
A.7.1 Address table for EtherNet/IP – protocol-specific objects A-26
A.7.2 Address table for EtherNet/IP – SBO-specific objects A-30
A.8 Data types A-42
A.9 Programming robot controllers A-44
A.9.1 Telnet communication with an ABB robot A-44
A.9.2 Telnet communication with a KUKA robot using XML A-53

B. Index B-1

Intended use

The SBO...-Q Compact Vision System has been designed for installation in a machine or automated system. It is used to analyse the quality and position of parts.

The SBO...-Q Compact Vision System may only be used as follows:

– According to its intended use.

– In its original state without unauthorised alterations. Only conversions and modifications described in the documentation supplied with the product are permitted.

– In perfect technical condition.

When the product is connected to commercially available components such as sensors and actuators, the specified limits for pressures, temperatures, electrical data, torques etc. must be observed. National and local safety regulations must also be observed.

The device is intended for use in an industrial environment. If it is used in residential buildings, interference suppression measures may need to be taken.

Range of application and certification

The product fulfils the requirements of the applicable EU directives and bears the CE mark.

Standards and test values which the product adheres to and fulfils can be found in the section "Technical data". Details of the EU directives relevant to the product can be found in the declaration of conformity.

Certain product configurations have been certified by Underwriters Laboratories Inc. (UL) for the USA and Canada. These configurations are marked as follows:

UL Recognized Component Mark for Canada and the United States

Safety instructions

Caution
During commissioning and programming of the device, the safety requirements as per this manual and as per the documentation for the controller and the other components used must always be observed.

Users must ensure that nobody has access to the positioning range of the connected actuators. Access to the possible danger area must be prevented by suitable measures such as protective screens and warning signs.

Caution
Electrostatically sensitive components! Electrostatic discharges can damage the internal electronics.

- Do not open the housing. Observe the handling specifications for electrostatically sensitive devices.

Caution
A dirty and scratched lens or dirty and scratched protective glass can lead to optical errors. Make sure that the lens/protective glass is not scratched. Do not use any abrasive cleaning agents.

Clean the lens/protective glass in case of dirt or other deposits:

– using a blower brush or clean, non-lubricated compressed air

– using a soft, moist cloth and a non-abrasive cleaning agent

Caution
If the permitted temperature range is exceeded, e.g. due to strong external light sources, this can lead to system errors and cause damage.

- Mount the Compact Vision System in a well ventilated location, screening it from the heat emitted by other devices and light sources.

Service

Please consult your local Festo repair service if you have any technical problems.

Target group

This manual is intended exclusively for technicians trained in control and automation technology who have experience in installing and commissioning electronic systems.

Important user instructions

Danger categories

This manual provides information on dangers that can arise if the product is not used correctly. The information is presented under headings such as Warning, Caution, etc., appears on a shaded background, and is additionally marked with a pictogram. A distinction is made between the following danger warnings:

Warning
... means that failure to observe this instruction may result in serious personal injury or damage to property.

Caution
... means that failure to observe this instruction may result in personal injury or damage to property.

Note
... means that failure to observe the instruction may result in damage to property.

The following pictogram denotes passages in the text which describe activities involving electrostatically sensitive devices:

Electrostatically sensitive devices: Incorrect handling can result in damage to components.

Marking of special information

The following pictograms designate texts that contain special information.

Pictograms

Information
Recommendations, tips and references to other sources of information

Accessories:
Specifications about necessary or useful accessories for the Festo product.

Environment:
Information on the environmentally friendly use of Festo products.

Text designations

- Bullet points indicate activities that may be carried out in any order.

1. Numerals denote activities which must be carried out in the numerical order specified.

– Hyphens indicate general lists.

About this manual

Note
This manual is for the following versions:

Hardware/software	Version
Compact Vision System SBO...-Q	From software version 3.5 From hardware version CA0508
CheckKon	Version 4.2 or higher
CheckOpti	Version 3.1 or higher
SBO-DeviceManager	Version 1.3 or higher

Tab. 0/1: Hardware and software versions

This manual contains general basic information on mounting, installation and operation of the Compact Vision System. Additional information on commissioning, parametrisation and diagnostics using the software packages can be found in the packages' Help system.

Type	Title	Contents
Electronics manual	Manual Compact Vision System SBO...-Q P.BE-SBO-Q-... (i.e. this manual)	Mounting, installation and commissioning of the Compact Vision System
Help system	Help for CheckKon P.SW-KON	CheckKon functional description and operating instructions
Help system	Help for CheckOpti P.SW-OPTI	CheckOpti functional description and operating instructions
Manual	SBO-DeviceManager Help	SBO-DeviceManager functional description and operating instructions

Tab. 0/2: Compact Vision System documentation

Product-specific terms and abbreviations

The following product-specific abbreviations are used in this manual:

Term/abbreviation	Meaning
Amplification	➔ Sensor amplification
Aperture	Opening through which light passes, via a lens before it hits the sensor. The larger the opening or aperture, the more light will reach the sensor. When the aperture opens, the depth of focus is reduced. The depth of focus increases as the aperture closes (larger f-number). Small apertures make longer exposure times necessary. Short exposure times require larger apertures.
Auto MDI-X	Recognises the Ethernet configuration of the other station and adapts the send and receive cables of a network connection automatically.
CANopen	Fieldbus protocol based on CAN, which is a European standard.
Check program	Definition of parts to be recognised and features to be determined.
CheckKon	Software package for configuration and commissioning.
CheckOpti	Software package for creating check programs.
CMOS sensor	Optoelectronic sensor which converts light signals into electrical signals. In addition to the sensor function, there are also image processing functions for e.g. exposure control and contrast correction integrated directly into the chip.
CoDeSys pbF	CoDeSys provided by Festo CoDeSys = Controller Development System CoDeSys provided by Festo permits the configuration, commissioning and programming of various Festo components and devices.
Condition-controlled	Recognition of a signal (e.g. input) reacts to a logic 1 or logic 0.
CP connection	Socket/plug on the CP modules which allows connection of the modules via the CP cable.
CP cable	Special cable used to connect the various CP modules in a CP string.

Term/abbreviation	Meaning
CP node (CP master)	Collective term for modules with one or more CP connections, to each of which one CP string can be connected. CP nodes include, for example, the CPX-CP interface, CP field bus nodes, and valve terminals with CP string extension.
CP modules	Collective term for the various modules that can be integrated in a CP system (CP functionality).
CP string	Entirety of the CP modules and CP cables connected together to a CP master's CP connection.
CP system / CPI system	Complete electrical installation system consisting of a CP master with one or more CP strings.
CP valve terminal	Type 10 CPV valve terminal or type 12 CPA valve terminal, in each case with a CP connection (also regarded as CP modules).
CPI modules	CP modules with CPI functionality (extended functionality)
CPX modules	Collective term for the various modules which can be integrated into a CPX terminal.
CPX terminal	Complete system consisting of CPX modules with or without pneumatics.
Data output	Output of selected check results and feature values for communication with controllers or robots, for graphical display or for production data acquisition. Setup of data output is performed in check programs using the software CheckOpti (version 3.1 or higher).
Depth of focus	The spatial area in front of and behind the focussed object which is still sharp (also known as focal depth). The depth of focus depends on the focal length of the lens and the aperture set. Small focal lengths with small aperture openings lead to greater depth of focus.
EasyIP	Protocol for the simple exchange of operands between Festo controllers (e.g. FEC Standard, PS1, etc.). EasyIP controllers are normally both client and server. But there can also be controllers without server function, such as diagnostic devices or visualisation computers, that participate in EasyIP.
Edge-controlled	Recognition of a signal (e.g. input) reacts to rising or falling edge.
Ethernet	Physical protocol and network for connecting various devices.
EtherNet/IP	Communication standard via TCP/IP in automation technology.

Term/abbreviation	Meaning
Exposure time	The duration for which the CMOS sensor is subjected to light during image capturing. The longer the exposure time, the more light will penetrate. The exposure time is chosen on the basis of e.g. speed of travel, amount of light available, and the light sensitivity of the sensor (➔ Sensor amplification). In the case of moving objects, long exposure times result in blurred images.
Falling edge	Transition from logic 1 to logic 0 (falling).
Feature(s)	Defined values (e.g. length) that are determined by the check program and used for analysis.
Fieldbus nodes	Provide the connection to specific fieldbuses. Transmit control signals to the connected modules and monitor their ability to function.
Focal depth	➔ Depth of focus
Focal length	Large focal lengths create a large image, while small focal lengths create a wide-angle image. Lenses with variable focal lengths are known as zoom lenses. Lenses with greater focal lengths usually have a smaller depth of focus and a lower luminous intensity.
I	Digital input.
I module	Input module.
I/O modules	Collective term for the modules which provide digital inputs and outputs (e.g. CPX I/O modules, CP input modules and CP output modules).
I/Os	Digital inputs and outputs.
Logic 0	Input or output supplies 0 V.
Logic 1	Input or output supplies 24 V.
Modbus TCP	Communication standard via TCP/IP in automation technology.
O	Digital output.
O module	Output module.
PLC/IPC	Programmable logic controller/industrial PC
Rising edge	Transition from logic 0 to logic 1 (rising).
SBO-DeviceManager	Software package for adjusting the network properties and firmware.

Term/abbreviation	Meaning
Sensor amplification	Influences the light sensitivity of the sensor. Increasing the amplification increases the light sensitivity. Excessive amplification can lead to grainy images.
String allocation	Type and order of the CP modules connected to one or more CP strings.
TCP/IP	Combination of the protocols TCP and IP, the most-widely used protocol in communication via Ethernet.
Telnet	Client-server protocol for general, bidirectional communication, using TCP. Telnet is normally used to offer users access to Internet computers via the command line.
Tools	Tools are used to evaluate images from the Compact Vision System in order to determine feature values. The tools are set up in check programs using the software CheckOpti.
TSP	Target Support Package A Target Support Package contains all of the configuration and expansion files that are needed to make a specific controller (here target system = target) available for the programming environment “CoDeSys provided by Festo”.
XML	Extensible Markup Language XML is a markup language for presenting hierarchically structured data in the form of text data.

Tab. 0/3: Product-specific terms and abbreviations

System overview

Chapter 1

Table of contents

1. System overview 1-1

1.1 Design of the SBO Compact Vision System 1-3
1.2 Variants 1-5
1.2.1 Mode of operation 1-8
1.2.2 Display and connecting elements 1-9
1.3 Software packages 1-12
1.4 Accessories 1-13
1.5 Selecting a lens for type SBOC 1-14
1.6 Selecting lighting 1-18

1.1 Design of the SBO Compact Vision System

Components

The SBO Compact Vision System is an intelligent camera with integrated electronics for image processing and communication. It is contained in a compact and robust housing and offers:

- an imaging CMOS sensor, with various resolutions in colour or monochrome, according to the model
- interfaces for communication and for connecting external devices
- SBOI:
  an integrated lens and integrated LED lighting
- SBOC:
  a standardised CS-Mount lens adapter, which is also usable as a C-Mount lens adapter if protective barrel is used for the lens.
  Appropriate lenses and additional optical elements, such as filters and lenses, are available on request.

Function

The SBO...-Q Compact Vision System allows optical check functions to be cost-effectively integrated into machines and systems in order to allow the quality and position of parts to be analysed. In addition, the system can perform control functions via the integrated CoDeSys PLC.

Configuration, commissioning and operation of the inspection functions of the SBO...-Q Compact Vision System are performed using the software packages CheckKon, CheckOpti and SBO-DeviceManager.

Configuration, commissioning and operation of the integrated CoDeSys PLC run-time system is performed using the software package “CoDeSys provided by Festo”.

User-specific firmware versions can be installed on the device for special applications.

Networking and control

The Compact Vision System can be linked directly to the PC via the Ethernet interface. Analyses can be controlled via digital I/Os or a PC. Further information can be found in chapter 3.2.2 on page 3-14.

1 Compact Vision System (pictured: SBOC)

2 PC with software packages

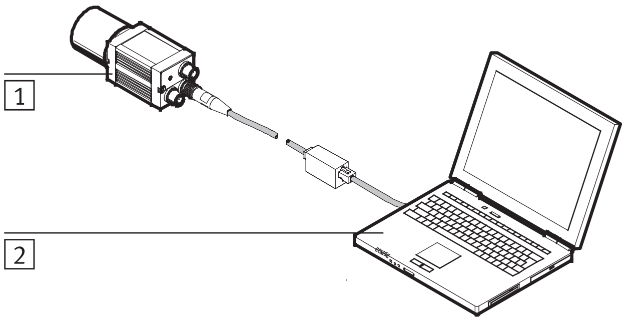
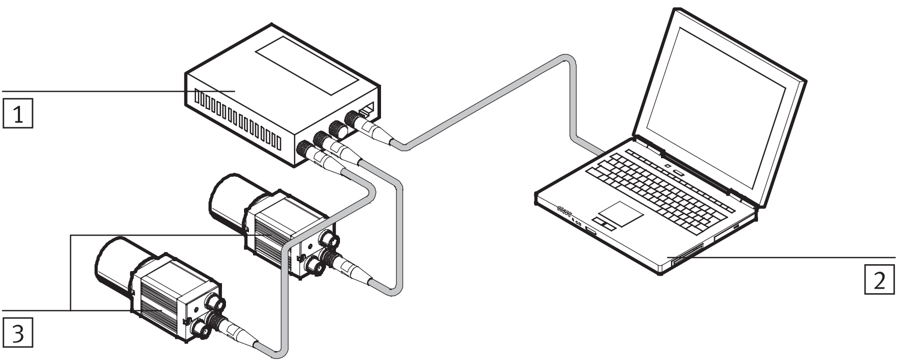


Fig. 1/1: Direct networking with the PC



1 Hub/switch

2 PC with software packages

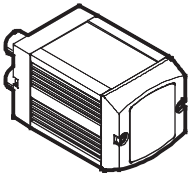
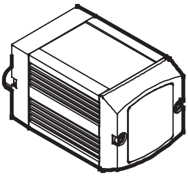
3 Compact Vision System (pictured: SBOC)

Fig. 1/2: Camera network

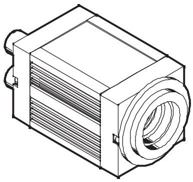
1.2 Variants

Feature	Type designation	
Sensor vision system	SBO	Compact Vision System
Design	I- C-	Integrated lens and integrated LED lighting Standardised CS-Mount/C-Mount lens adapter (only with protective lens barrel or intermediate ring)
Equipment	Q-	Surface-sensor-based camera for quality inspection
Sensor resolution	R1 R2 R3	VGA resolution (640 x 480 pixels) SXGA resolution (1280 x 1024 pixels) Wide VGA resolution (752 x 480 pixels)
Sensor type	B- C-	Monochrome Colour
Option	WB S1	Without fieldbus interface Firmware incl. firmware add-in “SBO...-Q Tools Add-In” with the following tools: – Data matrix code reader – Barcode reader – Optical character recognition (OCR).

Tab. 1/1: Type codes

Type [1]	Features	Advantages/ range of application
SBOI-Q-R1B 	– CMOS sensor with resolution 640 x 480 pixels; ½ inch; monochrome – Interfaces: Ethernet, digital I/Os and CAN – Integrated lens – Integrated LED lighting – Protection classes IP65 and IP67	– Especially suitable for short-range work (≥ 22 mm to approx. 1000 mm, longer ranges lead to loss of sharpness) – Integrated lighting for distances up to approx. 200 mm – Simple to integrate due to compact design
SBOI-Q-R1C	– Like SBOI-Q-R1B, but CMOS sensor in colour	
SBOI-Q-R3B-WB 	– CMOS sensor with resolution 752 x 480 pixels; 1/3 inch; monochrome – Interfaces: Ethernet and digital I/Os (no CAN) – Integrated lens – Integrated LED lighting – Protection classes IP65 and IP67	– Especially suitable for short-range work (≥ 20 mm to approx. 550 mm, longer ranges lead to loss of sharpness) – Integrated lighting for distances up to approx. 200 mm – Simple to integrate due to compact design
SBOI-Q-R3C-WB	– Like SBOI-Q-R3B-WB, but CMOS sensor in colour	
[1] All listed variants are also available with firmware add-in "SBO...-Q Tools Add-In". The type designation of these variants includes the extension "S1".		

Tab. 1/2: Variants of the SBOI-Q Compact Vision System

Type [1]	**Features**	**Advantages/ range of application**
SBOC-Q-R1B 	– CMOS sensor and interfaces as for SBOI-Q-R1B – Standardised CS-Mount/C-Mount lens adapter (only with protective lens barrel or intermediate ring) – No integrated lighting – Protection classes IP65 and IP67 [2]	– Any lens [3][4] can be used (focal length selectable) – Especially fast and high quality lenses for improving the photographic properties can be used – Filters and lenses depending on lens holder
SBOC-Q-R1C	– As for SBOC-Q-R1B, but CMOS sensor in colour	
SBOC-Q-R2B	– As for SBOC-Q-R1B, but CMOS sensor with resolution 1280 x 1024 pixels; 2/3 inch; monochrome	
SBOC-Q-R2C	– As for SBOC-Q-R1B, but CMOS sensor with resolution 1280 x 1024 pixels; 2/3 inch; in colour	
SBOC-Q-R3B-WB	– CMOS sensor and interfaces as for SBOI-Q-R3B-WB – Standardised CS-Mount/C-Mount lens adapter (only with protective lens barrel or intermediate ring) – No integrated lighting – Protection classes IP65 and IP67 [2]	– Any lens [3][4] can be used (focal length selectable) – Especially fast and high quality lenses for improving the photographic properties can be used – Filters and lenses depending on lens holder
SBOC-Q-R3C-WB	– As for SBOC-Q-R3B-WB, but CMOS sensor in colour	

1) These types (except SBOC-Q-R2C) are also available with firmware add-in "SBO...-Q Tools Add-In". The type designation of these variants includes the extension "-S1"
2) Only in conjunction with protective lens barrel supplied
3) Lenses with CS-Mount thread only possible without protective lens barrel; lenses with C-Mount thread only possible with protective lens barrel or intermediate ring (➔ chapter 1.5).
4) Entocentric, telecentric or hypercentric lenses can also be used

Tab. 1/3: Variants of the SBOC-Q Compact Vision System

1.2.1 Mode of operation

The SBO...-Q Compact Vision System can have different image sensors depending on the model.

The available processing functions are integrated in the operating system (firmware) of the device.

Communication

Via the Ethernet interface, the Compact Vision System can communicate via a network with the PC or a PLC. Additional I/O possibilities are available via the CAN interface (not SBO...-Q-... -WB).

Commissioning

Commissioning and operation are performed using the relevant software packages (➔ chapter 1.3).

1.2.2 Display and connecting elements

1. Integrated lens and LED lighting behind protective glass
2. Status LEDs
3. CAN interface (not SBO...-Q-...-WB)
4. Ethernet interface
5. Operating voltage supply and digital I/Os
6. Focus adjustment

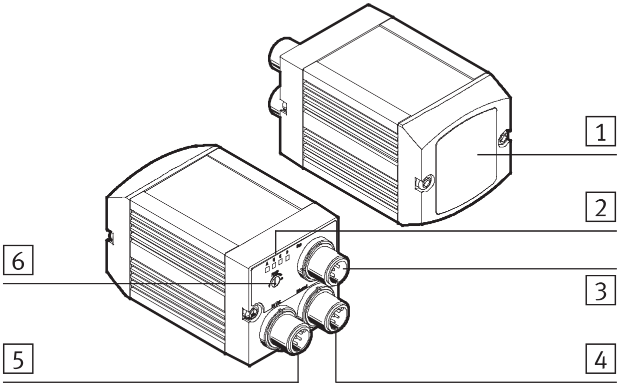


Fig. 1/3: Display and connecting elements in SBOI-Q

1 Adapter for protective barrel

2 Protective barrel

3 Lens (accessories)

4 Status LEDs

5 CAN interface (not SBO...-Q-...-WB)

6 Ethernet interface

7 Operating voltage supply and digital I/Os

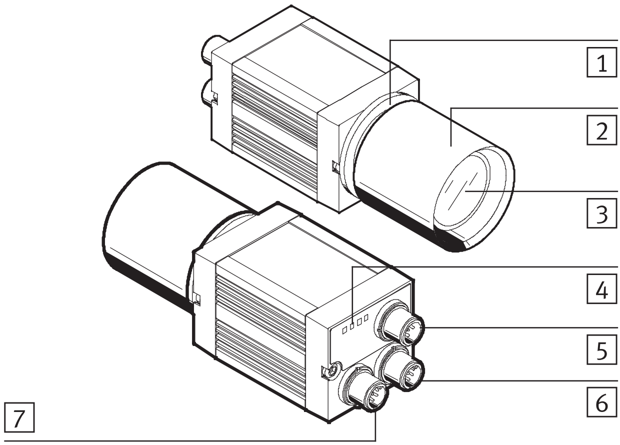


Fig. 1/4: Display and connecting elements in SBOC-Q

Connection		Description
24 V DC	Plug M12x1, 8-pin Pin allocation ➔ Tab. 3/3	Operating voltage supply and digital I/Os – Operating and load voltage supplies – I/O wiring (2 digital inputs and 3 digital outputs).
Ethernet	Plug M12x1, 4-pin, d-coded Pin allocation ➔ Tab. 3/6	Ethernet interface – Communication with higher-level devices, e.g. the PC or PLC – Data output (e.g. data analysis, etc.)
Bus	Plug M12x1, 5-pin Pin allocation ➔ Tab. 3/7	CAN interface – For extension of the I/O functions of the device and – For using the device as a CANopen master or I/O expansion (CPI modules) in combination with the PLC run-time system “CoDeSys embedded”. (Not SBO...-Q-...-WB)

Tab. 1/4: Compact Vision System connections

1.3 Software packages

The software packages CheckKon, CheckOpti and SBO-DeviceManager are used for commissioning and handling inspection tasks. They can run under the operating systems Windows 2000, XP, Vista and Windows 7 and offer a convenient user interface.

The minimum requirements your PC must meet for use of these software packages can be found in the respective software package's Help.

CheckKon provides the following functions:

Theme	Function
Configuration and commissioning	– Defining the signal behaviour – Defining the frame rate, sensor amplification – Defining the evaluation and output functions
Analysis	– Display of evaluated parts, live images, statistics and handling of check programs
Diagnosis	– Display of the device properties – Display of errors
Service	– Documentation of a system

Tab. 1/5: Functions of CheckKon

CheckOpti permits creation of check programs.

SBO-DeviceManager provides the following functions:

- Changing the network settings of the device (IP address, gateway, password, etc.).
- Transfer of firmware to the device.
- Transfer of software add-ins to the device.

1.4 Accessories

Please select the appropriate accessories from our catalogue: www.festo.com/catalogue.

Cables and plug connectors for continuous use in an industrial environment are available from dealers, e.g. HARTING Electronics GmbH & Co. KG (product series Harax M12 or Harting RJ Industrial) or from Franz Binder GmbH + Co. electrical components KG (series 825).

1.5 Selecting a lens for type SBOC

No lens is included with delivery of SBOC-Q. Lenses are available as accessories for the device.

Note
The protection classes IP65 and IP67 can only be achieved if the protective lens barrel is used (➔ chapter A.4 Technical data).

Operation with protective lens barrel

Use lenses according to the C-Mount standard.

- Please note the following lens dimensions:
  - The maximum possible lens diameter is 38 mm
  - The maximum possible length of the lens from the front edge of the lens to the flange surface of the thread side is 42 mm.

Note
- Note that with most lenses the lens length changes when the focus is reset. The setting "infinite" usually leads to the shortest lens dimension.

Operation without protective lens barrel

If you remove the protective lens barrel and the adapter, you can use lenses according to the CS-Mount standard.
A spacer ring (5 mm) is required for C-Mount lenses without protective lens barrel and without adapter for fastening the protective barrel.

- Screw an SBOL-C-5 adapter into the device to ensure the correct support dimension (➔ chapter 1.4).

Determination of the appropriate focal length

Somewhat more expensive lenses with adjustable aperture angle (zoom) enable you to adapt the field of view to your requirements. Lenses with fixed focal length are less expensive.

The suitability of a lens depends on:

- the smallest possible distance which can still be represented sharply (Minimum Object Distance - MOD)
- the focal length
- the luminous intensity
- the permissible distortion.

The object distance, i.e. the distance between the camera and the object to be photographed, can be calculated according to the laws of optics.

The following equation describes the distance between the so-called main level and the object to be depicted. With a thin lens the main level is the centre of the lens. With a standard lens the position of the main level cannot be ascertained so easily. As an estimate you can assume the centre of the lens to be the main level.

Formula	Description
$G = \frac{g}{f} \cdot B - B$	g : Object distance (working distance) G : Object size (size of field of view) f : Focal length B : Image or sensor size [1]
[1] The horizontal size, B, of the sensor is: – for SBO...-Q-R1... 6.61 mm – for SBO...-Q-R2... 8.60 mm – for SBO...-Q-R3... 4.51 mm	

Tab. 1/6: Calculation formula

The following diagrams offer help for typical fixed focal lengths and the necessary distances for specific horizontal fields of view. The vertical field of view can be obtained by multiplying the horizontal field of view by 0.75 (=480/640), 0.8 (=1024/1280) or 0.64 (=480/752).

Examples for SBO...-Q-R1:

- A lens with focal length of 12 mm generates a field of view approx. 100 mm wide at a working distance of 200 mm in SBO...-Q-R1.

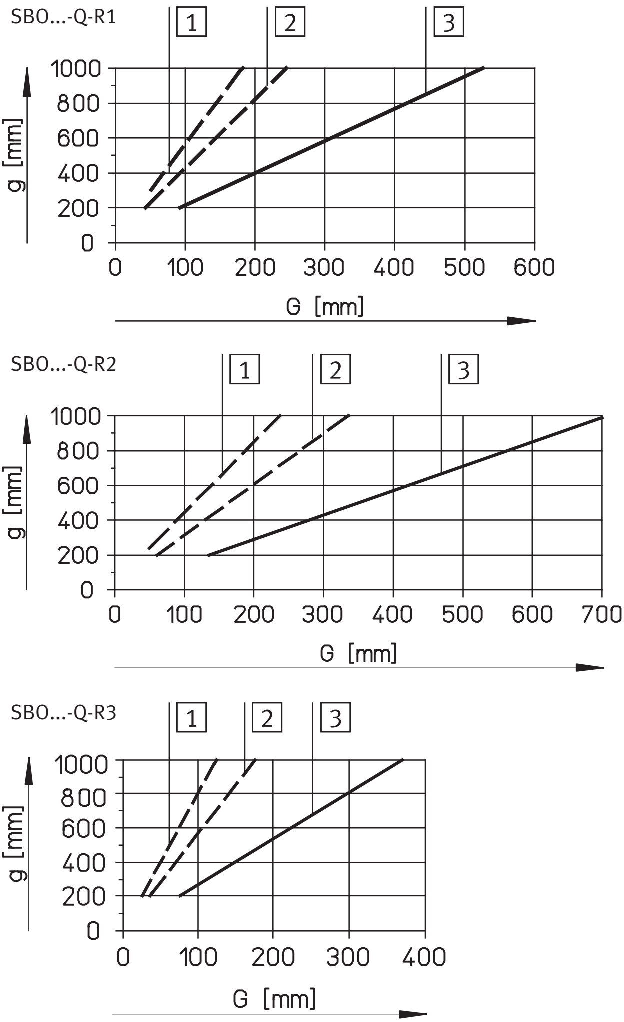
$$G = \frac{g}{f} \cdot B \cdot B \quad = \frac{200\ \text{mm}}{12\ \text{mm}}\ 6.61\ \text{mm} \cdot 6.61\ \text{mm} \;=\; 103.5\ \text{mm}$$

- A lens with focal length of 25 mm generates a field of view approx. 100 mm wide at a working distance of 400 mm in SBO...-Q-R1.

$$G = \frac{g}{f} \cdot B \cdot B \quad = \frac{400\ \text{mm}}{25\ \text{mm}}\ 6.61\ \text{mm} \cdot 6.61\ \text{mm} \;=\; 99.15\ \text{mm}$$

i

The following diagrams (➔ Fig. 1/5) can be used to make rough estimates.



1 Focal length 35 mm

2 Focal length 25 mm

3 Focal length 12 mm

Fig. 1/5: Working distances g [mm] and object size G [mm] for typical fixed focal lengths

1.6 Selecting lighting

SBOC:

- Always use external lighting.

SBOI:

- Check whether the device's internal lighting is adequate for the required application.

External lighting is commercially available.

Note
- Use shielding – e.g. a light-blocking, black housing – to protect against uncontrolled extraneous light (e.g. ceiling lights, windows, etc.).

Uncontrolled extraneous light affects the images created and thus the results.

If external lighting is used:

You can connect external lighting to the O2 output of the device. The lighting is then controlled by the device.

- Please note the correct system parameter configuration.
- Note the maximum total current at the outputs.

The light source used for lighting the check parts has significant influence on image quality. What light source is suitable for your application depends mainly on the characteristics of the check parts/the characteristics to be checked.

A light source is suitable if it emphasises the relevant check part characteristic with maximum contrast.

To recognise different characteristics or check parts, you should switch between various light sources, if necessary. Use the following light sources or lighting techniques:

Light source	Properties	Recognition of
Back light	Shines on the check part from the opposite side. The light shining through or around the check part shows the shape of the check part.	– Contours of (transparent) check parts – Levels of colourless liquids in transparent vessels
Dome light	Throws an indirect light from various directions onto the check part and generates a soft, diffuse and even light. Check parts with irregular shapes or curved surfaces are illuminated evenly. Glare is avoided.	– Printing on aluminium packaging – Coating errors or holes on check parts with crooked surface – Stains on uneven or crooked surfaces
Rod light	Generates even, direct light on long check parts. The diagonally impacting light generates diffuse reflection, which permits easy differentiation. For shiny surfaces, a polarisation filter is recommended.	– Faulty galvanisation of contacts – Edges of thin, transparent check parts
Flat angle light	Throws a direct light at a flat angle on the edges of the check part (diagonal lighting of the surface edges). This clearly emphasises deviations at the edges of the check part, and any surface faults.	– Chipping on low-contrast surfaces and edges – Thickness deviations and surface faults
Coaxial vertical light	Lights the check part in the same axis on which the lens is located. Reflecting light from shiny surfaces is reinforced, dark diffuse light is dispersed. This increases the contrast between dark and light surfaces.	– Embossings/stampings – Indentations on press-shaped pieces – Defects on the floor of deep-drawn pieces

Tab. 1/7: Light sources

Note

- Preferably use a back light if the type of recognition allows this.

This allows mostly very reliable images and results – even with interference from outside light.

Mounting

Chapter 2

Table of contents

2. Mounting 2-1

2.1 Mounting 2-3

2.1.1 Mounting the Compact Vision System 2-4

2.1.2 Dimensions of SBO Compact Vision Systems 2-5

2.1.3 Mounting with adapter kit type SBOA-HMSV-39 2-6

2.1.4 Mounting/removing the lens and shield tube on type SBOC 2-7

2.1.5 Removing the protective foil on type SBOI 2-8

2.1 Mounting

Warning
- Before carrying out mounting, installation and maintenance work, always switch off the power supply for the electronic components.

Caution
If the permitted temperature range is exceeded, e.g. due to strong external light sources, this can lead to system errors and cause damage.
- Mount the Compact Vision System in a well ventilated location, screening it from the heat emitted by other devices and sources of light.

Note
The Compact Vision System may be damaged if it is not handled correctly.
- Make sure that glass surfaces and lenses are not scratched or dirty.
- Mount the Compact Vision System in such a way that items passing by do not touch the device.

Note
- Use shielding – e.g. a light-blocking, black housing – to protect against uncontrollable extraneous light (e.g. ceiling lights, windows, etc.).

Uncontrollable extraneous light affects the images created and thus the results.

2.1.1 Mounting the Compact Vision System

- Mount the Compact Vision System using the mounting attachments so that the unit is oriented vertically and has an unobstructed view of the check part.
- Make sure that the distance between the Compact Vision System and the check part is adjusted to the optics.

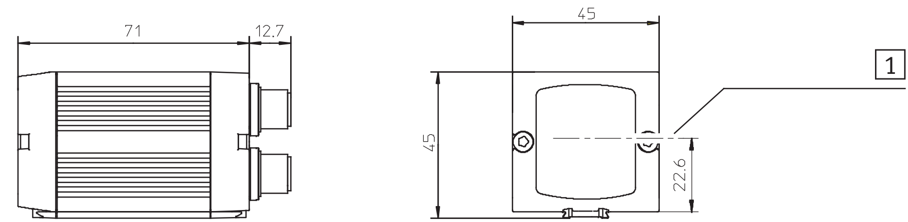
A fine adjustment is performed during commissioning (➔ chapter 4.7).

On the bottom of the device, there is a mounting profile with dovetail guide. The following adapter kits can be used for fastening:

Type	Designation	Description
SBOA-HMSV-39	Adapter kit	Adapter kit for mounting with screw-on adapter plate (contained in the adapter kit)
SBOA-HMSV-40	Adapter kit	Adapter kit for mounting with screw-on adapter plate, e.g. adapter plate HMSV-11 (not contained in the adapter kit)
SBOA-HMSV-41	Adapter kit	The adapter has an internal thread G 1/4" for fastening to commercially available photo/video tripods.

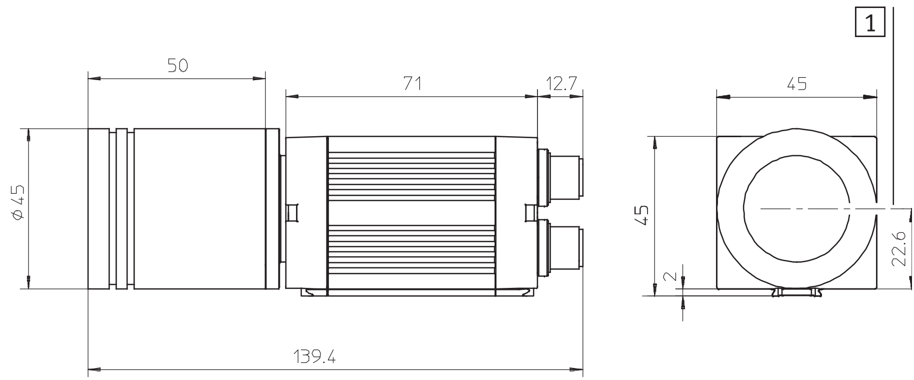
Tab. 2/1: Adapter kits for mounting

2.1.2 Dimensions of SBO Compact Vision Systems



1 Principal axis

Fig. 2/1: SBOI dimensions

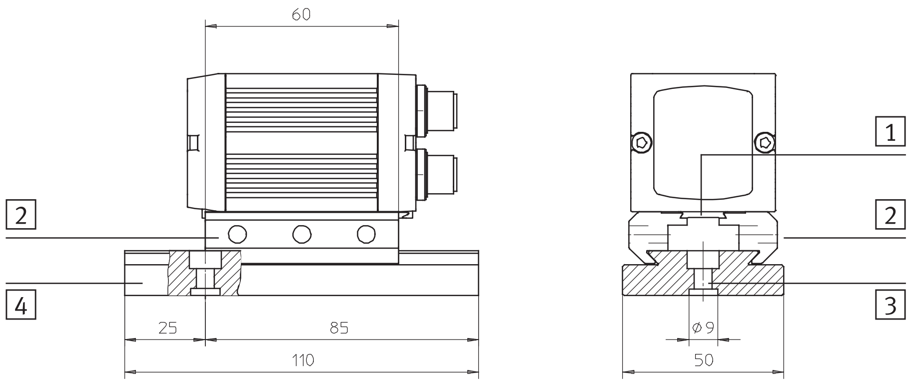


1 Principal axis

Fig. 2/2: SBOC dimensions with protective barrel

2.1.3 Mounting with adapter kit type SBOA-HMSV-39

The following diagram shows mounting with adapter kit SBOA-HMSV-39 taking as an example the SBOI Compact Vision System:



1	Dovetail of the Compact Vision System	3	Hole for socket head screw M5x16 with centring sleeve
2	Clamping devices	4	Adapter plate

Fig. 2/3: Mounting with adapter kit type SBOA-HMSV-39

Proceed with mounting as follows:

- Position the Compact Vision System so that the field of view is unobstructed and the LEDs on the rear of the housing can be seen.
- Tighten the mounting screws uniformly.

2.1.4 Mounting/removing the lens and shield tube on type SBOC

The camera is shipped sealed with a protective cover. The accompanying protective barrel consists of a support ring and shield tube.

Note
Handle the protective barrel and lens with care.
- Avoid dirt. Work in a clean environment.
- Do not touch the inside of the Compact Vision System, the lens or the glass surface of the protective barrel.

Mounting the lens

1. Unscrew the support ring from the shield tube.
2. If applicable, remove the protective cover from the lens and the camera housing.
3. Screw the support ring hand-tight into the fixture of the camera housing.
4. Screw the lens hand-tight into the support ring.

Mounting the shield tube

5. Check that the seal is seated correctly on the shield tube.
6. Pull the shield tube over the lens.
7. Screw the shield tube clockwise hand-tight into the support ring.

Removing the shield tube

- Turn the shield tube in an anti-clockwise direction and pull it forwards out of the fixture.

In the support ring of the protective barrel, there is a hole with a diameter of 2.5 mm. If you wish to separate the shield tube from the support ring of the protective barrel, you can fix the support ring, if required, with the aid of a pin (insert pin into hole).

Removing the lens

- Turn the lens in an anti-clockwise direction and pull it forwards out of the fixture.
- If applicable, attach the protective cover to the lens and to the camera housing.

2.1.5 Removing the protective foil on type SBOI

To protect them from scratching and dirt during transportation and mounting, a protective foil is attached to the front of the protective glass in SBOI devices.

- Remove this protective foil carefully before starting with commissioning.

Note
- Make sure that the protective glass is not exposed to any mechanical stress.

Scratches and grooves affect image quality and thus also the quality of the check.

Installation

Chapter 3

Table of contents

3. Installation 3-1

3.1 General instructions on installation 3-3

3.1.1 Selecting a power supply unit 3-5

3.2 Electrical connections 3-6

3.2.1 Connecting the operating voltage supply and I/Os 3-6

3.2.2 Connecting the Ethernet interface 3-11

3.2.3 Connecting the CAN interface 3-15

3.1 General instructions on installation

Note
Observe the following if compliance with the requirements of “Recognized Component Marks for Canada and the United States” is needed for your application:

- Rules for complying with UL certification can be found in the UL-specific brief operating instructions.
  The relevant technical specifications listed there also take precedence.
- The technical specifications in this documentation may include different values.

Warning
Unintended movements of the connected actuators can cause injury to human beings and damage to property.

Before carrying out installation and maintenance work, switch off the following:

- operating and load voltage supplies
- if applicable, other sources of energy, e.g. the compressed air supply.

Caution
Cables with high levels of interference can cause electromagnetic disturbances.

- Do not place controller cables in the vicinity of cables with high levels of interference If necessary, use separate wiring channels, separate cable bundles or separate cables.

Note
If you mount the SBO...-Q Compact Vision System on a moving part of a machine, you must provide strain relief on all connecting cables on the moving part.

Note
Long signal lines reduce resistance to interference.

- Make sure that the cable lengths specified in the following table are always observed:

Cable type	Permitted cable lengths
Signal cables	Max. 30 m
Supply cables	Max. 10 m

Tab. 3/1: Maximum permitted cable lengths

Note

- Check your EMERGENCY STOP concept to ascertain the measures necessary for putting your machine/system in a safe state in the event of an EMERGENCY STOP (e.g. switching off the operating voltage for the valves and output modules, switching off the compressed air).

3.1.1 Selecting a power supply unit

Warning

- Use only PELV (protective extra-low voltage) circuits as per IEC/DIN EN60204-1 for the electric power supply. Take into account also the general requirements for PELV circuits as per IEC/DIN EN 60204-1.
- Use only power units which guarantee reliable electrical isolation of the operating voltage as per IEC/DIN EN 60204-1.

Protection against electric shock (protection against direct and indirect contact) is guaranteed in accordance with IEC/DIN EN 60204-1 through the use of PELV circuits (Electrical equipment of machines - General requirements).

Recommendation: Use a separate closed-loop power supply unit which does not have to supply any other devices. You will then achieve the greatest possible resistance to interference.

Simple 24 V transformers with rectifier and filter capacitor achieve output voltages of 28 V or more at low loads. Correct operation can only be guaranteed if the permitted operating voltage range is not exceeded (➔ Technical data in appendix A.4).

3.2 Electrical connections

The following connection and display elements can be found on the rear of the Compact Vision System:

1 CAN interface (not SBO...-Q-...-WB)

2 Ethernet interface (plug M12x1)

3 Operating voltage supply and digital I/Os (plug M12x1)

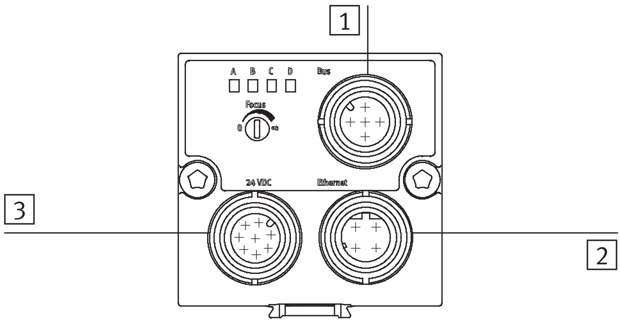


Fig. 3/1: Electrical connections of the SBO Compact Vision System

The following connection possibilities are available: (➔ chapter 4.10 and Tab. 4/5).

3.2.1 Connecting the operating voltage supply and I/Os

Caution
Damage to components

- Make sure that the max. permitted operating voltage range is not exceeded (➔ Technical data, chapter A.4).
- Protect the operating voltage supply for the Compact Vision System externally. For this, use a fast-acting 2 A micro fuse.

Note

- Use only one of the following original cables from Festo for connecting the operating voltage supply and the inputs/outputs.

The following table shows the original cables which should be used for connecting the operating voltage supply and the I/Os:

Type [1]	Designation	Description
SIM-M12-8GD-2-PU SIM-M12-8GD-5-PU	Plug socket with cable Plug socket with cable	2 m long 5 m long

[1] Cable for operating voltage supply, straight socket, M12x1, 8-pin, wire ends zinc-coated

Tab. 3/2: Cable for operating voltage supply and digital I/Os

The operating voltage is supplied together with the input/output circuit via the 8-pin M12 plug marked “24 V DC” (➔ Tab. 3/3).

The following components are supplied with +24 V DC via this connection:

– the internal electronics of the Compact Vision System

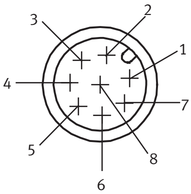
– the load current of activated outputs.

The maximum permitted current at the supply is 2 A.

Caution

Correct earthing is important for faultless operation.

- Connect the screening of the cable (SIM-M12-8GD) with low impedance to earth potential.

M12 plug [1)]	Pin	Signal	Description	Core colour [2)]
	1	I0	Rising edge [3)4)]: Trigger signal	White (WH)
	2	24 V DC	+24 V DC (tolerance: ± 10 %)	Brown (BN)
	3	Reserved	Do not connect	Green (GN)
	4	O1	Logic 1: Last evaluation indicated Good part [4)]	Yellow (YE)
	5	I1	Rising edge [3) 4)]: Apply Inputs signal Acknowledge Error signal in case of error	Grey (GY)
	6	O0	– Logic 1: Device ready for operation (Ready) [4)] – Logic 0: Device not yet ready for operation (e.g. evaluation running, system fault) [4)]	Pink (PK)
	7	0 V	0 V	Blue (BU)
	8	O2	Logic 1: Last evaluation indicated Reject part [4)]	Red (RD)
	Metal covering [5)]		Screening (shield)	

1) Tighten union nut by hand
2) Core colours of the original cable SIM-M12-8GD-...-PU
3) The signal levels/edges can be configured via system parameters; the specified description corresponds to the standard configuration.
4) The function can be configured via system parameters; the specified description corresponds to the standard configuration.
5) Connect cable screening with low impedance to earth potential (➔ Fig. 3/2)

Tab. 3/3: Operating voltage connection and I/Os at the 8-pin M12 plug "24 V DC"

The function and functional behaviour of the I/Os can be configured in CheckKon via system parameters. The functional behaviour of the I/Os in the different evaluation modes must be differentiated (➔ chapter 4.9 ff.).

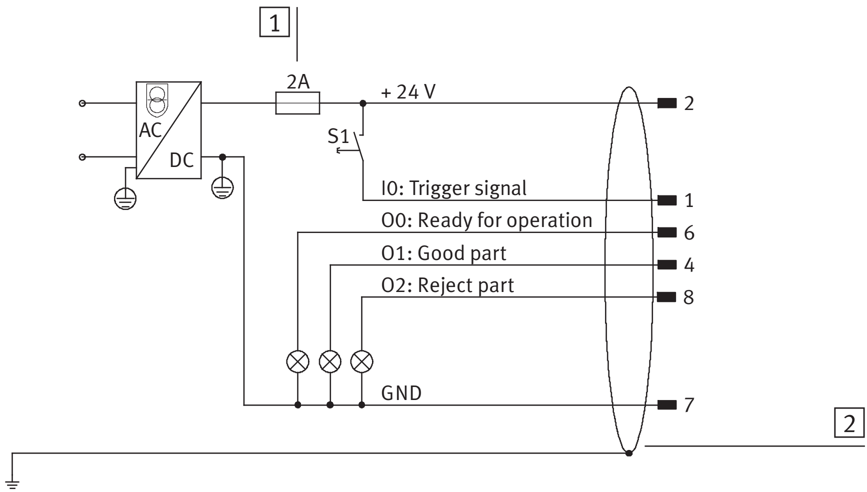
Connection example

Fig. 3/2 shows an example of a connection using a configuration of the system parameters with the standard values:

Internal I/Os	Configuration
Polarity I0	Trigger signal = rising edge
Polarity O0	Ready for operation = logic 1
Function at O1	Good part = logic 1
Function at O2	Reject part = logic 1

Tab. 3/4: System parameters with standard configuration

- Please note that:
  - The tolerance of 24 V DC ± 10 % must be observed.
  - The power supply for the Compact Vision System must be fused externally. A fast-acting 2 A micro fuse.
  - The total current of all outputs must not exceed 1.5 A.
  - The cable screening must be connected with low impedance to the earth potential.



1 External fuses
2 Cable shield

Fig. 3/2: Example configuration of system parameters with the standard values.

3.2.2 Connecting the Ethernet interface

Note
Unauthorised access to your Compact Vision Systems can cause damage or malfunctioning.

- Ask your system administrator how you should protect your network against unauthorized access, e.g. with a firewall.

Note
With an active connection to the cameras in the network, large amounts of data can be transmitted, depending on the mode of operation. The network between the PC and cameras will be correspondingly highly loaded. A direct connection is therefore to be preferred.

- If in doubt, ask your network administrator whether appropriate band widths are available for you or what an optimum network structure for you should look like.
- Observe the necessary system requirements.

For commissioning Compact Vision Systems you must create a connection between your PC and the Compact Vision Systems via Ethernet. For connection to a network or a PC you will require the following cables:

Connection	Type	Designation	Description
Connection via hub or switch	SBOA-K30E-M12S	Cable	Ethernet cable for simple demands [1] – Straight socket, M12, 4-pin, d-coded – RJ-45 Ethernet plug – 3 m long
Direct connection with the PC	SBOA-K30E-M12S	Cable	
	–	Coupling	Cable coupling for RJ45 plug connector [2]
	–	Cable	Ethernet cross link [2]

[1] The Ethernet cable SBOA-K30E-M12S has been designed for short-term use as a diagnostic cable or for continuous use as a fixed cable for simple demands.
[2] Only required if the network connection of the PC does not support automatic adaptation of the send and receive cables (AUTO MDI-X). This accessory is available from dealers.

Tab. 3/5: Cables for the SBO Compact Vision System

For special use in an industrial environment use a screened flexible Ethernet round cable of category 5 which will fulfil your requirements as regards resistance to oil, bending radius, permitted bending cycles etc.
Connections: M12 socket, 4-pin d-coded and RJ45 plug

Cables and plug connectors for special requirements are available from dealers,
e.g. Franz Binder GmbH & Co. (product series 825) or from HARTING Electronics GmbH & Co. KG (product series Harax® M12 or Harting RJ Industrial®).

Ethernet connection

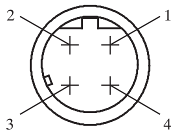
Via the Ethernet interface, the connection can be created to the PC and displays or higher-level controllers. To permit a connection, several prerequisites with regard to the device’s network address and the PC must be met (➔ chapter 4.5).

The network characteristics of the device can be adjusted using the SBO-DeviceManager (➔ chapter 4.5).

For data exchange via EasyIP, Telnet or other protocols, relevant system parameters must be selected (➔ chapter 4.10.7).

The results of a parts check (camera image, features and quality decision) can be displayed via an HTML-capable browser.

For connection to the Ethernet there is a d-coded M12 plug on the rear of the Compact Vision System.

M12 plug [1)]	Pin	Signal	Description
	1	TD+	Transmitted data+
	2	RD+	Received data +
	3	TD-	Transmitted data –
	4	RD-	Received data –
	Metal covering		Screening (shield)
[1)] d-coded			

Tab. 3/6: Pin allocation of the Ethernet interface

The Ethernet interface of the Compact Vision System complies with standard 10BaseT/100BaseTX for 100 Mbit networks.

Note
- Use a screened plug connector which will guarantee continuous contact between the screening/shield and the Compact Vision System.
- Connect the screening of the Ethernet cable with low impedance to earth potential.

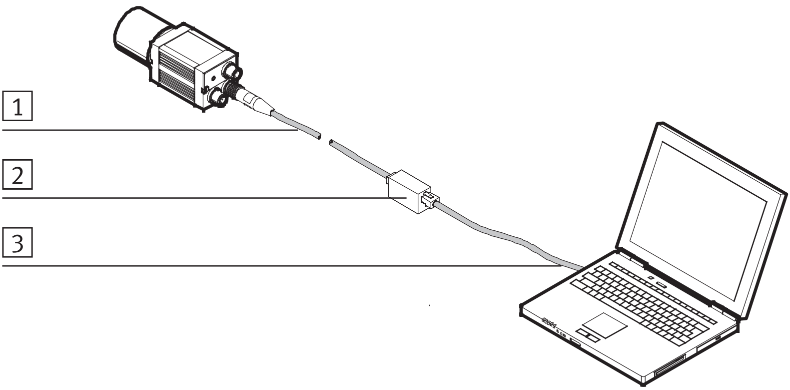
Connection via hub or switch

Recommendation: Use network components which support data rates of at least 100 Mbit/s.

If using a router, make sure that this is set so that the multicasts of address 239.255.2.3. are passed on. This address is used to search for devices in the network. If the routers are not configured accordingly, the devices cannot be found using the search function. If in doubt ask your network administrator.

Direct connection with the PC

If the PC's network connection does not support automatic adaptation of the send and receive cable (AUTO MDI-X), you will also require a crossover cable and a cable coupling in addition to the original cable (➔ Tab. 3/5).



1 Original cable SBOA-K30E-M12S

2 Cable coupling

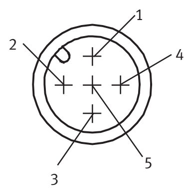
3 Crossover cable

Fig. 3/3: Direct connection with the PC (pictured: SBOC)

3.2.3 Connecting the CAN interface

Compact Vision Systems with a CAN interface (not SBO...-Q-... -WB) offer the following possibilities:

- Connecting external I/O modules (I/O expansion). Observe the installation and commissioning instructions in section 4.10.4.
- Connecting the Compact Vision System to a CP node (use as a CPI module). Observe the installation and commissioning instructions in section 4.10.5.
- Connection of any CANopen slaves by using the integrated run-time system CoDeSys. The Compact Vision System has full master function under CoDeSys. Observe the installation and commissioning instructions in section 4.10.6.

M12 plug	Pin	Signal
	1	n.c. (not connected)
	2	n.c. (not connected)
	3	GND
	4	CAN_H
	5	CAN_L

Tab. 3/7: Pin allocation of the CAN interface (bus)

The device does not feed any supply current into the CP string, nor does it take any supply current from the CP string.

The functional principle of the CAN interface depends on the system parameters set for the device. For the CAN interface to be used, it must be configured appropriately. The protocol to be used is also defined here (➔ chapter 4.10.4 and 4.10.5).

Commissioning

Chapter 4

Table of contents

4. Commissioning 4-1
4.1 Notes on commissioning 4-3
4.2 Installing the software packages 4-4
4.3 Network settings on the PC 4-5
4.4 Firewall settings on the PC 4-8
4.5 Network settings on the Compact Vision System 4-11
4.6 CheckKon connection to the Compact Vision System 4-14
4.7 Settings on the Compact Vision System 4-16
4.8 System parameters for preprocessing 4-19
4.9 Selection of the evaluation mode 4-21
4.9.1 I/O process in “Triggered” evaluation mode 4-22
4.9.2 I/O process with the “Free run” evaluation mode 4-28
4.9.3 I/O sequence with the “Fixed frame rate” evaluation mode 4-39
4.10 Connection to higher-level controller (PLC/IPC) 4-44
4.10.1 General information on use of inputs 4-46
4.10.2 General information on use of outputs 4-50
4.10.3 Use of internal I/Os 4-50
4.10.4 Use of the I/O expansion 4-52
4.10.5 Use of the device as CPI module at CP nodes 4-57
4.10.6 Use of the device as CANopen Master 4-63
4.10.7 Use of the Ethernet interface with EasyIP 4-65
4.10.8 Use of the Ethernet interface with Telnet 4-66
4.10.9 Use of the Ethernet interface with Telnet streaming 4-77
4.10.10 Use of the Ethernet interface with Telnet XML 4-79
4.10.11 Use of the Ethernet interface with Modbus 4-90
4.10.12 Use of the Ethernet interface with EasyIP 4-92
4.10.13 Use of CoDeSys 4-101
4.10.14 Display of inspection results with the “SBO...-Q WebViewer” 4-107
4.11 Creation of the check programs 4-110
4.12 Checking the system settings 4-111
4.13 Instructions on operation 4-112

4.1 Notes on commissioning

The device should be mounted and installed prior to commissioning.

- Use SBO-DeviceManager to change the network settings of the device.
- Carry out commissioning using the CheckKon software package (➔ CheckKon Help).
- Create check programs using the CheckOpti software package (➔ CheckOpti Help).

The commissioning steps are normally performed in the order shown in the next chapter. To optimise the system, it may be necessary to repeat steps that have already been carried out.

4.2 Installing the software packages

Note
The software packages CheckOpti, CheckKon and SBO-DeviceManager can be downloaded free of charge from the Internet.

Please consult your local Festo service if you have any questions.

The software packages are installed on your PC with an installation program.

SBO-DeviceManager is part of the CheckKon installation.

Note
Administrator rights are required for installation of the software packages.

You can install the software packages from the CD ROM as follows:

1. Close all programs.
2. Place the corresponding CD in your CD ROM drive.
   If Auto-Run is activated on your system, the installation starts automatically and you can skip steps 3 and 4.
3. Select [Execute] in the Start menu.
4. Enter D:\setup (if necessary replace D by the letter of your CD ROM drive).
   The Setup program installs the application(s).
5. Follow the instructions on the screen.

4.3 Network settings on the PC

Before a connection to the device can be established, the PC network connection that is to be used must be correctly configured.

The settings of the PC's network connection can be found in the Windows System Control under Network Connections. The available LAN connections are displayed there.

- Select the LAN connection you want to use to connect to the device, and display its characteristics.

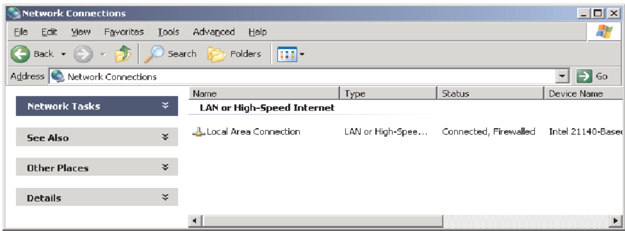


Fig. 4/1: Network and data transmission connections

- In the "Characteristics of LAN Connection" window, check in particular the characteristics of the "Internet Protocol (TCP/IP)" entry.

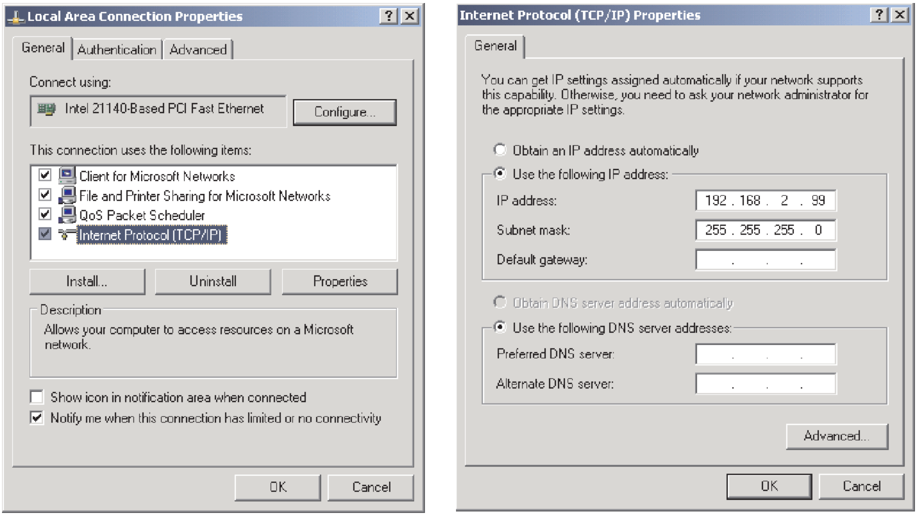


Fig. 4/2: Characteristics of LAN connections and characteristics of Internet Protocol (TCP/IP)

If the network connection is configured so that it automatically obtains your IP address (from a DHCP server), the DHCP server must be reachable via the network connection.

If that is not the case (e.g. with a direct connection to the Compact Vision System), the network connection must be provided with details of which IP address is to be used (➔ Fig. 4/2).

The network connection settings of PCs and of the Compact Vision System must be chosen on the basis of one another:

- Select different IP addresses that have not yet been assigned to other network participants (➔ Tab. 4/1).
- Choose an identical address range via the subnetwork mask (➔ Tab. 4/1).

IP settings	PC	Compact Vision System
IP address	192.168.2.99	192.168.2.10
Subnetwork mask	255.255.0.0	255.255.0.0

Tab. 4/1: Example of suitable network settings

You can find additional information on this in chapter 4.5 and appendix A.2.

Note
The following address is set on the Compact Vision System when it leaves the factory: 192.168.2.10

Note
Restart the PC if network settings have been changed.

4.4 Firewall settings on the PC

If there is a firewall on your PC, network traffic is permitted or prohibited according to the firewall rules defined. This also affects the communication of the software package with the Compact Vision System.

There are two options for permitting communication:

1. When the software packages installed are launched and the first attempt at communication between the program and the Compact Vision System is made, a dialogue normally appears to indicate that the firewall is blocking communication.

- Give the firewall permanent permission for the program to communicate via the network.

Setting the firewall at a later stage (manually) is normally also possible. Information on this can be found in the description / Firewall Help.

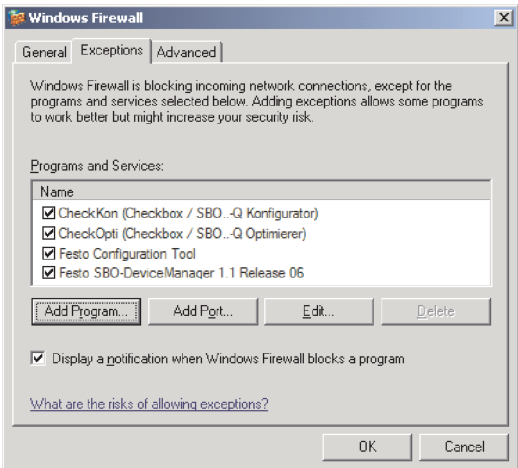


Fig. 4/3: Windows firewall

2. The software packages installed communicate with the Compact Vision System via so-called ports.

   Some firewalls permit network traffic through release of these ports.

- In the firewall, enable use of the following ports:

Port	Transmission	Purpose
502 1)	TCP	Modbus protocol
995	UDP	EasyIP protocol
1200	TCP	CoDeSys programming environment
2222	UDP	EtherNet/IP (multicast)
4386	TCP	Firmware download, add-in download and backup with PC software
9997 1)	TCP	Telnet XML
9998 1)	TCP	Telnet streaming
9999 1)	TCP	Telnet
10000	TCP	Data exchange with PC software
10001	TCP	Data exchange with PC software
10002	UDP	Device search with PC software (multicast)
44818	TCP	EtherNet/IP

1) Standard setting for the Compact Vision System

Tab. 4/2: Ports to be opened for software packages

Note
Opening a port poses a greater safety risk than program-based enabling.

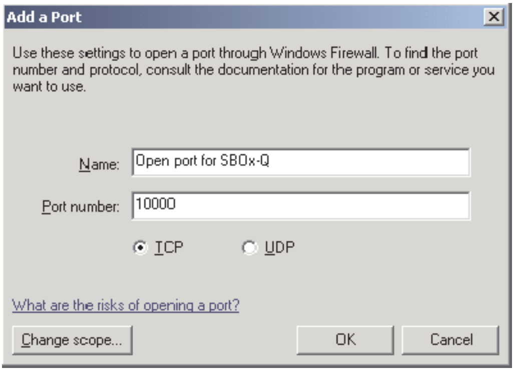


Fig. 4/4: Opening port #10000

4.5 Network settings on the Compact Vision System

You can use SBO-DeviceManager to define the settings for the IP addressing of your Compact Vision System. Proceed as follows:

1. Launch SBO-DeviceManager by double-clicking on the corresponding symbol on the Windows desktop,
   or for standard installation:
   In the Windows menu [Start], select the entry
   [Festo Software] – [SBO-DeviceManager].

2. Create a connection to the device using SBO-DeviceManager.

3. Set the network parameters of your Compact Vision System using SBO-DeviceManager (IP address, network mask and if applicable the IP address of the gateway).

Creating a connection to the device

Search function for devices

SBO-DeviceManager offers a search function. This search function allows creation of a connection to devices whose network address is not known.
The search function uses a special Ethernet procedure (multicast). With this procedure, messages can be transmitted simultaneously to several slaves or to a closed group of slaves.
The SBO devices always use the address 239.255.2.3 for this process. This cannot be changed.

If the search takes place within a network node, no special configuration of the network node (router, server etc.) is necessary. However, if devices which lie behind further network nodes need to be found, the network nodes must be enabled for multicast. You must also adapt the properties of the device search (search range and search duration) accordingly. If you have any problems, please consult your system administrator.

If you cannot create a connection to the device with the search function of the SBO-DeviceManager:

- Check whether the network nodes are enabled for multicast.
- Check the search range and search duration in SBO-DeviceManager.

If this is not successful:

- Connect the device as directly as possible to your PC.
  In doing this you will exclude incorrect network configurations.

Also, the search function will find the device even if the setting “Visibility in search requests” is set to “Local (1 Router)”.

Undertaking network settings

If you have created a connection using SBO-DeviceManager, you can also modify the network settings of the devices.

– Activation of automatic assignment of the IP address (requires a DHCP server in the network)

– Specification of a fixed IP address.

Assign IP address automatically

Select this setting if you are using a DHCP server in your network and if the IP address is to be assigned automatically via DHCP (DHCP stands for Dynamic Host Configuration Protocol). The DHCP server manages a range of IP addresses and assigns them to the DHCP-capable end terminals.

Note
The IP address of the equipment is set to 0.0.0.0 if
– the DHCP server is not found
– assignment of the IP address fails.

Specifying a fixed IP address

If you are not using a DHCP server, you must specify a fixed IP address. Make sure:

– that the IP address and the network mask of the devices are suited to the network settings of your PC
– that the IP addresses of the devices are different (even if the devices are not connected in the network at the same time).

• If you have any problems, please consult your system administrator.

Note
Additional information on IP addressing can be found in the appendix and in the SBO-DeviceManager manual.

4.6 CheckKon connection to the Compact Vision System

Note
- Switch the device to Stop status.

This is a prerequisite for connecting, disconnecting and making changes to the device.

CheckKon is used for the further commissioning of the Compact Vision System. Carry out the following steps:

1. Connect the Compact Vision System via the Ethernet interface to your PC or hub/switch. Follow the instructions in chapter 3.

2. Start CheckKon by double-clicking on the corresponding symbol on the Windows desktop,
   or for standard installation:
   In the Windows menu [Start], select the entry [Festo Software] – [Festo CheckKon].

The program starts and opens the dialogue window "Welcome to CheckKon" (start dialogue). This dialogue window is used to configure the connection to the Compact Vision System.

3. From the "Connection ..." possibilities offered, choose the entry "via Ethernet". Confirm this selection with the mouse; click on "Continue >".

4. Enter the IP network address in the following dialogue (factory setting: 192.168.2.10). Confirm this entry by clicking on "Continue >" – or click on "Search..." to find and select reachable devices.

A connection is set up between the PC and the Compact Vision System.

Note
If the network connection is interrupted, the device may remain in “assigned status”. In this case, setting up a new connection to the device is not possible.

This occurs when:

– Ethernet connection lines are unplugged
– The power supply to Ethernet network components (e.g. switch) is disconnected
– Deactivation of the network card in the PC due to energy-saving mode.

• Briefly disconnect the power supply to restart the device.

Additional information can be found in the CheckKon Help, e.g. regarding the basic functions Device connection, Device names and Device control.

4.7 Settings on the Compact Vision System

1. In CheckKon, open the “Live Image” window.

2. Activate the transfer and display of the camera image by clicking on this button.

3. If necessary, activate the display of the camera image by clicking on this button also.

The camera image corresponds to the image captured by the Compact Vision System, taking into account the system parameters set.

4. If necessary, switch on the Dynamic Help.
   In the “Live Image” window, you will then receive help in setting the image and lighting control parameters.

Now perform the following steps:

1. Switch the lighting on and direct it towards the check area.
   If the lighting is controlled by the device, set the system parameters for the lighting control appropriately.

2. Roughly adjust the following image parameters until an image of the check area can be identified:

   – Image field area (change the area displayed using the mouse or by inputting the parameters)

   – Exposure time

   – Sensor amplification

3. Place a check part in the check area, as for operation.

4. Now perform a fine adjustment of the camera mounting position, if necessary. The Compact Vision System is optimally positioned in relation to the check part when the part is in the centre of the camera image.

5. Adjust the optics of the Compact Vision System so that the camera image sharply depicts the check part. Adjust the lens focus accordingly. In SBOC, the protective lens barrel must be removed for this purpose.

Note

In SBO...-Q-R3, changes to system parameters relevant to creation of the image may only take effect or become apparent in the second image after the changes were made (not evident from the "Live Image" window).

- For devices SBO...-Q-R3, always take an additional (unused) image after you have changed these parameters.

A "Siemens star" is provided in this manual as a template for setting the image sharpness (➔ chapter A.3).

SBOI	SBOC with standard lens from Festo
• Use a screwdriver to turn the focus adjuster on the rear of the camera housing.	1. Loosen the locking screw on the lens. 2. Turn the focus ring to focus on an object. 3. Tighten the locking screw again slightly.

Tab. 4/3: Setting the focus

6. In SBOC devices: Adjust the aperture of the lens so that the sensor has enough light available.

SBOI	SBOC with standard lens from Festo
–	1. Loosen the locking screw on the lens. 2. Turn the aperture ring to adjust the aperture. 3. Tighten the locking screw again slightly.

Tab. 4/4: Setting the aperture

7. Now adjust the lighting mounting position, if necessary, to provide optimal lighting for the characteristics to be checked.
   The lighting is optimally oriented if it emphasises the relevant check part characteristics with maximum contrast.

8. Now perform an (iterative) optimisation of all steps and settings. Repeat steps 4 to 7.

The setting is optimal when the relevant check part characteristics are sharply depicted and emphasised with maximum contrast.

4.8 System parameters for preprocessing

Preprocessing

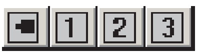
This involves processing the image data for the camera image in three consecutive steps.

In preprocessing, the image is digitally optimised and changed. The characteristics to be checked should be optimally highlighted to permit reliable checking of parts.

The camera image is the image captured by the Compact Vision System sensor, taking into account the system parameters set.

The images “Preprocessing 1 to 3” show the results of the corresponding preprocessing steps, taking into account the system parameters set.

The parts check takes place on the basis of the camera image and the image “Preprocessing 3”, whereby the test part must be depicted as a black object in Preprocessing 3; white surfaces represent the background.



Switch the display of the various images in the “Live Image” window on and off.

The preprocessing steps must be configured in the “Live Image” window in CheckKon so that the characteristics to be checked in the image “Preprocessing 3” are reliably depicted.

In addition, preprocessing should remove the areas of the image that are not part of the check part or of the characteristics to be checked, e.g. background, workpiece carrier, etc., as they can hinder determination of the check features.

The preprocessing functions available depend on the firmware used.

Switch on the Dynamic Help. In the “Live Image” window, you will then receive help on the preprocessing functions.

Depending on the preprocessing function used, the calculation time per check part can increase greatly. For that reason, image optimisation with regard to the characteristics to be checked should always be striven for even before preprocessing – e.g. through optimisation of:

– Type and position of lighting

– Position of the Compact Vision System

– Position of the inspection part

– Optical characteristics (lens, filter, etc.)

(➔ chapter 4.7).

4.9 Selection of the evaluation mode

The evaluation mode determines when a image of an inspection part is taken, processed and checked and the results output.

Selection of the evaluation mode specifies how the device reacts to input signals or when the test results are provided at the outputs. Note the descriptions of the signal processes starting in chapter 4.9.1 ff.

The following evaluation modes can be selected via the "Evaluation Mode" system parameter in the "System Parameters" window in CheckKon:

- Triggered: Frame capture and inspection with each valid triggering signal.
- Free run: Consecutive image taking and checking (without fixed frame rate) as long as the trigger signal is on.
- Fixed frame rate: Consecutive image taking and checking with established frame rate as long as the trigger signal is on (not for SBO...-Q-R3).

Selection of the evaluation mode depends on the respective application, in particular on:

- the inspection part rate and inspection part flow
- the inspection part (individual piece or continuous)
- interaction with a higher-level controller.

Application and function of the modes

Triggered

Testing of individual parts in stop-and-go or slow part flow.

The trigger signal is generated by a master controller or a sensor as soon as the check part is in front of the Compact Vision System. The inspection results are output following the inspection, and the device then waits for the next valid triggering signal.

Free run

Inspection of individual or continuous parts with medium to fast (continuous) part flow.

The trigger signal is present permanently, irrespective of whether or not there is a check part in front of the Compact Vision System. The device acts like a basic sensor. The inspection results are output following the inspection. After that, the device begins immediately with the next inspection. Additionally, there is the possibility to set up a image trigger.

Fixed frame rate

Inspection of continuous parts at a constant speed.

The triggering signal is permanently present. The inspection results are output following the inspection. The device starts the next inspection in accordance with the defined frame rate.

4.9.1 I/O process in “Triggered” evaluation mode

In the “Triggered” evaluation mode, exactly one cycle is started through a valid signal at the “Triggering Signal” input (edge controlled). One cycle thereby includes the image evaluation and output of results.

Triggering signals are valid only under the following conditions:

– Output “Ready for Operation” signals logic 1

and

– any time period set since output “Ready for operation” = logic 1 has expired.
  This setting is made over the system parameter “Ignore triggering signal after ready for operation”.

Image evaluation starts with the triggering signal. During image evaluation, the output “Ready for operation” signals a logic 0.

The start of image creation and triggering of the lighting can be established regarding the time of the triggering signal.

- Set the following system parameters:
  - Start of image evaluation after triggering
  - Start of the lighting
  - Duration of the lighting.

The results are provided at the outputs no earlier than the time image evaluation is completed.

If the system parameter setting “Earliest start of output is after triggering signal termination” = yes, output of results can be delayed by extending the triggering signal. In this case, the outputs can only be written when the signal at the “Triggering signal” input has been ended. With this, a higher-level controller (PLC) can signal its readiness to accept the results of the parts inspection.

- Also set output time and output duration via the following system parameters:
  - Earliest start of output after triggering
  - Ignore triggering signal after ready for operation

The output “Ready for operation” is set to logic 1 only when the results are valid at all outputs. The results can now be processed, e.g. by a higher-level controller (PLC/IPC).

These results remain valid at the outputs only until:

- the specification "Earliest start of output after triggering" (system parameter) of the next started cycle has expired and the signal at the "Triggering Signal" input of the next started cycle has been cancelled

or

- the system parameter "Reset output at outputs after expiration of Ignore triggering signal" = On and the time "Ignore triggering signal after ready for operation" has expired

or

- the system parameter "Reset outputs at outputs during evaluation" = Yes has been set and a new evaluation has begun.

Through the above conditions, it is possible to specify minimum times for evaluation and output as well as influence their starting time. A higher-level controller (PLC/IPC) can thus start a new cycle and only then read the results of the previous cycle. Such overlapping permits achievement of a high evaluation rate.

The outputs of the output module of the I/O extension O0 to O3, like the internal outputs, are configured via the system parameters, that is, they are set or reset together with the internal outputs.

Signal sequence with standard settings

The signal sequence with standard settings of the system parameters in the “Triggered” evaluation mode could appear as follows.

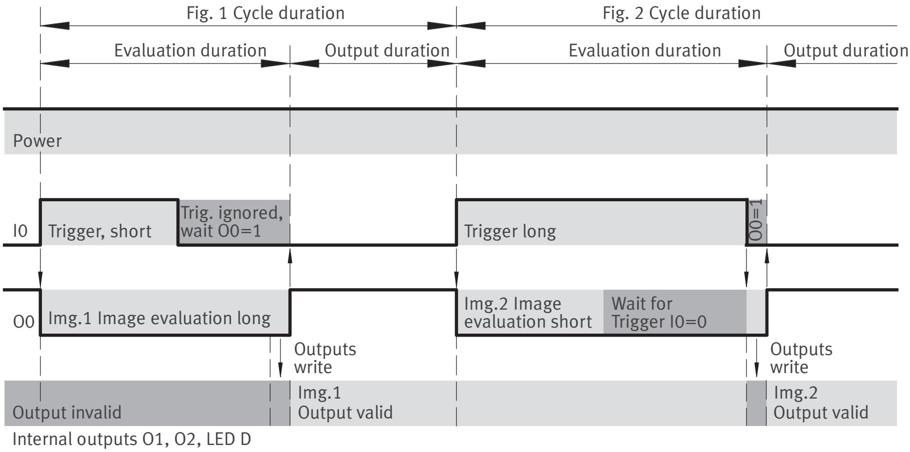


Fig. 4/5: “Triggered” evaluation mode – Signal curve with standard settings

Settings of the system parameters (standard values):

- Earliest start of output after triggering = 0 ms
- Ignore triggering signal after ready for operation = 0 ms
- Start of image evaluation after trigger = 0 ms
- Earliest start of output is after trigger signal termination = Yes
- Reset outputs at outputs after “Ignore triggering signal” = Off
- Reset outputs at outputs during Analysis = No
- Function at O2 = reject part
- Start of lighting = automatic
- Duration of lighting = automatic

Signal sequence with system parameters

"Triggered" evaluation mode.

Case 1

The cycle duration results from system parameter "Earliest start of output after triggering" or from long triggering signal of the PLC (PLC synchronisation)

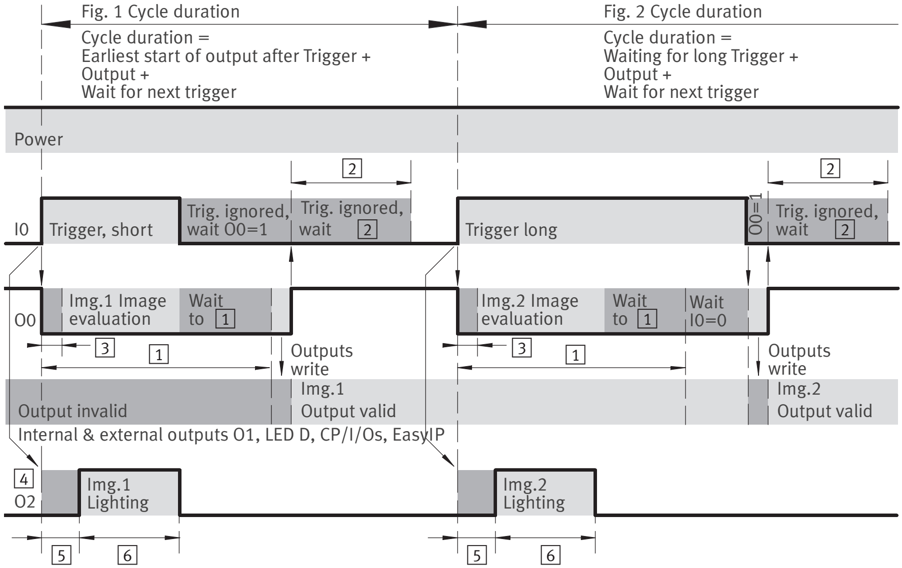


Fig. 4/6: "Triggered" evaluation mode – Signal sequence using system parameters – Case 1

Settings of the system parameters:

1. Earliest start of output after trigger = 220 ms
2. Ignore trigger signal after ready for operation = 110 ms
3. Start of lighting after trigger/cycle start = 40 ms
4. Start of image evaluation after trigger = 20 ms
5. Function at O2 = external lighting, start of lighting and duration of lighting = manual
6. Duration of lighting = 100 ms
   – Earliest start of output is after trigger signal termination = Yes
   – Reset outputs at outputs after "Ignore trigger signal" = Off
   – Reset outputs at outputs during evaluation = No

Case 2

The cycle duration is derived from the system parameter "Earliest start of output after triggering" or from a long image evaluation duration:

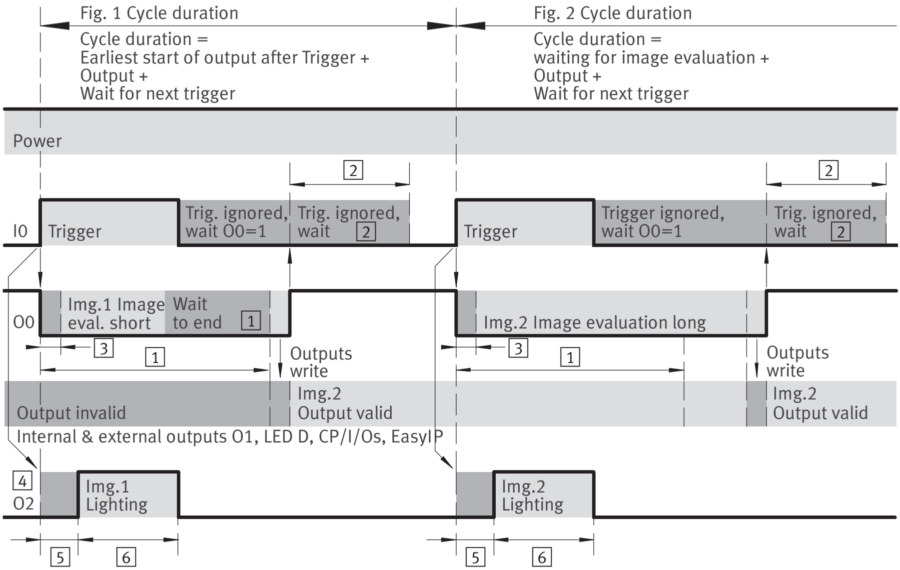


Fig. 4/7: "Triggered" evaluation mode – Signal sequence using system parameters – Case 2

Settings of the system parameters:

1 Earliest start of output after trigger = 220 ms

2 Ignore trigger signal after ready for operation = 110 ms

3 Start of image evaluation after trigger = 20 ms

4 Function at O2 = external lighting, start of lighting and duration of lighting = manual

5 Start of lighting after trigger/cycle start = 40 ms

6 Duration of lighting = 100 ms
- Earliest start of output is after trigger signal termination = Yes
- Reset outputs at outputs after "Ignore trigger signal" = Off
- Reset outputs at outputs during evaluation = No

Simplified depiction of the signal curves; these do not contain jitter, run times or system-related delay times.
The duration for the write process of the outputs and thus the duration for an invalid output condition is indefinite with use of external outputs (I/O expansion, device as CPI module, EasyIP or CheckKon). This duration is approximately 1 ms if the external outputs are deactivated by system parameters and only the internal outputs are used.

4.9.2 I/O process with the "Free run" evaluation mode

In the "Free run" evaluation mode, the mode remains active as long as a logic 1 (condition-controlled) is present at the "Trigger signal" input. During this time, images are cyclically generated and evaluated.

"Free run" evaluation mode without image trigger

Process

Image evaluation starts with the cycle start. During image evaluation, a logic 0 is emitted at the "Ready for operation" output. Between two cycles, a logic 1 can be output at the "Ready for operation" output.

The start of image creation and triggering of the lighting can be established regarding the time of the triggering signal.

- Set the following system parameters:
  - Start of image evaluation after cycle start
  - Start of the lighting
  - Duration of the lighting

The results are provided at the outputs no earlier than the time image evaluation is completed.

- Set output time via the following system parameters:
  - Earliest start of output after cycle start
  - Ignore triggering signal after ready for operation

Additionally, the outputs are still reset (logic 0), depending on the system parameters:

- Reset outputs at outputs after "Ignore triggering signal"
- Reset outputs at outputs during analysis.

Note
The rate at which the images are taken or the results output varies and depends on the evaluation duration of the individual images.

Function

In this evaluation mode, the device acts like a simple sensor that permanently inspects and outputs. As a result, continuous analysis can be achieved, but synchronisation of the outputs with a higher-level controller (PLC) is only conditionally possible.

If such signalling of the result outputs is needed, the system parameter "Ignore triggering signal after ready for operation" can be used. In this case, after output of the results, the "Ready for operation" output goes over to logic 1 for the set duration. The higher-level controller (PLC) can now read the validly present results.

If the “Trigger signal” input still shows logic 1 after expiry of the set duration, the next evaluation is started.

The results are only written to the outputs when the device is finished with the image evaluation and specification of the “Earliest start of output” system parameter has expired.
The outputs remain valid until the specification of the system parameter “Earliest start of output after cycle start” of the next cycle has expired.

Outputs to outputs can be invalid even earlier, depending on the setting of the following system parameters:

- Reset outputs at outputs after “Ignore triggering signal”
- Reset outputs at outputs during analysis.

If the “Trigger signal” input is reset to logic 0, the free run is ended. After the momentary image evaluation is completed (incl. “Earliest start of output after cycle start”), the “Ready for operation” output goes back to logic 1.

The outputs of the output module of the I/O extension O0 to O3, like the internal outputs, are configured via the system parameters, that is, they are set or reset together with the internal outputs.

Signal sequence with standard settings

The signal sequence with standard settings of the system parameters in the “Free run” evaluation mode could appear as follows.

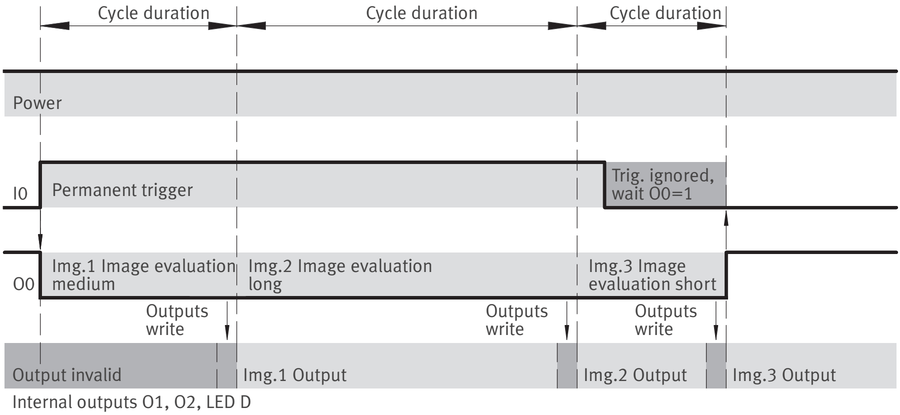


Fig. 4/8: “Free run” evaluation mode – Signal sequence with standard settings

Settings of the system parameters (standard values):

- Earliest start of output after cycle start = 0 ms
- Ignore trigger signal after ready for operation = 0 ms
- Start of image evaluation after cycle start = 0 ms
- Function at O2 = reject part
- Start of lighting = automatic
- Duration of lighting = automatic
- Reset outputs at outputs after “Ignore triggering signal” = Off
- Reset outputs at outputs during Analysis = No

Signal sequence with system parameters

The signal sequence using system parameters in the "Free run" evaluation mode could appear as follows.

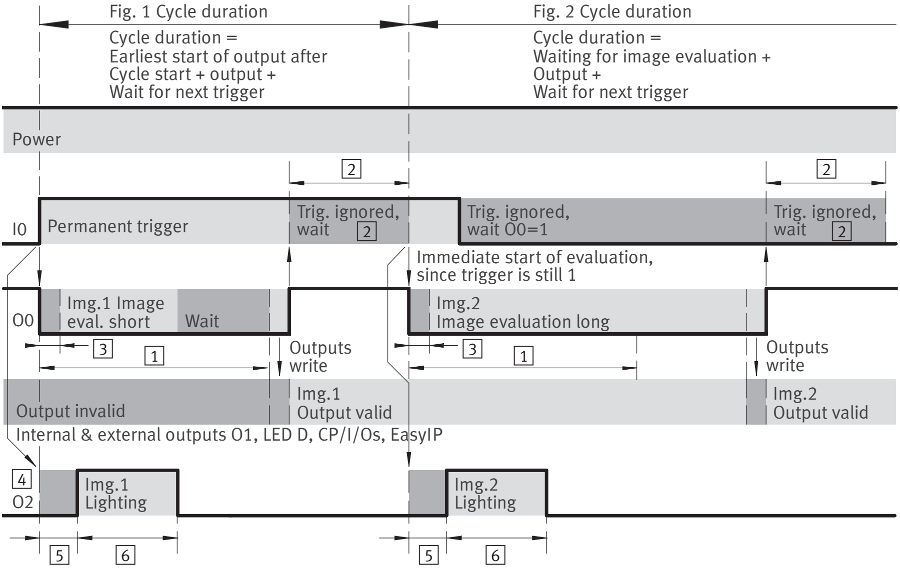


Fig. 4/9: "Free run" evaluation mode – Signal sequence using system parameters

Settings of the system parameters:

1 Earliest start of output after cycle start = 220 ms

2 Ignore trigger signal after ready for operation = 65 ms

3 Start of image evaluation after cycle start = 20 ms

4 Function at O2 = external lighting, start of lighting and duration of lighting = manual

5 Start of lighting after trigger / cycle start = 40 ms

6 Duration of lighting = 100 ms

– Reset outputs at outputs after "Ignore triggering signal" = Off

– Reset outputs at outputs during Analysis = No

Simplified depiction of the signal curves; these do not contain jitter, run times or system-related delay times.
The duration for the write process of the outputs and thus the duration for an invalid output condition is indefinite with use of external outputs (I/O expansion, device as CPI module, EasyIP or CheckKon). This duration is approximately 1 ms if the external outputs are deactivated by system parameters and only the internal outputs are used.

"Free run" evaluation mode with image-based trigger (only SBO...-Q-R...B)

The "Free run" evaluation mode also offers the possibility to set up a image-based trigger.
This combination is only available in types SBO...-Q-R...B (monochrome image sensor).

The function of the "Free run" evaluation mode with image-based trigger is similar to that of the "Triggered" evaluation mode. However, the trigger signal is not tripped from outside, but depends on the content of the current camera image.

Procedure

As long as the Free run mode with image-based trigger is activated (e.g over the input I0 "Trigger signal" = logic 1), the device permanently takes camera images.
With each camera image, a check is made whether conditions of the image-based trigger are fulfilled:

Condition not fulfilled

The camera image is deleted without further analysis (that is, without pre-processing, feature determination, quality decision, output, etc.), and the next camera image is taken immediately. As a result, camera images can be checked in very fast sequence for fulfillment of the conditions of the image-based trigger.

Condition fulfilled

The camera image is evaluated, taking into account all system parameters of the Free run mode:

– Start of image evaluation after cycle start

– Earliest start of output after cycle start

After output of the results of the check, camera images are taken again until the conditions of the image-based trigger are met again.

The system parameter "Start of image evaluation after cycle start" influences the frame rate for analysis of the camera images of the image-based trigger.

- For quick reaction of the image-based trigger, put the setting of this system parameter to "0".

The process after a fulfilled trigger condition depends on the system parameter "Image after fulfilled trigger condition":

Setting "Ongoing"

As long as the condition of the image-based trigger is fulfilled, the images are evaluated and the results output.

Setting "Single image"

After successful evaluation of a image and output of the corresponding results, camera images are taken again. But these images are evaluated only when the conditions of the image-based trigger are **not** fulfilled for at least 1 image. Only then will just the first image that fulfills the conditions of the trigger be evaluated.

The "Single image" setting ensures that the results for a part are only output once, even if the part remains in front of the Compact Vision System for an extended period and thus again fulfills the conditions for the image-based trigger.

Configuration of the image-based trigger

Requirements:

– Device is in the “Free run” evaluation mode

– “Image-based trigger” system parameter is switched on.

The following image shows the “Live image” CheckKon window, in which you can make additional settings for the image-based trigger.

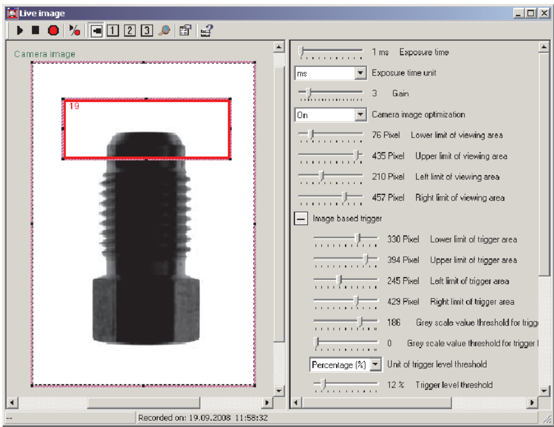


Fig. 4/10: “Live image” CheckKon window

Trigger field area

Evaluation of the image-based trigger is made only for one image area within the field of view of the camera image. This so-called “trigger field area” can be set easily in the “Live image” window (similar to the field of view).

- Move the mouse to the node points of the “trigger field area”. With pressed (left) mouse button, change the position and size of the frame.

- Alternatively, actuate the slide control to set the upper, lower, left and right trigger field limit.

Note

The frame rate (images per second) for examination of the image-based trigger depends on the exposure time, the system parameter “Start of image evaluation after cycle start” and the size of the field of view. The time period between two examinations must particularly be observed when the part to be checked is moved.

- Select the size of the trigger field area so the trigger will be safely tripped.

The parameter “Image taking duration after triggering” informs you of the time period between the trigger signal and the moment when the image is available in the processor. You determine the image taking frequency with this parameter while the image-based trigger is still invalid.

All pixels are analysed within the trigger field area.
The number of pixels is determined whose brightness value lies between the following system parameters.

- “Grey value threshold for trigger dark”
- “Grey value threshold for trigger light”

Conditions of the image-based trigger

The decision to trigger the trigger signal is made dependent on the following conditions (evaluation of the image).

- “Unit of the trigger threshold” = “pixels” or “percent”
- “Trigger threshold” = limit value in the unit of the trigger threshold
- “Initiate trigger” with “rising edge” or “falling edge”.

Example

The example in Fig. 4/10 shows a screw located in the trigger field area.

The number of pixels is determined whose brightness lies between the following system parameters:

– 0 (black) = “Grey value threshold for trigger dark”

– 90 (dark grey) = “Grey value threshold for trigger light”

The number of these pixels corresponds here to a share of 16 % of the trigger field area. This value is displayed in the “Live image” window.

The following conditions must be fulfilled so that the image-based trigger is tripped:

– Trigger threshold = 10 %

– Initiate trigger with “rising edge”

The specified conditions are fulfilled, so the image-based trigger is tripped. In the “Live image” window, this is shown by a red border around the trigger field area.

When the specified conditions of the image-based trigger are not fulfilled, this is shown with a blue border in the trigger field area.

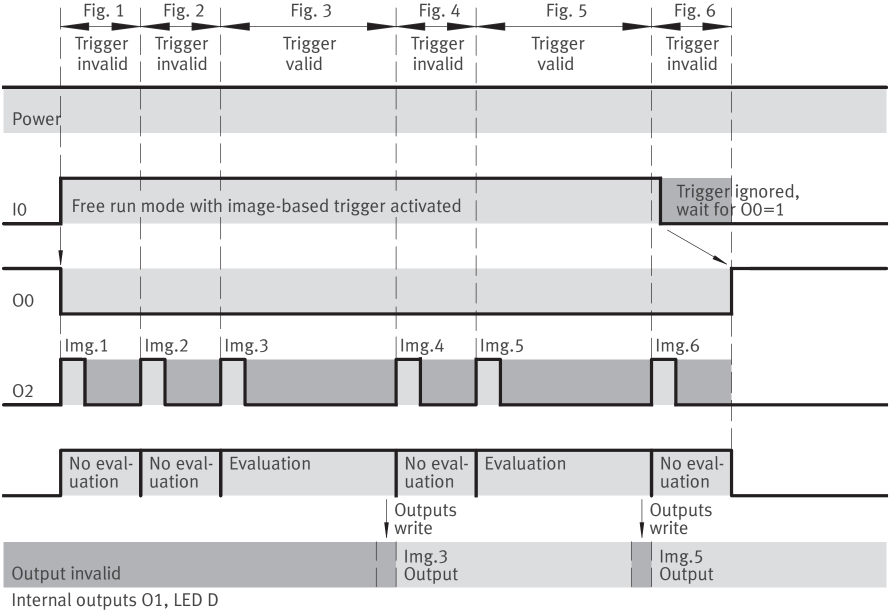


Fig. 4/11: Free run mode with image-based trigger

Settings of the system parameters (Standard values):

- Earliest start of output after cycle start = 0 ms
- Ignore trigger signal after readiness for operation = 0 ms
- Start of image evaluation after cycle start = 0 ms
- Function at O2 = external lighting
- Image after fulfilled trigger condition = single image
- “Free run mode” evaluation mode Trigger signal evaluation = level-controlled
- Image-based trigger = On
- No external inputs (CAN bus and EasyIP = deactivated)
- No external outputs (CAN bus and EasyIP = deactivated)

4.9.3 I/O sequence with the “Fixed frame rate” evaluation mode

This evaluation mode is not available in SBO...-Q-R3.

The “Fixed frame rate” evaluation mode remains active as long as logic 1 (condition-controlled) is present at the “Trigger signal” input.

– During this time, images are generated in a specified cycle.

– During the evaluation, the “Ready for operation” output shows logic 0.

Image creation and lighting are started with the cycle start. Only the duration of the lighting can be influenced via system parameters.

Evaluation and output of the results are temporally decoupled from each other.

– The images are generated in a fixed time cycle.

– The time until output of the related images is postponed depending on the evaluation duration of the current image and, if necessary, of the previous images as well.

A rising edge at one of the outputs signals that an evaluation has been completed.

• Set the system parameter “Reset outputs at outputs during evaluation” = Yes.

If the required evaluation duration is longer than the established frame rate, images are stored temporarily in the device. The output time can be postponed by several cycles. If temporary storing is no longer possible, the device enters a fault status or emits a warning (configurable via “Image buffer overflow” system parameter). Temporarily stored images are lost thereby.

This evaluation mode is especially suitable for time-critical inspections in which the inspection part cannot be stopped or for which a constant inspection rate is needed – as with continuous material inspection.

Intermediate storage of the images should only be used to compensate for fluctuations in the evaluation duration. A cycle duration that is longer than the average evaluation duration should not be selected.

If the “Fixed frame rate” evaluation mode is stopped (“Trigger signal” input = logic 0), all temporarily stored images will be completely evaluated, if necessary, and the inspection result output – only then does the “Ready for operation” output switch to logic 1.

The outputs of the output module of the I/O extension O0 to O3, like the internal outputs, are configured via the system parameters, that is, they are set or reset together with the internal outputs.

The capacity of the buffer can be changed via the system parameter “Buffer size (number of full images)”. The buffer size is specified as a multiple of the sensor resolution of the respective Compact Vision System.
The buffer is created when the Compact Vision System restarts, i.e. changes to this system parameter do not take effect immediately.

Note
The buffer created via the system parameter “Buffer size (number of full images)” reduces the available memory of the Compact Vision System. Hence it only makes sense to select a value greater than 2 in “Fixed frame rate” evaluation mode, because only this mode requires a buffer accommodating several images.

Signal sequence with standard settings

The signal sequence with standard settings of the system parameters in the “Fixed frame rate” evaluation mode could appear as follows.

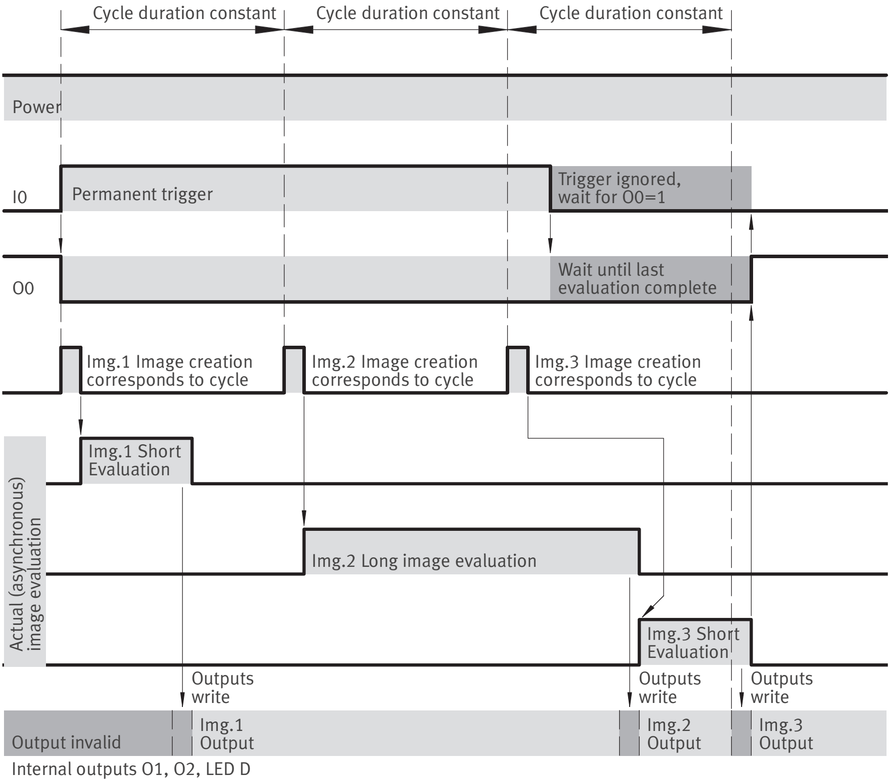


Fig. 4/12: “Fixed frame rate” evaluation mode – signal curve with standard settings

Settings of the system parameters (Standard values):

- Earliest start of output after cycle start = 0 ms
- Function at O2 = reject part
- Start of lighting = automatic
- Duration of lighting = automatic
- Cycle duration = 220 ms
- Reset outputs at outputs during Analysis = No

Signal curve using system parameters

The signal curve with use of system parameters in the “Fixed frame rate” evaluation mode could appear as follows.

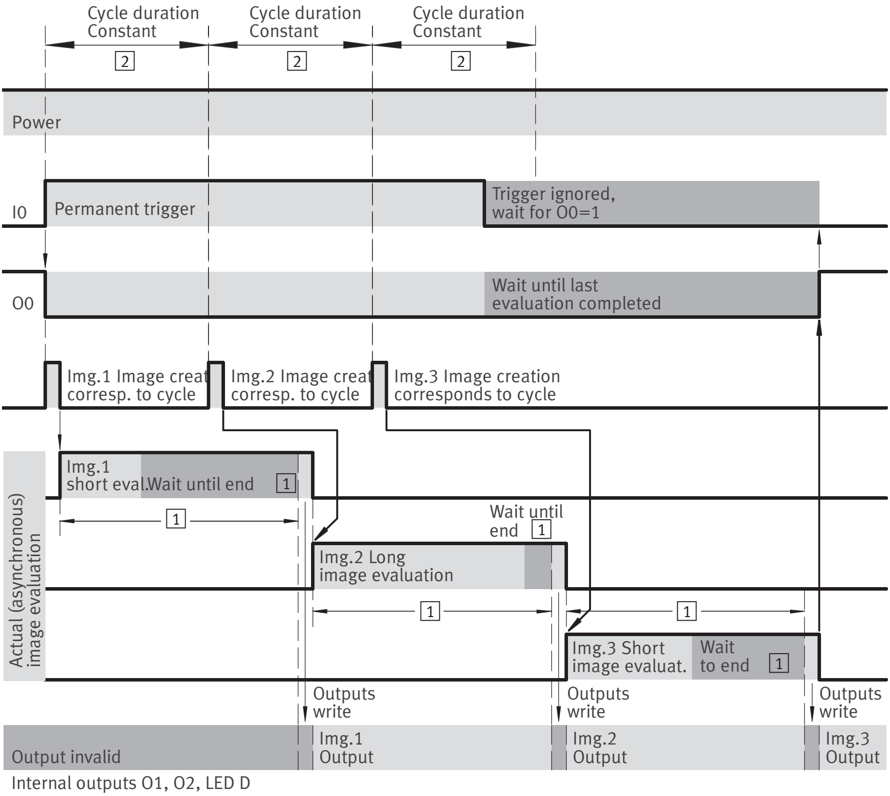


Fig. 4/13: “Fixed frame rate” evaluation mode – signal curve using system parameters

Settings of the system parameters:

1 Earliest start of output after cycle start = 300 ms

2 Cycle duration = 220 ms

Negative example

Earliest start of output after cycle start as selected is too large – an error was triggered after N images (➔ Settings of the system parameters).

Simplified depiction of the signal curves; these do not contain jitter, run times or system-related delay times.

The duration for the write process of the outputs and thus the duration for an invalid output condition is indefinite with use of external outputs (I/O expansion, device as CPI module, EasyIP or CheckKon). This duration is approximately 1 ms if the external outputs are deactivated by system parameters and only the internal outputs are used.

4.10 Connection to higher-level controller (PLC/IPC)

For controlling the device or processing the check results, a higher-level controller (PLC) can be connected to the Compact Vision System according to the connection possibilities available.

The following connection possibilities are available, depending on the device and firmware used:

Connection	Inputs	Outputs
Device-internal I/Os at “Operating voltage supply and digital I/Os” plug	– Trigger signal – Apply Inputs signal / Acknowledge Error signal – Input value for check program	– Ready for operation – Wide range of configuration options – Check program output
I/O expansion over CAN interface [1]	– Preselection check program – Input value for check program	– Recognized parts type – Check program output
Device acts as CPI module via CAN interface [1]	– Trigger signal – Apply Inputs signal – Acknowledge Error signal – Preselection check program	– Ready for operation – Fundamental inspection results with recognised parts type – Warning and fault status
Communication over Ethernet interface – with EasyIP protocol – with Telnet protocol – with Modbus protocol – with EtherNet/IP protocol	– Trigger signal – Apply Inputs signal – Acknowledge Error signal – Preselection check program – System parameters – Tolerances check program	– Ready for operation – Warning and fault status – Detailed inspection results with recognised parts type and features
Communication with CoDeSys via Ethernet interface	– Trigger signal – Apply Inputs signal – Acknowledge Error signal – Preselection check program – System parameters – Tolerances check program	– Ready for operation – Warning and fault status – Detailed inspection results with recognised parts type and features
[1] Not SBO...-Q-...-WB		

Tab. 4/5: Connection possibilities for the Compact Vision System

Additional connection possibilities on request.

The connection possibility to be used depends on the required I/O functions of the application.

Device-internal I/Os in combination with a powerful higher-level controller (PLC) are preferred with automatic parts flow, that is, a fast inspection rate, since here the lowest delay times are expected.

A higher-level controller must be programmed according to the signal behaviour of the selected evaluation mode. Note the descriptions of the signal processes starting in chapter 4.9.1 ff.

4.10.1 General information on use of inputs

There are inputs with signal function and other inputs.

Inputs with signal function

These inputs are edge- or condition-controlled, depending on the evaluation mode. They are constantly read. If the signals regarding the current operating state are valid, the corresponding action is carried out immediately. Inputs with signal function are, for example:

– Trigger signal

– Apply Inputs signal

– Acknowledge Error signal

Other inputs

These inputs are condition-controlled, that is, they react to logic 1 or logic 0. They are read only after a valid “Apply Inputs signal”.

Example for “normal input”: Preselection check program.

i

Regardless of the function selected, the internal inputs and the I/O expansion inputs can be read via the communication protocols EasyIP, Telnet, Modbus and EtherNet/IP, or in the check program.

Input I1 has a double function and can be mapped internally to two different offset addresses:

– Apply Inputs signal FW 1

– Acknowledge Error signal FW 2

For reading input I1, flag word 1 or flag word 2 can be used (➔ chapter A.6.1)

Note
Reading of input statuses takes place while the check program is running, following image capture.

The time of reading during image evaluation is however not precisely defined, i.e. changes to input statuses during the evaluation period can lead to unexpected results.

- Ensure that **no** changes are made to input statuses during evaluation.

"Trigger signal" input

The "Trigger signal" input starts and stops the inspection process. Mode of operation, signal detection and signal sequence depend on the evaluation mode selected (➔ chapter 4.9.1 ff.).

"Apply Inputs signal" input

The "Apply Inputs signal" is used to load a new check program. The number of the new check program must be specified previously via the I/O possibilities. The "Apply Inputs signal" is edge-controlled detected and only accepted if logic 0 changes to logic 1 at the "Ready for operation" output. And so the "Apply Inputs signal" must first be reset to read inputs again (load check program).

As long as the inputs are read and the check program is loaded, the "Ready for operation" output emits logic 0. During this time, no parts can be inspected; (The trigger is not valid as long as the "Ready for operation" output emits logic 0).

As soon as the check program has been loaded, the "Ready for operation" output emits logic 1 again. Trigger signals for an inspection are now accepted.

Note
If a check program has been loaded, the check results and the associated flag words (FW) and strings (STR) are reset.

- All feature values = invalid
- All Invalid flags = set
- Deviation = 999
- Acceptance of the new feature names
- Acceptance of the new name of the check program.

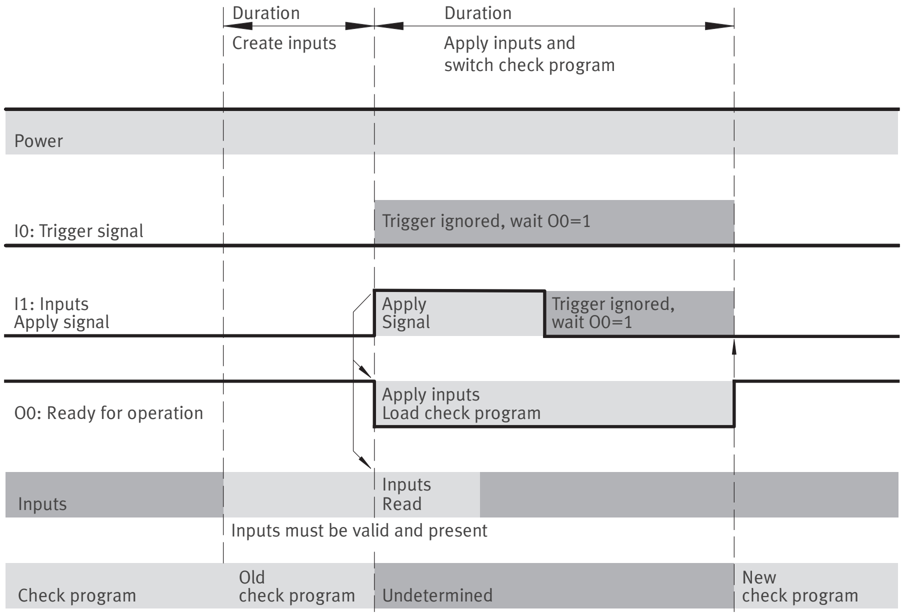


Fig. 4/14: Signal curve: Apply inputs – Switch check program

Simplified presentation. The signal curves do not contain jitter, run times or system-related delay times.

The inputs must be valid and present for a certain time (min. 30 ms), depending on the I/O possibilities used.

“Acknowledge Error signal” input

If an error status occurs in the device, logic 1 is displayed at the “Error status” output. Additionally, the “Ready for operation” output switches to logic 0.

Incoming trigger signals are not valid, that is, no more evaluations are possible until the error has been corrected.

The error must be corrected through appropriate measures while the system remains in error status. Diagnosis can be conducted and system parameters changed during this time.

- For this, use CheckKon or all protocols that can access system parameters.

As soon as the error has been corrected, this must be notified to the Compact Vision System via the “Acknowledge Error signal” input. The “Acknowledge Error signal” is edge-controlled detected.

If a warning condition is signaled (e.g. at the “Warning” output), this does not need to be acknowledged with the Acknowledge Error signal. The warning condition is automatically lifted when the cause is eliminated.

Information on the errors (description and recommendations for remedy) can be found in chapter 5.1.2.

4.10.2 General information on use of outputs

Writing of the outputs takes different amounts of time, depending on the connection. For time-critical applications, only the device-internal outputs should be used and other connection possibilities deactivated.

4.10.3 Use of internal I/Os

The functions and the signal sequence for the device-internal I/Os therefore depend on which evaluation mode the device is in (➔ chapter 4.9).

The function of selected I/Os can be established via system parameters. This makes a flexible adaptation to the application's requirements possible.

The following functions are available at the I/Os:

I/O	Configurable functions
I0	– Trigger signal (standard) – Polarity (rising/falling edge or logic 1 / logic 0) can be changed via system parameters – Deactivated for use under CoDeSys embedded and/or check program
I1	– Apply Inputs signal and in case of error: Acknowledge Error signal – Polarity (rising/falling edge or logic 1 / logic 0) can be changed via system parameters. – Deactivated for use under CoDeSys embedded and/or check program

I/O	Configurable functions
O0	– Ready for operation (standard value) – Output Good part – Output Reject part – Output correctly oriented – Output incorrectly oriented – Warning – Error – CoDeSys – Check program
O1	– Output Good part (standard value) – Output Reject part – Output correctly oriented – Output incorrectly oriented – Warning – Error – CoDeSys – Check program
O2	– Output Reject part (standard value) – Output Good part – Output correctly oriented – Output incorrectly oriented – Warning – Error – External lighting – CoDeSys – Check program

Tab. 4/6: Functions at internal I/Os

4.10.4 Use of the I/O expansion

This I/O expansion is not available in SBO...-Q-... -WB.

If the I/O expansion is used, selected modules can be connected to the CAN interface of the Compact Vision System. This connection possibility is used to expand the internal I/Os.

Use an I/O expansion:

– to select check programs via digital inputs,

– for reading into the check program from external digital inputs,

– for signalling recognised parts types via digital outputs,

– for signalling results of individual check features via external digital outputs.

Notes on installation

Only the following I/O modules are permissible.

Output module types	**Input module types**
– CP-A04-M12-CL	– CP-E08-M12-CL
Quantity: max. 1 module	Quantity: max. 1 module

I/O modules require operating voltage and load voltage supplies. The Compact Vision System does not provide any operating voltage and load voltage supply for external I/O modules.

- To provide power to the external I/O modules, first connect cable SBOA-K20CP-SUP from Festo to the Compact Vision System (➔ Fig. 4/15).

- Then connect the I/O modules as depicted in the example Fig. 4/15. To do this, use connecting cable KVI-CP-3. Please note that the total cable length must not exceed 10 m.

1 Compact Vision System SBO...-Q

2 Cable for power input SBOA-K20CP-SUP

3 Connecting cable KVI-CP-3

4 Input module CP-E08-M12-CL (max. 1)

5 Output module CP-A04-M12-CL (max. 1)

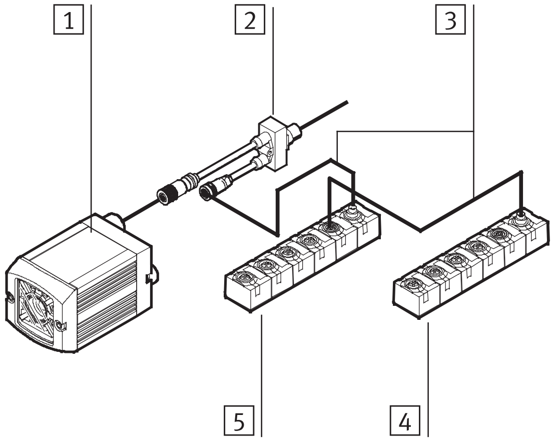


Fig. 4/15: Compact Vision System SBOI-Q with I/O expansion (example).

- Alternatively, connect the modules in the following sequence:
  - Compact Vision System – output module
  - Compact Vision System – input module
  - Compact Vision System – output module – input module

The 0 V line of the cable SBOA-K20CP-SUP 2 is connected electrically with the 0 V line of the Compact Vision System 1.

- Use suitable measures in order to avoid equalizing currents, e.g. a common power unit for the compact camera system and the power supply for the I/O modules, or by separate low-impedance potential equalization.

Commissioning:

The nominal configuration of the I/O expansion is specified by the system parameter "CAN interface". When the power supply is switched on and during operation, the device checks whether the line allocation corresponds to that specified by the system parameters.

1. Make a connection to the device with the CheckKon program and set the system parameter "Function at CAN interface" to "Off".
   End the CheckKon program.

2. Switch the system off in order to avoid faults or damage. Disconnect the power supply for the device and the modules.

3. Wire the device and the modules.

4. Connect the power supply for the device and the modules.

5. Make a connection to the device with the CheckKon program and set the system parameter "Function at CAN interface" to "I/O expansion".

6. Adjust the system parameters under "Configuration I/O system extension" according to their configuration.

7. Confirm at the device any displayed errors of the I/O expansion with the "Acknowledge Error signal" at input I1 or in CheckKon in the "System status" window.

Depending on the system parameters, the device displays a warning or error if:

- The current I/O expansion does not correspond to the nominal configuration of the I/O expansion (also note the sequence of the modules)

- Modules Report errors (e.g. overload).

The assignment of the functions to the individual I/Os of the modules can be configured via system parameters.

Input module

Inputs 0 ... 7 of the module can be read in at any time via the communication protocols Telnet, EasyIP, Modbus and also in the check program ("I/O access" tool), provided the following conditions are met:

- an input module is connected,
- the system parameter "Function at CAN interface" is configured to "I/O expansion",
- the system parameter "Check program preselection" is configured to "I/O possibilities".

Input format for the input module	
Binary	Inputs 0 ... 7 of the module are interpreted as byte value (standard setting). The byte corresponds to the check program preselection (0–255): "0" = check program 1.
1 of N	Inputs 0 ... 7 of the module are interpreted as individual bits and correspond to the check program preselection (0 ... 7): "Bit 0" = check program 1.

The input format for the input module can be specified via system parameters.

Output module

Output format for the output module	
Recognised parts types as binary value	Outputs 0 ... 3 of the module are interpreted as byte value (standard setting). This corresponds to the recognised parts type (0..15): "0" = parts type 1.
Recognised parts types as 1 of N value	Outputs 0 ... 3 of the module are interpreted as individual bits and correspond to the recognised parts type (0 ... 3): "Bit 0" = parts type 1.
Check program output	The functions of outputs 0 ... 3 of the module are defined via the check program, "I/O access" tool.

The output format for the output module is defined via system parameters.

Further information on the process and signal sequence can be found in chapter 4.9 ff.

Further information on the modules can be found in the P.BE-CP-EA-CL manuals.

Use of the PLC run-time system "CoDeSys embedded" also allows use of an I/O expansion. The following I/O modules are permissible:

- CP-A04-M12-CL
- CP-E08-M12-CL
- CP-E08-M8-CL
- CP-E16-KL-CL

• Make sure that the system parameter "Function at CAN interface" has been configured to the value "Off (CoDeSys)".

- Please note the following specifications (➔ Online Help for the Target Support Package):
  - Max. 4 modules in any order
  - Max. 32 inputs and 32 outputs
  - Communication with the I/O modules takes place via the PLC run-time system.

4.10.5 Use of the device as CPI module at CP nodes

If the firmware and hardware of the device support CPI module functionality, this can be activated via a system parameter (not applicable to SBO...-Q-... -WB).

If use of the device as a CPI module is activated, the device corresponds to a CP module with extended functions (CPI module) in the related CP line. This allows it e.g. to be operated on a CPX-CP interface of a CPX terminal.

Notes on installation

To connect the device to a CP string, use a suitable cable like cable SBOA-K20CP-WS from Festo.

The device does not have any ongoing CP connection and can only be connected at the end of a CP line.

The 0 V line of the Compact Vision System is connected electrically to the 0 V line of the CPX terminal.

- Use suitable measures in order to avoid equalizing currents, e.g. a common power unit for the Compact Vision System and the CPX terminal, or by separate low-impedance potential equalization.

Commissioning:

Note
In order for the Compact Vision System to be recognised in the CP line when the power supply for the CP master is switched on (e.g. of a CPX-CP interface), the Compact Vision System must already be ready for operation.

- Switch on the power supply for the Compact Vision System at least 15 seconds before the power supply for the CP master.

1. Make a connection to the device with the CheckKon program and set the system parameter “Function at CAN interface” to “Off”.
   End the CheckKon program.

2. Switch the system off in order to avoid faults or damage. Disconnect the power supply for the device and the CPX terminal.

3. Attach the Compact Vision System to the desired CP line.

4. Connect the power supply for the device.

5. Make a connection to the device with the CheckKon program and set the system parameter “Function at CAN interface” to “CPI module”.

6. Set the system parameter “SBO...-Q connected to line X2 or X4” to “No” if the Compact Vision System is connected to the CP line X1 or X3. If connected to the CP line x2 or x4, the system parameter must be set to “Yes”.

7. First connect the power supply to the CPX-CP terminal.

8. Actuate the Save button on the CPX-CP terminal to save the new line assignment.

9. Restart the CPX-CP terminal. Briefly disconnect the power supply to the CPX-CP terminal.

10. Confirm at the device any displayed faults of the CPI module function with the “Acknowledge Error signal” at input I1 or in CheckKon in the “System status” window.

A line assignment together with other modules could look as follows:

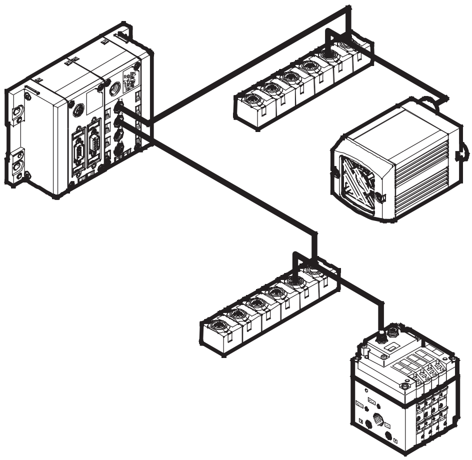


Fig. 4/16: Example line assignment: Compact Vision System SBOI-Q as CPI module

The Compact Vision System can be addressed via a higher-level controller that either:

– is integrated in the CPX terminal (e.g. CPX-FEC -... version R5 or higher)

or

– is connected to the higher-level field bus.
For this, the related CPX terminal must also be connected to the same fieldbus (e.g. to Profibus via the CPX module "CPX-FB13-...", version R12 or higher).

Additional information on CP and CPX can be found in the manual "System Description CPI" (P.BE-CPX-CP).
Further information on fieldbus nodes can be found in the "Electronics Manual" (P.BE-CPX-FB).

The device corresponds to a CPI module and always occupies 16 inputs and 16 outputs at CP masters (with and without expanded functions).

The following table shows an overview of the addresses assigned for the CP input and output module.

CP modules			**Assigned I/Os**			
Type	**Type**	**Module supports extended functions**	**on CP masters with extended functions**		**on CP masters without extended functions**	
			I	**O**	**I**	**O**
Compact Vision System acts as CP input and output module	SBO	Yes	16	16	16	16

Tab. 4/7: Assigned I/Os for the CP input and output module

Note
The input and output numbers specified here must be converted according to the above address assignment for the corresponding CP line number and line assignment.

6 CP inputs (from the viewpoint of the device, these are outputs)

Input	Function
Bit I0	Ready for operation
Bit I1	Output result good
Bit I2	Output result reject
Bit I3	Output result correctly oriented
Bit I4	Output result incorrectly oriented
Bit I5	Unused
Bit I6	Warning (corresponds to "LED C" red & flashing)
Bit I7	Fault status (corresponds to "LED C" red)
Bit I8 ... 11	(counts as a byte) Recognised parts type (0..15): "0" = Parts type 1
Bit I12 ... 15	Unused

Tab. 4/8: CP inputs

16 CP outputs (from the viewpoint of the device, these are inputs)

Output	Function
Bit O0	Trigger signal
Bit O1	Apply Inputs signal
Bit O2	Acknowledge Error signal

Output	Function
Bit O3 ... 7	Unused
Bit O8 ... 15	Preselection check program (is read after bit O1=1)

Tab. 4/9: CP outputs

Assignment of the device's addresses depends on:

a) the CPX fieldbus node used or CPX-FEC

b) the line number used

c) the CP modules used in the line in front of the device.

Example:

The device is operated on a CPX-CP interface with a CPX-FB13 Profibus fieldbus node. The device is located in the 1st line in position 2 after a CP-A04-M12-CL output module (➔ Fig. 4/16).

Address assignment for the CPX-CP interface:

Inputs (string 1: I0 ... I31):

– I0 ... I15 is assigned to the device's 16 inputs

– I16 ... I31 free

Outputs (string 1: O0 ... O31).:

– O0 ... O7 is assigned to CP-A04-M12 -CL (8 assigned outputs, of which 4 are used)

– O8 ... O23 is assigned to the device's 16 outputs

– O24 ... O31 free

The address O8 thus corresponds to the trigger signal.

Information on the process and signal sequence can be found in chapter 4.9 ff.

CPX-FEC controller: The device can be controlled via the CPI module function of a CPX-FEC. Corresponding system programs can be transferred to the (CPX -FEC) controller using the Festo FST programming software.

Note

- Make sure that the system is in good working order and can run:

The current line configuration must already be stored.

After transfer of a new system program, the controller must be restarted to initialise it.

- Briefly disconnect the power supply to the CPX-FEC.

4.10.6 Use of the device as CANopen Master

When the PLC run-time system “CoDeSys embedded” is used, the Compact Vision System can be used as CANopen master. Any CANopen slaves can be connected to the CAN interface, e.g. servo controller or CPX-FB14. In this way, complete production processes can be directly controlled via the device. (Not available in SBO...-Q-...-WB).

- Make sure that the system parameter “Function at CAN interface” is set to the value “Off (CoDeSys)”.
- Use a twisted, screened 4-core cable as CAN bus line.

Communication with CANopen slaves over the CAN interface takes place with the PLC run-time system “CoDeSys embedded”.

The connected CAN bus slaves are not supplied with power over the CANopen interface.

Commissioning:

If the Compact Vision System to be connected is located at the end of the fieldbus, you need a terminating resistor.

- Connect the terminating resistor (120 Ω, 0.25 W) to the fieldbus socket between the contacts for CAN_H (pin 4) and CAN_L (pin 5) (➔ chapter 3.2.3).
- Use a screened plug connector which will guarantee continuous contact between the screening/shield and the Compact Vision System.
- Connect the screening of the CAN cable with low impedance to earth potential.

➔

Note
Data transmission errors can arise due to signal reflections and signal attenuations.

- Avoid possible causes:
  - missing or incorrect terminating resistor
  - faulty screened connection
  - branches
  - large distances
  - inappropriate cables.

4.10.7 Use of the Ethernet interface with EasyIP

For data transmission and control, the Compact Vision Systems make the Festo EasyIP protocol available. This makes communication possible with the following components:

- Festo controllers (e.g. CPX-FEC) with EasyIP support
- Festo FED with Ethernet connection
- Festo OPC Server

This allows very comprehensive output and control possibilities, such as for further processing of inspection results in a higher-level controller.

Support for the EasyIP protocol by the Compact Vision System must be enabled via the system parameters.

For communication via Ethernet, FEDs generally require a bus interface (Ethernet interface). Further information on FEDs (and if necessary bus interfaces) can be found in the descriptions for the product concerned.

Reading and writing take place via the data packages defined by EasyIP, whereby the data to be read/written are defined through memory addresses.

The Compact Vision System not only makes I/Os available via memory addresses, but also allows access to the results of a check and to the settings of the system parameters.

Certain memory addresses can be both read and written to, while others can only be read or only written to (➔ tables in appendix A.6).

Commissioning:

1. Switch the system off in order to avoid faults or damage. Disconnect the power supply to the Compact Vision System and to the device to be coupled (e.g. CPX-FEC or FED).

2. Using the specified cables, connect the Compact Vision System to the device to be connected, e.g. via an Ethernet switch or hub.

3. Reconnect the power supply.

4. Make a connection to the device with the CheckKon program and set the system parameter “EasyIP Server” to “On”.

5. Set the IP address of the Compact Vision System at the device to be connected.

6. Program the device to be connected to enable it to access Compact Vision System data.

General information on inputs can be found in chapter 4.10.1. The available memory addresses and their function can be found in appendix A.6.

4.10.8 Use of the Ethernet interface with Telnet

The device makes available the Telnet protocol for communication with a higher-level controller, robot or PC. This permits very wide-ranging output and control possibilities. Inspection results can be processed further in the higher-level controller, in order to grab a part with a robot, for example.

Support for the Telnet function by the Compact Vision System must be enabled via the system parameters. Reading and writing take place via a text-based command line with defined commands, whereby the data to be read/written are defined through memory addresses.

The device not only makes simple commands available for image capture, but also allows access to the results of a check and the system parameter settings, among other things.

Certain memory addresses can be both read and written to, while others can only be read or only written to (➔ tables in appendix A.6).

Commissioning:

1. Switch the system off in order to avoid faults or damage. Disconnect the power supply to the Compact Vision System and the device to be coupled.

2. Using the specified cables, connect the Compact Vision System to the device to be connected, e.g. via an Ethernet switch or hub.

3. Reconnect the power supply.

4. Make a connection to the Compact Vision System with the CheckKon program and set the system parameters under "Telnet function":

   – The system parameter "Authentication required" determines whether the device to be coupled has to identify itself (password).

   – The system parameter "TCP port" determines which port should be used for the Telnet communication.

   – The system parameter "Telnet server" activates the Telnet function and establishes additional protocol characteristics:
     – "On" (normal function): The device to be coupled does not use the Telnet S7 building block
     – "On" (S7 SBOxQ building block) if you use the Telnet S7 SBOx-Q building block on a corresponding controller.

5. Specify the IP address and the Telnet port of the Compact Vision System at the device to be connected.

6. Program the device to be connected to enable it to access Compact Vision System data.

Please consult your local Festo service when using the S7 SBOxQ building block.
General information on inputs can be found in chapter 4.10.1.
The available memory addresses and their function can be found in appendix A.6.

Test of Telnet communication

Most PC operating systems have a Telnet program at command line level. With this program, you can use the Telnet connection to the Compact Vision System.

Windows operating system

Requirement in operating systems from Windows Vista ™

- Activate the option "Telnet Client".
  This setting can be found in "Windows functions ..." under "Programs and Functions" in the system control.

- Open a window with command line via the Windows Start Menu [Programs] [Accessories] [Entry].

- Start the Telnet program and pass on the IP address of the Compact Vision System as well as the port number determined in the "TCP port" system parameter.

Fig. 4/17: Calling up Telnet program

This builds up a connection to the Compact Vision System.

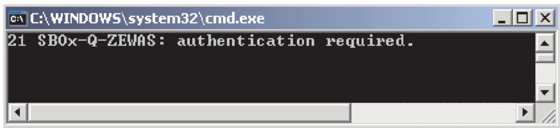


Fig. 4/18: Telnet program with SBO...-Q acknowledgement (here with authentication)

The Compact Vision System now awaits authentication, if necessary.

If your entries in Telnet are not visible (e.g. with Telnet in Windows 2000), you must activate the “Local Echo” in Telnet. Information on this can be found in the Help for the Telnet program.

Authentication

If authentication has been activated through the “Authentication required” system parameter, users must identify themselves after the connection is made.

This occurs with the following commands:

– USER ‹User name›

– PASS ‹Password›

The user “root” is set up on the Compact Vision System.

The related password in the delivery condition of the Compact Vision System is “Festo”.

Additional users are not supported.

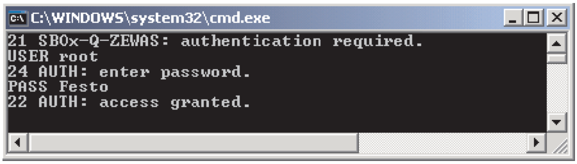


Fig. 4/19: Authentication in type SBO...-Q

If the combination of user name and password is valid, the message “22 AUTH: access granted” appears.

– After this message, all control and data commands are valid for the duration of the connection.

If the user name or password is invalid, the message “23 AUTH: access denied” appears.

– The Compact Vision System accepts a new authentication attempt only after a safety time of 2 seconds.

Before successful authentication, all control and data commands are rejected (exception: “exit”).

The Telnet function of the Compact Vision System supports up to 10 active connections. If the connection is interrupted, this enables the connection to be restored quickly.

If you attempt to make more than 10 connections to the Compact Vision System, the following error message appears:

```
-6 ERROR: maximum number of connections reached
```

After this message, the connection is ended.

Telnet commands

WFW – Write FlagWord

WFW ‹address, value›
Writes a value to the flag word address specified (➔ appendix A.6).

The following example writes the value 5 to the flagword address 33 (fast access to check program preselection).

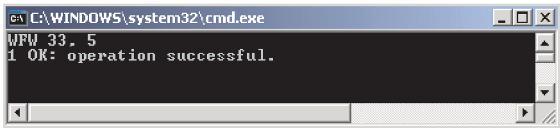


Fig. 4/20: Writing to a flagword address

In case of fault, the response is with an error code and error message.

> **Note**
> - Use only integers when writing values on flagword addresses.

RFW – Read FlagWord

RFW ‹address›
Outputs the current value of the specified flag word address as text (➔ appendix A.6).

With this command, up to 64 addresses can be read out. Here the flag word addresses should be separated by commas:
RFW ‹address1, address2, address3, ...› .

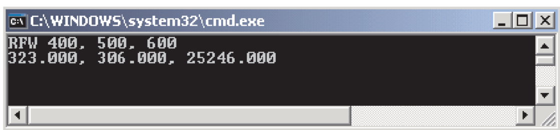


Fig. 4/21: Reading out flag word addresses

RNV – Read Named Value

RNV [name of feature]
RNV “Name of feature”
Gives the current feature value of the feature specified
(➔ appendix A.6).

With this command, up to 64 feature values can be read out. Here the feature names should each be written in square brackets or quotation marks, and separated by commas:
RNV [name 1], [name 2], [name 3], ...

Note
Values are output only when the feature name(s) has/have been completely and correctly written.

- Do **not** use any of the following characters for feature names:
  – square brackets [ ]
  – quotation marks “ ”
  – umlauts
  – special characters.
- Pay attention to upper case and lower case letters in feature names.

If no match is found, the device puts out the following message:

```
-55 ERROR: one or more feature name(s) not valid
```

The feature names are stored in the check programs and can be established when creating the check program (in CheckOpti).

- Preferably use meaningful feature names.

If several feature names are identical, the value of the first feature whose name matches is output.

RSTR – Read String

RSTR ‹address›
Outputs the current character string of a string address (➔ appendix A.6.11). Only one string address can be read out per command. Simultaneous reading of multiple character strings is not possible.

Commas, apostrophes and control characters contained in the strings are replaced by "_" characters.

RDO – Read Data Output

RDO [name of data output]
RDO "name of data output"
Outputs the values of the respective data output.

➔

Note
Data outputs can only be requested if they have been created previously in CheckOpti in the selected check program with the corresponding name.

If no data output has been created, an error message appears.

Details of configuration of a data output can be found in the CheckOpti online Help.

Values are output only if the data output name has been entered completely and correctly.

- Do **not** use any of the following characters in data output names:
  – square brackets [ ]
  – quotation marks " "
  – umlauts
  – special characters.
- Use correct capitalisation in data output names.

If the data output exists and uses the Telnet protocol, you will receive the result in accordance with the output format:

- Telnet – SBO...-Q Part Detector: binary coded
- Telnet – SBO...-Q Data Collection: binary coded
- Telnet – XML: output in the following form:

```
<Name of data output>
  <float32_1>RESULT VALUE_1</float32_1>
  <float32_2>RESULT VALUE_2</float32_2>
      :
  <string_1>RESULT VALUE_N</string_1>
      :
</Name of data output>
```

IMAGE

Image evaluation
The IMAGE command is only allowed in the "Triggered" and "Free run" evaluation modes. The command performs a complete evaluation. The inspection results and features are thereby entered in the corresponding flagword and string addresses.

CHANGEPRG

Change check program
This command allows the check program to be changed. The values 1 to 256 are permitted as check program numbers.

Requirements:

- The system parameter "Check program preselection" is configured to "I/O possibilities". Otherwise, you will receive an error message.
- The device is ready for operation:
  Output "Ready for Operation" = logic 1.

EXIT

Terminate connection
The Compact Vision System closes the Telnet connection.

VERSION

Inquiry of the version of the Telnet server and the version of the device.

Messages and error descriptions

Code	Message/Error	Description
24 AUTH	enter password	– Request for entry of the password after transmission of a user name. Use the command “PASS”
23 AUTH	access denied	– Message after failed authentication attempt.
22 AUTH	access granted	– Message after successful authentication attempt.
21 SBOx-Q-ZEWAS	authentication required	– Initial message of the server if authentication is necessary.
20 SBOx-Q-ZEWAS	no authentication required	– Initial message of the server after a connection is built up if no authentication is required.
1 OK	operation successful	– Command successfully executed.
–1 [1] ERROR	parse error, or command unknown	– Command invalid. – Error in the instruction.
–5 ERROR	another client is already connected	– Message with existing connection.
–6 ERROR	maximum number of connections reached	– Warning if an attempt is made to set up more than 10 connections.
–10 ERROR	camera not ready	– Compact Vision System not ready for action when the command is executed.
–11 ERROR	timeout during last operation	– Timeout when executing the command.
–12 ERROR	response line overflow	– The response line exceeds the valid number of characters.
–20 ERROR	program switch failed	– Check program could not be switched.
–21 ERROR	program number not valid	– Specified check program number is not valid.

1) up to firmware 3.4x: code = 0; firmware 3.5 or higher: code = –1

Code	Message/Error	Description
–30 ERROR	read offset not valid	– One or more of the specified flagword addresses is invalid for read access.
–40 ERROR	write offset not valid	– Specified flagword address is invalid for write access.
–50 ERROR	string offset not valid	– String address not valid.
–55 ERROR	one or more feature name(s) not valid	– One or more name(s) in the RNV (Read Name Value) command are invalid.
–56 ERROR	read data output	– Miscellaneous) data output error.
–58 ERROR	read data output: name not defined	– Data output not set up for name specified.
–59 ERROR	read data output: data output not compatible	– Data output not suitable for output via Telnet or to the CoDeSys module.
–61 ERROR	no authentication required	– No authentication required (after use of the control commands USER or PASS).
–62 ERROR	already authenticated	– No authentication required (after use of the control commands USER or PASS).
–63 ERROR	no user provided	– The command PASS was received before transmission of a user name.
-200 ERROR	unspecified error	– Error not further described.

Tab. 4/10: Messages and error descriptions

4.10.9 Use of the Ethernet interface with Telnet streaming

The device makes the Telnet protocol available for streaming for communication with a higher-level controller, robot or PC. This permits very wide-ranging output and control possibilities. Check results can be processed further in the higher-level controller, in order to have a workpiece grabbed by a robot, for example.

Note
Support for the Telnet streaming function by the Compact Vision System must be enabled via the system parameters (➔ CheckKon).

The Telnet streaming function outputs data in the configured TCP channel without explicit request. Up to 10 clients can receive the data.

In the Telnet streaming channel, data are only output provided the following conditions are met:

– the Telnet streaming function is activated,

– the check program selected contains Telnet data outputs,

– the Telnet data outputs are configured for automatic output (streaming).

Data outputs can be inserted in a check program and configured with CheckOpti.

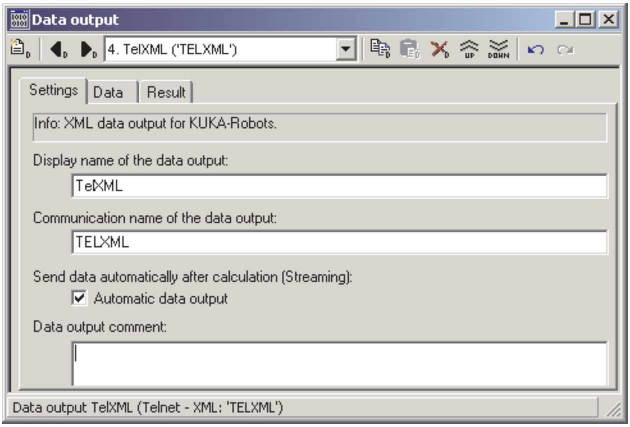


Fig. 4/22: Configuring data output with Telnet streaming

In addition to defining which data are to be output, a data output also defines the format of the data (binary, XML).

Additional information on creating check programs can be found in the CheckKon Help. Please consult your local Festo service if necessary.

Note
In the Telnet streaming channel ...
- no commands are evaluated by clients,
- no error messages are output.

4.10.10 Use of the Ethernet interface with Telnet XML

The device makes the Telnet protocol for XML available for communication with a higher-level controller, robot or PC. This permits very wide-ranging output and control possibilities.
Check results can be processed further in the higher-level controller, in order to have a workpiece grabbed by a robot, for example.

➔

Note
Support for the Telnet XML function by the Compact Vision System must be enabled via the system parameters (➔ CheckKon).

Reading and writing take place via a text-based command line with defined commands, whereby the data to be read/ written are defined through memory addresses.

The device not only makes simple commands available for image capture, but also allows access to the results of a check and the system parameter settings, among other things.

Certain memory addresses can be both read and written to, while others can only be read or only written to (➔ tables in appendix A.6).

For commissioning:

1. Switch the system off in order to avoid faults or damage. Disconnect the power supply to the Compact Vision System and the device to be coupled.

2. Using the specified cables, connect the Compact Vision System to the device to be connected, e.g. via an Ethernet switch or hub.

3. Reconnect the power supply.

4. Make a connection to the Compact Vision System with the CheckKon program and set the system parameters under “Telnet function”:
   - The system parameter “XML authentication required” determines whether the device to be coupled has to identify itself (password).
   - The system parameter “XML TCP port” determines which port should be used for the Telnet communication with XML.
   - The system parameter “XML Telnet server” activates the Telnet function with XML.
5. Specify the IP address and the XML Telnet port of the Compact Vision System at the device to be connected.
6. Program the device to be connected to enable it to access Compact Vision System data.

General information on inputs can be found in chapter 4.10.1. The available memory addresses and their function can be found in appendix A.6.

Test of Telnet communication

Most PC operating systems have a Telnet program at command line level. Using this program, you can test the Telnet connection to the Compact Vision System.

Requirement in Windows Vista or higher

- Activate the option “Telnet Client”.
  This setting can be found in “Windows functions ...” under “Programs and Functions” in the system control.
- Open a window with command line via the Windows Start Menu [Programs] [Accessories] [Entry].
- Start the Telnet program and indicate the IP address of the Compact Vision System as well as the port number defined in the “XML TCP port” system parameter.

Fig. 4/23: Starting the Telnet program via the command line

This establishes a connection to the Compact Vision System:

Fig. 4/24: Telnet program with SBO...-Q acknowledgement (here with authentication)

The Compact Vision System now awaits authentication, if necessary.

It is possible that the data you input in Telnet are not visible (e.g. in Telnet for Windows 2000).
Activate the so-called “Local Echo” in Telnet.
Information on this can be found in the Help for the Telnet program.

XML structure

The return values from the Compact Vision System in XML format always adhere to the following syntax:

– There is exactly one root element with the name

```
<Camera>
...
</Camera>
```

– Confirmation and error messages consist of an integer code and an explanatory text in the following format:
  <CommandResultCode>INTEGER</CommandResultCode>
  <CommandResultText>STRING</CommandResultText>

Command	Return value if operation successful
Read FlagWord	<DataResult_RFW>STRING/REAL</DataResult_RFW>
Read Named Value	<DataResult_RNV>STRING/REAL</DataResult_RNV>
Read String	<DataResult_RSTR>STRING</DataResult_RSTR>
Read Data Output	<DataResult_RDO> /* User-defined – start */ <Name of data output> <float32_1>REAL</float32_1> <float32_2>REAL</float32_2> </Name of data output> /* User-defined – end */ </DataResult_RDO>

Detailed information on the individual commands can be found in the section "Telnet XML commands".

Example

```
RNV "Angle of edge" [ENTER]
<Camera>
  <DataResult_RNV>108.949</DataResult_RNV>
</Camera>
```

Authentication

If authentication has been enabled via the "XML authentication required" system parameter, users must identify themselves after the connection is made.

This occurs with the following commands:

– USER ‹User name›

– PASS ‹related password›

The user “root” is set up on the Compact Vision System. The related password in the delivery condition of the Compact Vision System is “Festo”. Additional users are not supported.

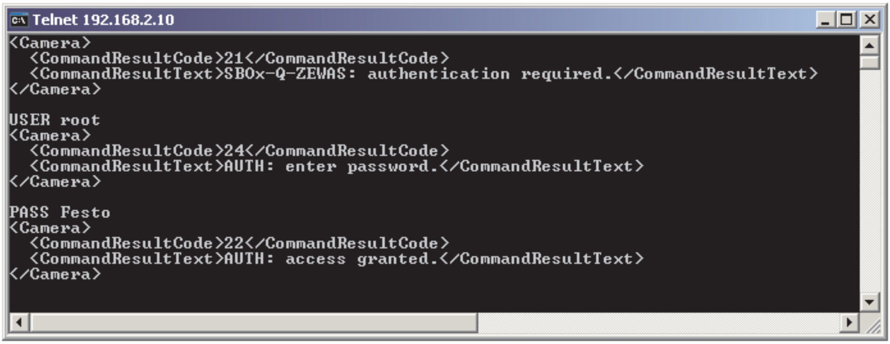


Fig. 4/25: Authentication in type SBO...-Q

If the combination of user name and password is valid, this message appears:

```
<Camera>
        <CommandResultCode>22</CommandResultCode>
        <CommandResultText>AUTH: access granted.</CommandResultText>
</Camera>
```

– After this message has been displayed, all control and data commands are valid for the duration of the connection.

If the user name or password are invalid, the following message appears:

```
<Camera>
        <CommandResultCode>23</CommandResultCode>
        <CommandResultText>AUTH: access denied.</CommandResultText>
</Camera>
```

– The Compact Vision System accepts a new authentication attempt only after a safety time of 2 seconds.

Until authentication is successful, all control and data commands are rejected (exception: “exit”).

The Telnet function of the Compact Vision System supports up to 10 active connections. An attempt to establish more than 10 connections to the Compact Vision System results in the following message:

```
<Camera>
        <CommandResultCode>-6</CommandResultCode>
        <CommandResultText>ERROR: maximum number of connections
          reached.
        </CommandResultText>
</Camera>
```

After this message is shown, the connection is ended.

Telnet XML commands

WFW – Write FlagWord

WFW ‹address, value›
Writes a value to the flag word address specified (➔ appendix A.6).

The following example writes the value 7 to the flagword address 33 (fast access to check program preselection).

Fig. 4/26: Writing to a flag word address

If the operation is successful you will receive a confirmation message- in the format:

```
<Camera>
        <CommandResultCode>1</CommandResultCode>
        <CommandResultText>OK: operation successful.
          </CommandResultText>
</Camera>
```

In case of error, an error code and error message are shown.

Note
- Use only integers when writing values to flag word addresses.

RFW – Read FlagWord

RFW ‹address›
Outputs the current value of the specified flag word address as text (➔ appendix A.6).

Only 1 address can be read out in each case using the Telnet XML function.

Fig. 4/27: Reading out the flagword address

If the operation is successful, the result will appear in the following format:

```
<Camera>
        <DataResult_RFW>RESULT VALUE
         </DataResult_RFW>
</Camera>
```

RNV – Read Named Value

RNV [name of feature]
RNV “Name of feature”
Ouputs the current feature value of the specified feature name (➔ appendix A.6).

Only 1 feature value can be read out in each case using the Telnet XML function..

Note
Only if the feature name is written completely and correctly will a value be output.

- Do **not** use any of the following characters in the feature name:
  – square brackets [ ]
  – quotation marks “ ”
  – umlauts
  – special characters.
- Use correct capitalisation in feature names.

If the operation is successful, the result will appear in the following format:

```
<Camera>
       <//DataResult_RNV>RESULT VALUE
        </DataResult_RNV>
</Camera>
```

If no match is found, the device puts out the following message:

```
<Camera>
       <CommandResultCode>-55</CommandResultCode>
       <CommandResultText>ERROR: one or more feature name(s)
         not valid”.
       </CommandResultText>
</Camera>
```

The feature names are stored in the check programs and can be established when creating the check program (in CheckOpti).

- Preferably use meaningful feature names.

If several feature names are identical, the value of the first feature whose name matches is output.

RSTR – Read String

RSTR ‹address›
Outputs the current character string of a string address (➔ appendix A.6.11). Only one string address can be read out per command. Simultaneous reading of multiple character strings is not possible.

Commas, apostrophes and control characters contained in the strings are replaced by "_" characters.

If the operation is successful, the result will appear in the following format:

```
<Camera>
       <DataResult_RSTR>RESULT VALUE
        </DataResult_RSTR>
</Camera>
```

RDO – Read Data Output

RDO [name of data output]
RDO “name of data output”
Outputs the values of the respective data output.

Note
Data outputs can only be requested if they have been created previously in CheckOpti with the corresponding name.

If no data output has been created, an error message appears.

Details of configuration of a data output can be found in the CheckOpti online Help.

Values are output only if the data output name has been entered completely and correctly.

- Do **not** use any of the following characters in data output names:
  – square brackets [ ]
  – quotation marks “ ”
  – umlauts
  – special characters.
- Use correct capitalisation in data output names.

If the data output exists and uses the Telnet XML function, you will receive the result in the following format:

```
<Camera>
  <DataResult_RDO>
    <Name of data output>
      <float32_1>RESULT VALUE_1</float32_1>
      <float32_2>RESULT VALUE_2</float32_2>
          :
      <string_1>RESULT VALUE_N</string_1>
          :
    </Name of data output>
  </DataResult_RDO>
</Camera>
```

IMAGE

Image evaluation
The IMAGE command is only allowed in the “Triggered” and “Free run” evaluation modes. The command performs a complete evaluation. The inspection results and features are thereby entered in the corresponding flagword and string addresses.

CHANGEPRG

Change check program
This command allows the check program to be changed. The values 1 to 256 are permitted as check program numbers.

Requirements:

- The system parameter “Check program preselection” is configured to “I/O possibilities”. Otherwise, you will receive an error message.
- The device is ready for operation:
  Output “Ready for Operation” = logic 1.

EXIT

Terminate connection
The Compact Vision System terminates the Telnet XML connection.

VERSION

Requests the version of the Telnet XML server and of the device.

Messages and error messages (➔ Tab. 4/10).

4.10.11 Use of the Ethernet interface with Modbus

The device makes available the Modbus protocol for communication with a higher-level controller. This protocol permits very wide-ranging output and control possibilities. Inspection results can be processed again in the higher-order controller.

Support of the Modbus Protocol by the Compact Vision System must be activated via system parameters. Reading and writing take place via the data packages defined by Modbus, whereby the data to be read/written are defined through memory addresses.

The device not only makes I/Os available via memory addresses, but also allows access to the results of an inspection and to the settings of the system parameters.

Certain memory addresses can be both read and written to, while others can only be read or only written to (➔ tables in appendix A.6).

The following function codes are supported:

– 3 (Read Holding Registers = reading several 16-bit registers)

– 16 (Write Holding Registers = writing several 16-bit registers)

Commissioning:

1. Switch the system off in order to avoid faults or damage. Disconnect the power supply to the Compact Vision System and the device to be coupled.

2. Using the specified cables, connect the Compact Vision System to the device to be connected, e.g. via an Ethernet switch or hub.

3. Reconnect the power supply.

4. Set up a connection to the Compact Vision System using the CheckKon program and set the following system parameters under “Modbus function”:

   – The system parameter “TCP port” defines which port should be used for Modbus communication.

   – The system parameter “Protocol type” determines which protocol type of the Modbus protocol should be used with a higher-order controller. At the moment, only TCP mode is available.

   – The system parameter “Maximum number of connections” determines how many Modbus clients are permitted to connect simultaneously to the SBO...-Q Compact Vision System.

Note
If there is more than one connected Modbus client, this can result in contradictory parameter settings, since all Modbus clients are equally prioritized. The requests are worked off in the order of their receipt.

5. Set the IP address and the used Modbus port of the Compact Vision System in the device to be connected.

6. Program the device to be connected to enable it to access Compact Vision System data.

General information on inputs can be found in chapter 4.10.1. The available memory addresses and their function can be found in appendix A.6.

4.10.12 Use of the Ethernet interface with EasyIP

The device makes the EtherNet/IP protocol available for communication with a higher-level controller.
This allows comprehensive exchange of data and control of the device. Data can be preprocessed in the Compact Vision System or directly transmitted to the controller for further processing.

Read/write processes are performed via the defined EtherNet/IP objects, which allow access not just to I/O data but also to system parameters.

Certain memory addresses can be both read and written to, while others can only be read or only written to (➔ tables in appendix A.7).

Commissioning:

1. Switch the system off in order to avoid faults or damage. Disconnect the power supply to the Compact Vision System and the device to be coupled.
2. Using the specified cables, connect the Compact Vision System to the device to be connected, e.g. via an Ethernet switch or hub.
3. Reconnect the power supply.
4. Set up a connection to the Compact Vision System using the “CheckKon” program and set the system parameters under “Ethernet interface”:
   - The system parameter “EtherNet/IP Server” activates the EtherNet/IP function.

Notification of slave features using “EDS Hardware Installation Tool (Rockwell Automation)”

When you commission a new EtherNet/IP slave for the first time, this allows the system to be notified of its characteristics in advance. This has the advantage that the device is correctly identified in a scan and does not require further configuration.

The characteristics of the various slaves are usually administered in a list or library e.g. EDS library (EDS = electronic data sheets).

The following possibilities are available for extending an EDS library:

– Installing EDS files
  The EDS file serves simply to identify the Compact Vision System in the network.

– Enter slave features manually

Installation of the EDS file is not a prerequisite for placing the device in operation. As an alternative, a generic device with suitable characteristic settings can also be created (➔ section "Entering slave properties manually").
The use of EDS files is not supported by RSLogix Version 17, but is planned for version 18.

Reference source

Current EDS files, icon files and information on the EDS files can be found at the following Internet address:
www.festo.com/fieldbus

Installing EDS files

The following files are required for the Compact Vision System:

– SBOx-Q.eds

– SBOx-Q.ico (Icon file for representation in the configuration program)

Installing EDS files

Install the files as per the RSLogix 5000 documentation.

Symbol files

Depending on the configuration program used, you can assign a symbol file (.ico format) to the Compact Vision System.
The device is then displayed accordingly in the configuration program.

Notes on installing the EDS files and the icon files can be found in the documentation for your controller.

Enter slave features manually

When an EDS file is installed, the following information about the EtherNet/IP slave is added to the EDS library.

Basic information	**Value**	**Description**
Vendor name	Festo	Manufacturer
Vendor ID	26	Manufacturer ID
Device Type	0	Generic device
Product code	21314	Product code
Major version/Minor version	3.5	Version/revision
Input size/output size	32 bit/16 bit	Size of the cyclically transmitted I/O data
Device Type	SBOx Q	Device type

Extended Ethernet/IP slave features	**Value**	**Description**
Request Packet Interval (RPI)	15 ms	Minimum cycle time
Exclusive Owner connections	1	Number of direct connections permitted. (Only one controller may connect with the device.)
Input only connections	0 (not supported)	Used for devices without outputs – not relevant here.
Listen only connections	3	Requires an existing “Exclusive Owner connection”. Up to three devices can “listen”, e.g. for purposes of visualisation on a display.

Assembly Instances	Value	Description
Inputs	769d (301h)	Corresponds to FW 0...31 (1 bit per FW in each case)
Outputs	770d (302h)	Corresponds to FW 16...31 (1 bit per FW in each case)
Configuration data	771d (303h)	Not used

This information can also be entered manually.

When the EDS library has been extended, the Compact Vision System is entered in the slave list as a possible EtherNet/IP slave. It can now be added to a network.

Creating a project in RSLogix 5000

These instructions relate to the use of a CompactLogix controller.

1. In the “I/O Configuration” group in the “RSLogix 5000” software, right-click on “Ethernet” and select the command “New Module ...” in the context menu.

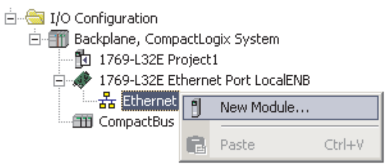


Fig. 4/28: Creating a new module

2. Select “ETHERNET MODULE - Generic Ethernet Module” in the window “Select Module” and confirm with “OK”.

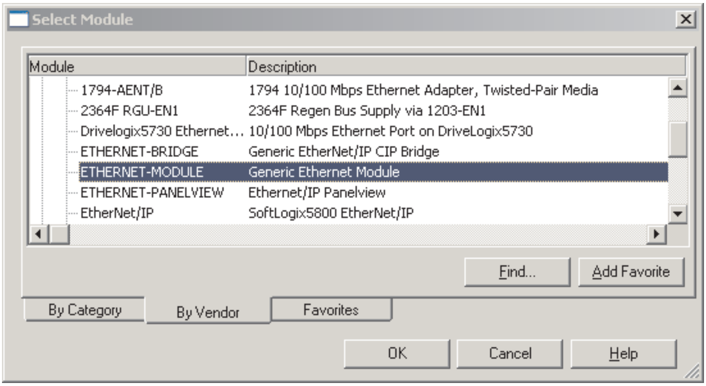


Fig. 4/29: Selecting the Ethernet module

3. Enter the following values in the dialogue for the new module and select “OK”.

Input field	Explanation	Value
Name	Name of fieldbus node	freely selectable
Description	Description	freely selectable
Comm Format	Communication format	Data - INT
IP Address	IP address of the Compact Vision System	e.g. 192.168.2.10
Connection parameters		
– Assembly Instance Input – Size	– Instance for inputs – Size	769 2 (2 x 16 bit)
– Assembly Instance Output – Size	– Instance for outputs – Size	770 1 (1 x 16 bit)
– Assembly Instance Configuration – Size	– Instance for configuration data – Size	771 0 (0 x 16 bit)

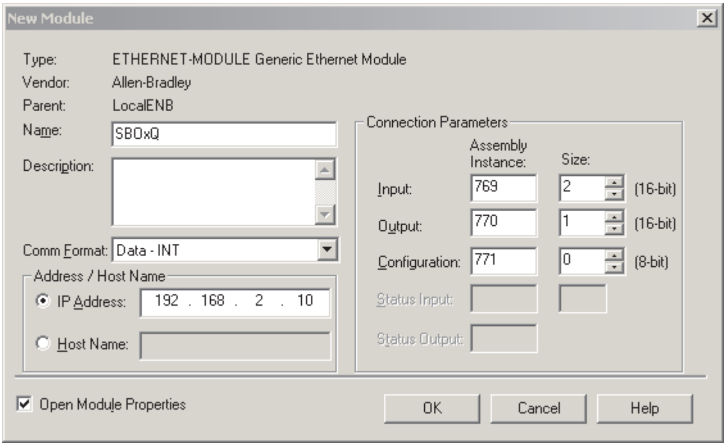


Fig. 4/30: Configuration of the EtherNet/IP module

Data output via EtherNet/IP

Using CheckOpti, you can compile a data output for a check program, which will be available in a field. The data will then be available in the form of a byte array and can be called up via "Explicit Message".

The byte array consists of a fixed header followed by a variable number of data. The nature and number of the data are defined in the header:

Number ...	Data type	Position in header (byte)	Application
Values of type double64	UINT	0	–
Values of type REAL	UINT	2	Yes
Values of type int32	UINT	4	–
Values of type int16	UINT	6	–
Values of type byte	UINT	8	–
Values of type str	UINT	10	Yes

Number ...	Data type	Position in header (byte)	Application
Bytes for length specification in type str	UINT	12	Yes
Characters in type str	UINT	14	Yes

Tab. 4/11: EtherNet/IP data output

At present only data of types “REAL” and “str” are transmitted. Details regarding the creation and configuration of the data output can be found in CheckOpti Help.

Automatic generation of the data type

Creating the structure for data output in RSLogix is very complex and time-consuming. To facilitate this process, CheckOpti offers the possibility of creating a structure file (.l5x). This XML-based file can then be imported into RSLogix. The data type which then becomes available corresponds to the data output previously defined in CheckOpti.

- Create a .l5x file in CheckOpti in the “Result” register in the “Data output” window, after selecting the relevant data output.

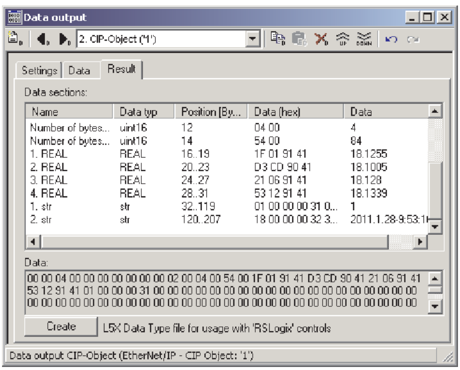


Fig. 4/31: Data output of the EtherNet/IP module

Importing a data type

1. In the "Data Types" group in the "RSLogix 5000" software, right-click on "User-Defined" and select the command "Import Data Type..." in the context menu.

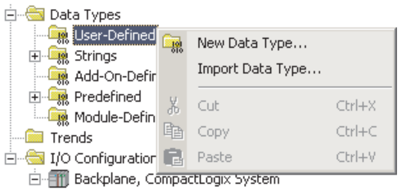


Fig. 4/32: Importing a data type

2. Select a structure file (.l5x) and then "OK".

For each data output defined in CheckOpti, you will find a "DataOutput_" data type in the "User-Defined" area after the import, ending with the respective number (= instance) of the data output.

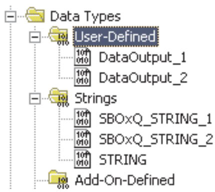


Fig. 4/33: Imported data types

3. When you double-click on a data type, its structure will be displayed.

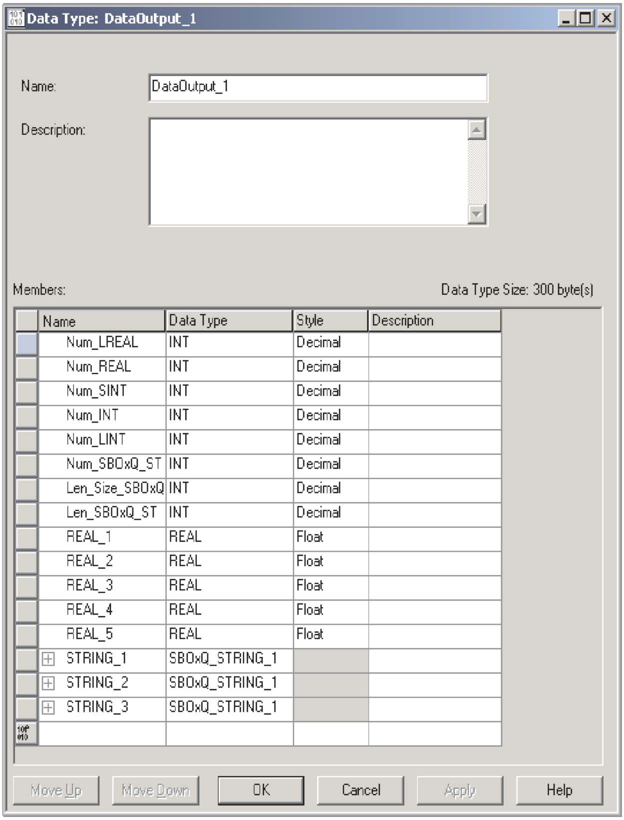


Fig. 4/34: Structure of an imported data type

4.10.13 Use of CoDeSys

In addition to the quality check, the Compact Vision System also includes the “CoDeSys” PLC run-time system as a firmware add-in (included on devices with firmware version 3.4 or higher).

Information about this firmware add-in can be found in CheckKon 4.1 in the “Firmware info” field of the “System information” window.
For updates or in the case of missing firmware add-in, contact Festo Service.

The PLC run-time system “CoDeSys” is a soft PLC that can be programmed with languages standardized by IEC 61131-3.

The library “Festo_SBOx-Q.lib” is available under CoDeSys for simple communication between the CoDeSys run-time system and the “Quality check” application.

Generally, the external communication options (EasyIP, Telnet, etc.) are also available to the device-internal CoDeSys run-time system.
This allows e.g. a Telnet connection to be established from a CoDeSys program on the Compact Vision System to the “Quality check” application on the same device, for purposes of data exchange.

Detailed information on data exchange can be found in the online Help of the programming environment.
“CoDeSys provided by Festo”.

This opens up many options:

- Simple linking of the calculation results of several devices in one camera network.
  Example: A Compact Vision System as a master system can read the calculation results of another Compact Vision System, link them with its own results and, based on this, perform an action.
- Complex inspection sequences can be implemented.
  Example: Switching between check programs and comparing the results.
- Small stand-alone production sequences can be controlled directly from the Compact Vision System.
  Advantages: Reduction of complexity, increase in system availability
- Direct control of servo controllers via CANopen (Compact Vision System is CANopen master).
  Example: No additional controller is needed for fine positioning or control of handling units for flexible gripping of components.

To program the integrated CoDeSys soft PLC, you will need the PC software environment "CoDeSys provided by Festo" as well as the Target Support Package for the respective Compact Vision System firmware.
The Target Support Package for the Compact Vision System can be downloaded at no charge through the Internet.
Please consult your local Festo Service, if necessary.

Commissioning:

1. Switch the system off in order to avoid faults or damage. Disconnect the power supply to the Compact Vision System and, if applicable, the device to be coupled.

2. Using the specified cables, connect the Compact Vision System to the device to be connected, e.g. via an Ethernet switch or hub.

3. Reconnect the power supply.

4. Set up a connection to the Compact Vision System using the CheckKon program and set the system parameters under “CoDeSys control mode”:

   – System parameter “Start CoDeSys (after power supply)” determines whether the CoDeSys run-time system on the Compact Vision System should be activated or deactivated.

Note
The CoDeSys run-time system is activated only after a restart of the Compact Vision System.

   – In a similar way to a selector switch on a PLC, you can use the system parameter “Run/stop operation” to start or stop operation of the PLC program on the Compact Vision System.

5. Now program the integrated CoDeSys soft PLC to enable data from the Compact Vision System to be accessed, and actions to be triggered.

Note
The CoDeSys run-time system depicts its own PLC, which runs on the Compact Vision System independently of the quality inspection software.

And so to operate CoDeSys, an externally coupled device is not mandatory.

General information on inputs can be found in chapter 4.10.1. The available memory addresses and their function can be found in appendix A.6.

“CoDeSys provided by Festo” programming environment

For use of a controller (target) under “CoDeSys provided by Festo”, a so-called Target Support Package is needed for the corresponding target. This permits access to the system functions of the target and contains corresponding information in the form of online help.

The Target Support Package makes CoDeSys functions usable for the respective device or limits them, if necessary.

With the Target Support Package, CoDeSys can support all these characteristics and functions of these devices. And so CoDeSys provided by Festo contains many functions that are available only on certain devices.

The Target Support Package for the Compact Vision System can be downloaded at no charge through the Internet. Please consult your local Festo Service, if necessary.

Installation of the Target Support Package (TSP)

If the software package “CoDeSys V2.3 provided by Festo” has already been installed and operated with other controllers, it may only be necessary to install the Target Support Package (TSP) for the new target system.

The TSP contains information about the target system (Compact Vision System) needed by CoDeSys V2.3 provided by Festo.

For the Compact Vision System, you will need the Target Support Package “Festo SBOx-Q / SBOx-Q-WB (FW 3.5)”.

Note
The Target Support Package is often made available in the form of a ZIP file. Before it is installed, the file must first be unzipped.

- Use a temporary folder for this.
- Make sure that the folder structure contained in the ZIP file is retained.

1. Start the "InstallTarget" program. This program can be found in the start menu under [Programs] [Festo Software] [CoDeSys V2.3 by Festo].
2. Already installed TSP are displayed in the "Installed target systems" area of the program window.

Note
Installation of a new version for an existing Target Support Package requires deinstallation of the old version. Otherwise, files which are no longer needed (e.g. libraries) will be left on your PC drive.
This is how to remove the old version:

- Select the old version of the TSP and click on the "Remove" button.
- Erase the remaining files and folders manually from your PC (standard directory: ...CoDeSys V2.3TargetsFestoSBOx–Q_fw35).

3. Actuate the "Open" button to select the TSP "Festo SBOx-Q / SBOx-Q-WB (FW 3.5)".
4. In the temporary folder (of the unzipped ZIP file), choose the file "SBOx-Q.tnf". Then click on the "Open" button. The new TSP is displayed in the "Possible target systems" area.

5. If necessary, under “Installation directory”, enter the path in which the TSP should be installed. You can select a new installation directory via the “...” button.
   The standard entry for the installation directory is:
   ...CoDeSys V2.3TargetsFestoSBOx-Q_fw35.

6. Highlight the entry “Festo SBOx-Q / SBOx-Q-WB (FW 3.5)” on the left-hand side under “Possible target systems” and click on the “Install” button.
   If the specified installation directory does not exist, a safety check apprears.

   – Click on the “Yes” button if a new installation directory is to be created.

   – Click on the “No” button if you want to select a different installation directory.

Now the new Target Support Package should be visible on the right side under “Installed target systems”.

7. End the program using the “Close” button.

A detailed description of the function of the integrated CoDeSys run-time system can be found in the online Help of the Target Support Package “Festo SBOx-Q / SBOx-Q-WB (FW 3.5)”.

4.10.14 Display of inspection results with the "SBO...-Q WebViewer"

With the "SBO...-Q WebViewer", inspection results and camera images can be depicted in a Web browser.

Web browsers are available on all PC operating systems (e.g. Windows® Internet Explorer).

Note
Use of the WebViewer extends the evaluation duration of the Compact Vision System.

- Check whether the required parts rate can still be achieved.

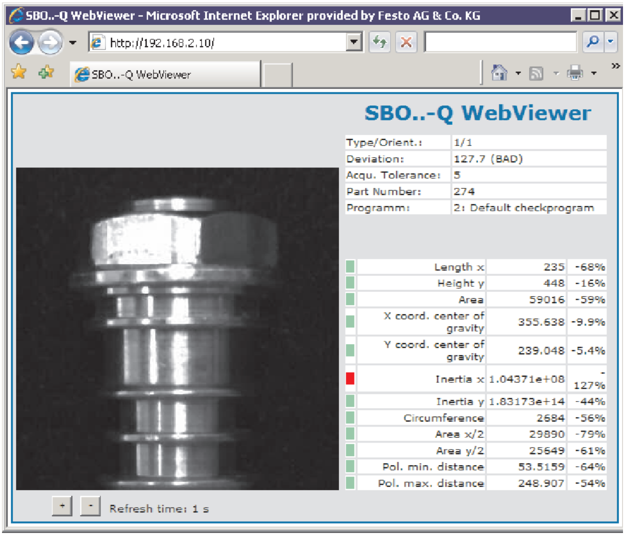


Fig. 4/35: Inspection results in the SBO...-Q WebViewer

Support of the WebViewer by the Compact Vision System must be activated via the system parameters.
Once this has been done, a Web browser can be directly connected with the "SBO...-Q WebViewer" through entry of the IP address of the device, and the check results displayed.

The display interval can be set in the Web browser between 0.5 and 3 seconds. The display is updated accordingly. The results and the camera image of the last inspection are thereby displayed.

Note
For inspections with cycletimes below the set display interval, not all results and images can be displayed in the WebViewer.

Commissioning:

1. Set up a connection to the Compact Vision System using the CheckKon program.
2. Select a setting from the following table for the "WebViewer" system parameter in the section "Ethernet interface":

Setting	Output
On (all results)	All features calculated by the Compact Vision System
On (results from data output HTTP – WebViewer)	All features configured in the data output HTTP – Webviewer using CheckOpti

Note
For each check program, only one data output of type "HTTP – WebViewer" is possible.

If you have not configured a "HTTP – WebViewer" type data output, you will not receive any feature results.

3. Start the Web browser and enter there the IP address of the Compact Vision System.
   Examples:

   – “http://192.168.2.10” ‹ENTER›
     for display of image, inspection result and feature values.

   – “http://192.168.2.10/imageview.html” ‹ENTER›
     for display of the taken image **without** inspection result and feature values.

4. Trigger a new inspection to obtain a display.

If the Web browser reports a connection error, check the Ethernet and Internet settings for the Web browser or the system on which the Web browser is running.

- Open “Internet Options” in [Settings] [System Control] in the Windows operating systems.

- Check, in particular, whether you might have to deactivate the so-called proxy server to obtain a display.
  You will find the configuration for the Internet and also the settings for the proxy server under [Connections] [Settings] in the “Characteristics of Internet” dialogue.

4.11 Creation of the check programs

Check programs define how parts are to be checked, and in particular what features of a check part are to be calculated (e.g. length of check part) and what values are permissible for a good part.

The check program to be used is determined via the check program preselection. It can be selected via:

- System parameters
- I/O options.

Check programs located on the device can be updated using the following system parameters in the section System Operating Modes Teach Mode.

- Teach mode
- Parts type
- Parts orientation.

Update of a check program becomes necessary as soon as system parameters that have an influence on image creation and image processing are changed. These include, in particular, system parameters in the sections

- Evaluation
- Lighting
- Camera image and pre-processing.

Information on updating check programs and for setting the system parameters can be found in the parameter help in the "System parameters" window of the CheckKon software package.

Check programs can be easily created and evaluated with the CheckOpti software package. The created check programs can then be transferred to the device with CheckOpti or CheckKon.

Additional information on preparing check programs can be found in the CheckKon Help. Please consult your local Festo Service if necessary.

4.12 Checking the system settings

At completion of commissioning, the following points must always be rechecked:

- Emergency stop concept and function
- Cabling
- Control program
- Reliability of the results for various inspection parts
- Reliability of the results for various extraneous light conditions.

Save the data of the software packages as files.

4.13 Instructions on operation

Caution

- Make sure that no danger can emanate from the systems connected to the Compact Vision System.

If the permitted temperature range is exceeded, this will be recognised by the internal electronics and will lead to an error status.

Caution

Further heating beyond this point can lead to uncontrolled malfunctions.

- Make sure that the permitted temperature range is complied with (➔ Technical data).

Diagnostics and error handling

Chapter 5

Table of contents

5. Diagnostics and error handling 5-1

5.1 General diagnostic options 5-3

5.1.1 Status display 5-3

5.1.2 Error handling 5-6

5.1 General diagnostic options

The following diagnostic options are available:

- CheckKon can display operating conditions and error messages for the Compact Vision Systems used (➔ CheckKon Help).
- Four LEDs on the rear of the Compact Vision System supply the status information listed in the following section.

5.1.1 Status display

Operating statuses are displayed via LEDs.

Operating status LED (A)			
LED	**Sequence**	**Status**	**Meaning/error handling**
 LED flashes green		Device is ready to operate.	–
 LED is red		Initialization is in progress	Wait until initialization is complete
 LED is off		Indeterminate status, e.g. operating voltage not present	Check power supply to the electronics

Tab. 5/1: Operating status LED (A)

Ethernet traffic LED (B)			
LED	**Sequence**	**Status**	**Meaning/error handling**
LED flashes green	ON OFF	Ethernet data traffic	–
LED is off	ON OFF	No Ethernet data traffic (no traffic)	–

Tab. 5/2: Ethernet traffic LED (B)

Activity LED (C)			
LED	**Sequence**	**Status**	**Meaning/error handling**
LED is red	ON OFF	Error	–
LED flashes red	ON OFF	Warning	–
LED is yellow	ON OFF	Device is ready to operate; evaluation can begin	–
LED is off	ON OFF	Evaluation is in progress	–

Tab. 5/3: Activity LED (C)

Output LED (D)			
LED	**Sequence**	**Status**	**Meaning/error handling**
LED is red	ON OFF	Last evaluation detected bad part	–
LED is yellow	ON OFF	Last evaluation detected good part	–
LED is off	ON OFF	No result	–

Tab. 5/4: Output LED (D)

The function of the output LED can be configured via system parameters; the descriptions here correspond to the standard configuration.

5.1.2 Error handling

Problem	Cause	Remedy
The device is not performing evaluations	– Operating voltage not present or is below the permitted tolerance	• Switch on operating voltage or comply with tolerances.
	– System parameters incorrect	• Check system parameters using CheckKon
	– Trigger signal absent or polarity incorrect	• Check trigger signal
	– Hardware error	Servicing required
The device's evaluations detect only bad parts	– Incorrect parts type	• Diagnose process using CheckKon
	– Check program incorrect	
	– System parameters incorrect	• Check system parameters using CheckKon
The device firmware freezes. (Status LED A is not flashing)	– Electromagnetic interference in the environment caused by non CE-compliant devices	• Eliminate source of interference • Check that the screening of the connecting cables for the Compact Vision System is routed correctly with low impedance. • Use a separate power unit for the Compact Vision System
The evaluation image is blurred or unfocused	– The device moved (e.g. due to vibration in the machine/system).	• Check mounting, reduce vibrations
	– The object is moving too quickly.	• Reduce exposure time
	– The object lies outside the focal range.	• Ensure minimum distance is adhered to. SBOI-Q: 22 mm SBOC-Q: dependent on lens selected
	– Lens not focussed	• Focus the lens

Problem	Cause	Remedy
Optical errors in the evaluation image	– Lens or protective glass dirty	• Carefully clean lens or protective glass
CheckKon cannot establish a connection to the Compact Vision System	– Incorrect cable	• For a direct connection to the PC, you will probably require a connecting piece and a so-called crossover cable in addition to the original cable. This is not necessary for a connection via a hub or a switch (➔ chapter 3.2.2.).
	– Your network is preventing data exchange.	• Make sure that your router passes on the multicast address 239.255.2.3. If in doubt, consult your system administrator.
	– Firewall of the PC or network does not permit a connection.	• Enable program or ports in firewall.
	– PC network card is deactivated (e.g. in notebook without power supply).	• Adapt Windows settings (➔ Power options).
	– Device not in stop status.	• Cancel trigger signal at the device.
	– Device already connected to a different program/user.	• Disconnect other connection.
	– Cause not identifiable	• Reset device (switch power off and on).
Windows error message	– Insufficient free virtual memory	• Ensure system requirements are met (➔ CheckKon Help)

Problem	Cause	Remedy
The programming environment CoDeSys provided by Festo cannot connect to the Compact Vision System	– Incorrect cable	• For a direct connection to the PC, you will probably require a connecting piece and a so-called crossover cable in addition to the original cable. This is not necessary for a connection via a hub or a switch (➔ chapter 3.2.2.).
	– Firewall of the PC or network does not permit a connection.	• Enable program or ports in firewall.
	– CoDeSys run-time system is not activated	• Activate the run-time system (➔ chapter 4.10.13)
	– Incorrect communication parameters in the CoDeSys project	• Change the communication parameters in the CoDeSys project to reflect the current Compact Vision System settings (➔ Programming environment online Help)

Tab. 5/5: Error elimination

Technical appendix

Appendix A

Table of contents

A. Technical appendix A-1

A.1 Cleaning and care A-3
A.2 Addressing in the Ethernet (basics) A-4
A.3 Siemens star A-7
A.4 Technical data A-8
A.5 Error messages A-11
A.6 Address table for EasyIP, Modbus, Telnet and CoDeSys embedded A-14
A.6.1 Input register A-14
A.6.2 Output register A-15
A.6.3 Rapid access to the input and output registers A-16
A.6.4 Further system status/system information A-17
A.6.5 System time of the device A-18
A.6.6 Total tolerance of the type in the current check program A-18
A.6.7 Basic results of the last check A-19
A.6.8 Free flag words (non-remanent) A-21
A.6.9 Features – results of the last check A-21
A.6.10 System parameters A-24
A.6.11 String address table A-25
A.7 Address table for EtherNet/IP A-26
A.7.1 Address table for EtherNet/IP – protocol-specific objects A-26
A.7.2 Address table for EtherNet/IP – SBO-specific objects A-30
A.8 Data types A-42
A.9 Programming robot controllers A-44
A.9.1 Telnet communication with an ABB robot A-44
A.9.2 Telnet communication with a KUKA robot using XML A-53

A.1 Cleaning and care

Caution
A dirty and scratched lens or dirty and scratched protective glass can lead to optical errors.

- Make sure that the lens/protective glass is not scratched.
- Do not use any abrasive cleaning agents.

- Switch off the power supply before cleaning.
- Clean the lens/protective glass in case of dirt or other deposits:
  - using a blower brush or clean, non-lubricated compressed air
  - using a soft, moist cloth and a non-abrasive cleaning agent
- Clean the device if it is dirty.

Permitted cleaning agents are soap suds (max. +60 °C) and all non-abrasive agents.

A.2 Addressing in the Ethernet (basics)

Due to the separation into logical and physical protocol layers (Ethernet and TCP/IP), there are two types of addresses in a network:

– a fixed Ethernet address (MAC ID) for each device and

– an IP address which is assigned to every device in the network.

From the application, data are always sent to or received from an IP address. In order that the data reach the receiver, a link must be created between the logical IP address and the physical Ethernet address. The Address Resolution Protocol ARP is used for this purpose: An ARP table is saved in each network PC. This table lists the relevant physical Ethernet address for each IP address in the network. If an Ethernet address is not listed in the ARP table, the IP driver can ascertain it with the aid of an ARP request.

Ethernet address (MAC ID)

The unchangeable, globally unique Ethernet address (MAC-ID) of the Compact Vision System can be found on the type plate. In this way you can clearly distinguish between the different Compact Vision Systems.

IP address

An IP address as per the IPv4 standard is usually expressed as 4 decimal numbers separated by dots (each 1 byte).

Example of an IP address: 192.168.2.10

An IP address addresses both a network and an individual slave in the network. The IP address therefore contains:

– the net ID (specifies the address of a network) and

– the host ID (specifies the address of an individual slave in this network).

Net mask

Which of the numbers in an IP address represent the net ID and which the host ID is defined through specification of a so-called “net mask”.

The telephone number of Festo Germany can be used as an example to explain IP addresses and net masks: 00497113470

To know which part of this number is the dialling code and which the subscriber number, it is additionally necessary to know: “that the first 7 digits are the dialling code, and the last 4 the subscriber number”. This is the “net mask” of the telephone number.

Net classes

Using “0” as a placeholder, the net mask for IP addresses defines which bytes are used for the addressing of the slave (host ID). Networks belong to different net classes according to the number of such bytes:

Net class	**Net mask**	**Explanation**
A	255.0.0.0	Large network
B	255.255.0.0	Medium-sized network
C	255.255.255.0	Small network with max. 254 slaves

Tab. A/1: The main network classes with their corresponding net masks (example)

Gateway

Networks with different network IDs are connected to each other via routers or gateways. If a slave in a network is to send data to slaves in other networks, the IP address of the gateway must be specified.

Three pieces of information are therefore required for addressing in the Internet Protocol IP:

– IP address
– IP net mask
– IP address of the gateway

Note
The factory settings are as follows:
– IP address: 192.168.2.10
– IP net mask: 255.255.0.0
– IP address of the gateway: –

A.3 Siemens star

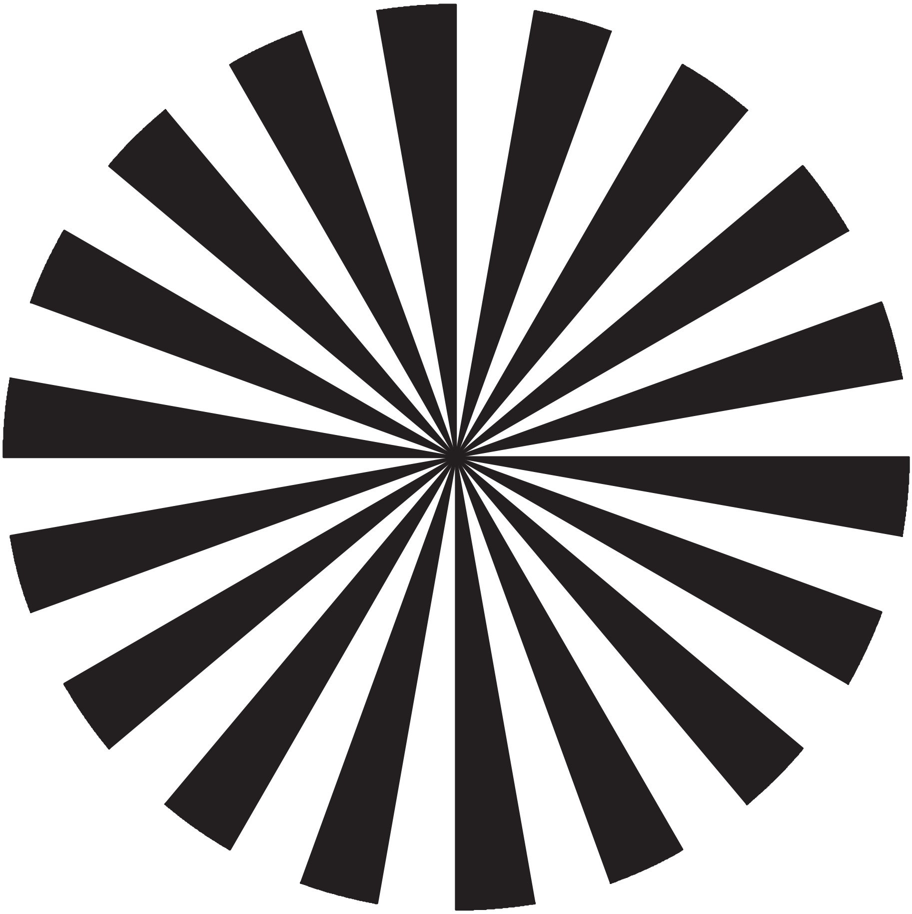


Fig. A/1: Siemens star

The Siemens star depicted here is a helpful template for rough focussing.

A.4 Technical data

Type		SBOC-Q-R1	SBOI-Q-R1	SBOC-Q-R3	SBOI-Q-R3	SBOC-Q-R2
Sensor resolution	**[pixels]**	**640x480**		**752x480**		**1280x1024**
Image sensors						
Exposure time	[ms]	0.039 ... 1000		0.018 ... 200		0.008 ... 1000
Frame rate (full image)	[fps]	150		60		27
Sensor size	[inch]	1/2		1/3		2/3
Sensor type		CMOS Global Shutter; B = monochrome				
		CMOS Global Shutter; C = colour				
Lens mount		CS-Mount 1)	Integrated lens	CS-Mount 1)	Integrated lens	CS-Mount 1)
Working distance	[mm]	depends on lens selected	22 ... 1000	depends on lens selected	20 ... 550	depends on lens selected
Field of view	[mm]	depends on lens selected	14x10 ... 520x390	depends on lens selected	7.9x5.5 ... 195x125	depends on lens selected
Electrical data						
Nominal operating voltage	[V DC]	24				
Permissible voltage fluctuations	[%]	±10				
Current consumption with load-free outputs	[mA]	120				
Max. total current	[A]	1.5 at the 24 V outputs				
Inputs		Functionality in SBO...-Q Input 1: – Trigger signal, use by CoDeSys/check program Input 2: – Apply inputs, acknowledge errors, use by CoDeSys/check program				
Outputs		Functionality in SBO...-Q Outputs can be parameterised: – Ready for operation, good part, bad part, correctly oriented, incorrectly oriented, warning, error, external lighting, use by CoDeSys/check program				

1) C-Mount only with protective lens barrel or SBOL-C-5 adapter

Type	**SBOC-Q-R1**	**SBOI-Q-R1**	**SBOC-Q-R3**	**SBOI-Q-R3**	**SBOC-Q-R2**
Sensor resolution [pixels]	**640x480**		**752x480**		**1280x1024**
Technical data (continued)					
Protection class 2)	IP65, IP67 3)	IP65, IP67	IP65, IP67 3)	IP65, IP67	IP65, IP67 3)
Protection against direct and indirect contact	PELV (Protected Extra Low Voltage)				
Resistance to interference	as per EN 61000-6-2				
Emitted interference	as per EN 61000-6-4 (industry)				
CE mark ➔ Declaration of conformity	as per EU EMC Directive 4)				
Certification	C-Tick, C-UL US Recognized (OL)				
Vibration and shock					
Vibration resistance	tested as per IEC 68/EN 60068 part 2-6; 0.35 mm path at 10 ... 60 Hz; 5 g acceleration at 60 ... 150 Hz				
Shock resistance	tested as per IEC 68/EN 60068 part 2-27; ±30 g at 11 ms duration; 5 shocks in each direction				
Continuous shock resistance	tested as per IEC 68/EN 60068 part 2-29; ±15 g at 6 ms duration; 1000 shocks in each direction				
Ethernet					
Bus interface	IEEE802.3U (100BaseT)				
Connector plug	M12 plug				
Transmission speed [Mbps]	100				
Supported protocols	EtherNet/IP				
	EasyIP				
	Modbus TCP				
	Telnet				
Fieldbus interface					
Type	CAN		–		CAN
Connection technology	M12 plug		–		M12 plug
Supported protocols	CP fieldbus		–		CP fieldbus

2) Protection class as per EN 60 529: Plug connectors inserted or fitted with protective cap
3) With protective barrel
4) In residential buildings, measures for radio interference suppression may be necessary

Type		**SBOC-Q-R1**	**SBOI-Q-R1**	**SBOC-Q-R3**	**SBOI-Q-R3**	**SBOC-Q-R2**
Sensor resolution	**[pixels]**	**640x480**		**752x480**		**1280x1024**
Operating and environmental conditions						
Ambient temperature	[°C]	–10 ... +50				
Storage temperature	[°C]	–10 ... +60				
Ambient conditions		Screened from extreme external light sources				
		Cleanest possible ambient air				
Geometry						
Width	[mm]	45	45	45	45	45
Height	[mm]	45	45	45	45	45
Length	[mm]	139.4 5)	83.7	139 5)	83.7	139.4 5)
Materials						
Housing		Anodised aluminium				
Cover		Acrylic butadiene styrene, glass fibre reinforced				
Note on materials		Contain no copper or PTFE, RoHS-compliant				
Product weight	[g]	182 6)	184	172 6)	174	182 6)

5) With protective barrel
6) Without protective barrel

Tab. A/2: Technical data

A.5 Error messages

Name	No.	Error/warning	Description
General errors			
E00	0	–	No error
E09	9	F	Overload at internal I/Os
E12	12	F [1)]	Overheating
E19	19	F	Firmware not compatible or defective
Teach error			
E20	20	F	Error in check program; orientations cannot be disconnected
E21	21	F	Error in check program; types or orientations cannot be disconnected
Error in image generation			
E30	30	F [1)]	Image buffer overflow (only in evaluation mode = fixed frame rate)
E31	31	F	Error in image generation
Error in check program or system parameter			
E40	40	F	Check program could not be read/found
E41	41	F	System parameters cannot be read/interpreted

[1)] Error can be configured

Name	No.	Error/warning	Description
E43	43	F	Check program is not compatible with firmware At least one of the following tests was not successful: – Check program may be used with firmware version – Tool settings are compatible with firmware version – Data output settings are compatible with firmware version – Required check program licence (e.g. GSLO-S1) present on Compact Vision System
E45	45	F	Check program could not be activated/loaded - new attempt after Acknowledge Error signal
E46	46	F	Insufficient memory, operation cannot be executed Remedy: – Reduce the size of the active check program – Reduce the size of the active field of view of the Compact Vision System
CP I/O expansion error			
E100	100	F [1]	General CAN error
E101	101	F [1]	General I/O expansion error
E102	102	F [1]	An output module was not found
E103	103	F [1]	An input module was not found
E105	105	F [1]	Communication error with an output module
E106	106	F [1]	Communication error with an input module
E107	107	F [1]	Overload/short circuit at an output module
E108	108	F [1]	Overload/short circuit at an input module
E109	109	F [1]	Low voltage at an output module
E110	110	F [1]	Low voltage at an input module

[1] Error can be configured

Name	No.	Error/warning	Description
CPI module error			
E150	150	F 1)	Communication errors
EasyIP error			
E200	200	F	General EasyIP error
E201	201	F	EasyIP server is not running
E202	202	F	EasyIP communication error
E203	203	W	Invalid EasyIP request
Modbus error			
E300	300	F	General Modbus error
CoDeSys error			
E500	500	F 1)	Error in the CoDeSys® process Detailed information is provided by the system parameter “Current CoDeSys error number” and its error code as per the “errors.ini” file of the Target Support Package
E501	501	F 1)	CAN interface is busy
1) Error can be configured			

Tab. A/3: Error messages

A.6 Address table for EasyIP, Modbus, Telnet and CoDeSys embedded

Read and/or write access to the following entries is possible via EasyIP, Modbus, Telnet and the integrated PLC run-time system CoDeSys.
The entries have the access type “Flag word” (FW) or “String” (STR).

Note
Depending on the programming environment, “Flag word” (FW) may also be labelled “Merkerwort” (German for “flag word”).

A.6.1 Input register

Name	Read/ write	FW	Value type	Permitted values	Remarks
Trigger signal	R/W	0	uint16	0 or 1	
Apply Inputs signal	R/W	1	uint16	0 or 1	
Acknowledge Error signal	R/W	2	uint16	0 or 1	
not used		3	uint16		
not used		4	uint16		
not used		5	uint16		
not used		6	uint16		
not used		7	uint16		
Check program preselection bit 0	R/W	8	uint16	0 or 1	Application of the preselected check program through setting of Apply Inputs signal.
Check program preselection bit 1	R/W	9	uint16	0 or 1	
Check program preselection bit 2	R/W	10	uint16	0 or 1	
Check program preselection bit 3	R/W	11	uint16	0 or 1	

Name	Read/ write	FW	Value type	Permitted values	Remarks
Check program preselection bit 4	R/W	12	uint16	0 or 1	Application of the preselected check program through setting of Apply Inputs signal.
Check program preselection bit 5	R/W	13	uint16	0 or 1	
Check program preselection bit 6	R/W	14	uint16	0 or 1	
Check program preselection bit 7	R/W	15	uint16	0 or 1	

Tab. A/4: Input register

A.6.2 Output register

Name	Read/ write	FW	Value type	Remarks
Ready for operation	R	16	uint16	
Output result good	R	17	uint16	
Output result reject	R	18	uint16	
Output result correctly oriented	R	19	uint16	
Output result incorrectly oriented	R	20	uint16	
not used	R	21	uint16	
Warning (corresponds to LED C red & flashing)	R	22	uint16	
Error status (corresponds to LED C red)	R	23	uint16	

Name	Read/ write	FW	Value type	Remarks
Recognised parts type bit 0	R	24	uint16	Independent of the system parameter “Output module format”. The recognised parts type is binary coded (bit 0 to bit 7): 00000000 = Parts type 1 00000001 = Parts type 2 ... 11111111 = Parts type 256
Recognised parts type bit 1	R	25	uint16	
Recognised parts type bit 2	R	26	uint16	
Recognised parts type bit 3	R	27	uint16	
Recognised parts type bit 4	R	28	uint16	
Recognised parts type bit 5	R	29	uint16	
Recognised parts type bit 6	R	30	uint16	Independent of the system parameter “Output module format”.
Recognised parts type bit 7	R	31	uint16	

Tab. A/5: Output register

A.6.3 Rapid access to the input and output registers

Name	Read/ write	FW	Value type	Permitted values	Remarks
Rapid access to recognised parts type	R	32	uint16		Corresponds to flag word 24 to 31
Rapid access to check program preselection	R/W	33	uint16	0 to 255 when writing	Corresponds to flag word 8 to 15. The Apply Inputs signal must then be set for application

Tab. A/6: Rapid access to the input and output registers

A.6.4 Further system status/system information

Name	**Read/ write**	**FW**	**Value type**	**Remarks**
Error code of the current error	R	100	uint16	0 = No error x = Error number (➔ table in chapter A.5)
Error code of the current warning	R	101	uint16	0 = No error x = Error number (➔ table in chapter A.5)
Device type	R	102	uint16	SBOI-Q-R1B: 701 SBOC-Q-R1B: 702 SBOI-Q-R1C: 703 SBOC-Q-R1C: 704 SBOC-Q-R2B: 705 SBOC-Q-R2C: 706 SBOI-Q-R3B-WB: 707 SBOC-Q-R3B-WB: 708 SBOI-Q-R3C-WB: 709 SBOC-Q-R3C-WB: 710
Version of the major firmware	R	103	uint16	e.g. version 3.2.0.9: high byte = 3, low byte = 2
Version of the minor firmware	R	104	uint16	e.g. version 3.2.0.9: high byte = 0, low byte = 9
Connection to PC	R	130	uint16	0 = Device is not connected to the PC 1 = Device is connected to CheckKon, for example

Tab. A/7: Further system status/system information

A.6.5 System time of the device

Name	Read/ write	FW	Value type	Permitted values	Remarks
Date-Year	R/W	150	uint16	2000 to 9999	The system time must be reset after the device is restarted.
Date-Month	R/W	151	uint16	1 to 12	
Date-Day	R/W	152	uint16	1 to 31	
Time-Hours	R/W	153	uint16	0 to 23	
Time-Minutes	R/W	154	uint16	0 to 59	
Time-Seconds	R/W	155	uint16	0 to 59	

Tab. A/8: System time

A.6.6 Total tolerance of the type in the current check program

Name	Read/ write	FW	Value type	Permitted values	Remarks
Parts type 0	R/W	200	uint16	0 to 20	
Parts type 1	R/W	201	uint16	0 to 20	
Parts type 2	R/W	202	uint16	0 to 20	
Parts type 3	R/W	203	uint16	0 to 20	
Parts type 4	R/W	204	uint16	0 to 20	
Parts type 5	R/W	205	uint16	0 to 20	
Parts type 6	R/W	206	uint16	0 to 20	
Parts type 7	R/W	207	uint16	0 to 20	
Parts type 8	R/W	208	uint16	0 to 20	
Parts type 9	R/W	209	uint16	0 to 20	

Name	Read/ write	FW	Value type	Permitted values	Remarks
Parts type 10	R/W	210	uint16	0 to 20	
Parts type 11	R/W	211	uint16	0 to 20	
Parts type 12	R/W	212	uint16	0 to 20	
Parts type 13	R/W	213	uint16	0 to 20	
Parts type 14	R/W	214	uint16	0 to 20	
Parts type 15	R/W	215	uint16	0 to 20	

Tab. A/9: Total tolerance of the type in the current check program

A.6.7 Basic results of the last check

Name	Read/ write	FW	Value type	Remarks
Check program name	R	234	byte[32]	Name of the check program of the last check
Check program used	R	250	uint16	1 ... 256
Mode used	R	251	uint16	0 = Teach, 2 = Auto
Recognized parts type	R	252	uint16	1 ... 16
Recognised orientation	R	253	uint16	1 ... 8
Auto mode: – Recognition quality Teach mode: – C value	R	254	uint16	Auto mode: – Recognition quality 0 ... 999 – Good part: 0 ... 100 – Bad part: › 100 Teach mode: – Scatter of features C value 0 ... 100
Orientation quality	R	255	uint16	Orientation quality

Name	Read/ write	FW	Value type	Remarks
Total tolerance applied	R	256	uint16	Total tolerance applied in checking part
Part no. low word (LSW)	R	257	uint16	The part no. is binary-coded with 32 bit and divided between 2 FW: FW257 = bit 1 ... 16 (LSW) FW258 = bit 17 ... 32 (MSW) Example: Part no. 500,000 FW257 = 1010 0001 0010 0000 FW258 = 0000 0000 0000 0111
Part no. high word (MSW)	R	258	uint16	
Date – year of capture	R	259	uint16	With trigger signal
Date – month of capture	R	260	uint16	With trigger signal
Date – day of capture	R	261	uint16	With trigger signal
Time – hours of capture	R	262	uint16	With trigger signal
Time – minutes of capture	R	263	uint16	With trigger signal
Time – seconds of capture	R	264	uint16	With trigger signal
Processing time	R	265	uint16	Processing time of the part in ms from trigger signal, up to max 32 s
Number of features actually used	R	266	uint16	1 ... 256 (determined by check program)
Summary of feature results	R	300 - 315	uint16	The results of up to 256 features are summarised at the bit-level to 16 bits each (= 1 flag word). Feature result: 0 = bad; 1 = good

Tab. A/10: Basic results of the last check

A.6.8 Free flag words (non-remanent)

Name	Read/ write	FW	Value type	Remarks
Free flag words (non-remanent)	R/W	320 - 383	uint16	These flag words are 16 bit locations which by default are not assigned any function. These flag words are saved in the Compact Vision System's memory but not permanently, i.e. the data in these flag words are lost when the power is switched off.

Tab. A/11: Free flag words (non-remanent)

A.6.9 Features – results of the last check

Addressing of the features values is summarised in blocks. The blocks start at flag word 10000 and go up in increments of 100.

And so

- the 1st block of feature values starts at 10000
- the 2nd block of feature values starts at 10100
- etc.

The feature values are also stored as strings of characters (String).

In total, there can be no more than 256 blocks of features. The number of features actually used is contained in flag word 266 (➔ Tab. A/10).

Note
- Addressing of the feature values is backward compatible with old PLC programs. The range starting at flag word 400 continues to function, but only for 64 features.
- The feature values (flag word/string display) in the “Parts contour” window of the “CheckKon” program are displayed correctly only from version 4.1, release 03.

For each feature, the respective “Valid Flag” flag word contains information about whether the feature was successfully calculated in the check.
If the result of the feature is a very large negative number, calculation of the feature has probably failed. A request for the “Valid Flag” is unnecessary.

Name	**Read/ write**	**FW** 1)	**Value type**	**Remarks**
Feature value	R	10000	double64	
Tolerance	R	10004	double64	Tolerance applied for feature (incl. tolerance factor)
Feature value, alternative	R	10008	double64	Alternative feature value, e.g. after coordinate transformation
Feature value as text	R	10012	byte[64]	Feature value as character string
Tool name	R	10044	byte[32]	Name assigned by the user (truncated if necessary)
Feature name	R	10060	byte[32]	Fixed feature name (truncated if necessary)
Valid Flag	R	10076	uint16	– 1 = feature was successfully calculated – 0 = calculation failed
Feature type	R	10077	int16	ID of the feature type
Deviation	R	10078	int16	Deviation (-32000 ... 32000), Good part = -100 ... 100

Name	Read/ write	FW [1]	Value type	Remarks
Feature value, before decimal point	R	10079	int16	Pre-decimal places of the feature value as integer – Maximum value: 32767 – Minimum value: -32768
Feature value, after decimal point	R	10080	uint16	Post-decimal places of the feature value x 10,000 as integer [2]
Valid Flag, alternative	R	10081	uint16	Used e.g. after coordinate transformation – 1 = alternative feature value was successfully calculated – 0 = calculation failed
Feature description	R	10082	uint16	ID of feature description
Feature description, alternative	R	10083	uint16	ID of the alternative feature description, e.g. after coordinate transformation

[1] Example for the 1st block of feature values starting at flag word 10000.
[2] Example: 0.99 is stored as 9900.

Tab. A/12: Features – results of the last check

The features available depend on the check program and the device's firmware version.

➔

Note
Additional information on features and tools can be obtained from your local Festo service.

Results in text form are also available as the access type "String" (STR) (➔ chapter A.6.11)

A.6.10 System parameters

Name	Read/ write	FW	Value type	Permitted values
System parameters "..."	R/W	7000	uint16	→ dynamic help in the "System parameters" window in CheckKon

Tab. A/13: System parameters

→

Note
For the devices SBO...-Q-R3, changes to system parameters relevant to creation of the image may only take effect or become apparent in the second image after the changes were made (not evident from the "Live Image" window).

- For devices of type SBO...-Q-R3, always capture an additional (unused) image after making changes to system parameters.

A.6.11 String address table

The string address table is only available for the following protocols:

- EasyIP
- Telnet

Addressing of the feature values starts with string 0 and proceeds in increments of 5.

And so

- the string of the 1st feature starts at 0
- the string of the 2nd feature starts at 5
- the string of the 3rd feature starts at 10
- etc. (➔ Tab. A/14)

Name	**Read/ write**	**STR [1)]**	**Value type**	**Remarks**
Feature value	R	0	string	
Tool name	R	1	string	Name assigned by the user (truncated if necessary)
Feature name	R	2	string	Fixed feature name (truncated if necessary)
Unused	R	3	string	
Unused	R	4	string	

1) Addressing example for the 1st feature from STR0; the 2nd feature starts at STR5.

Tab. A/14: Features – results of the last check as character string

A.7 Address table for EtherNet/IP

A.7.1 Address table for EtherNet/IP – protocol-specific objects

This chapter describes the representation of the Compact Vision System within the EtherNet/IP object model.

The information is in English in order that the original terms of the EtherNet/IP specification can be used unambiguously.

Identity object

Object class: 1

Instances: 1

Attributes

Attr. no.	Access	Description	Type	Remarks
1	Get	Manufacturer ID	UINT	= 26d (Festo)
2	Get	Device type	UINT	= 0d
3	Get	Product code	UINT	= 21314
4	Get	Revision	USINT, USINT	Major Rev., Minor Rev.
5	Get	Status	WORD	
6	Get	Serial number	UDINT	
7	Get	Product name	SHORT_STRING [1]	= "SBOx-Q"
[1] Character string (1 byte per character + 1 byte for string length)				

Services

Service code	Service name	Remarks
01d (01h)	GetAttributeAll	
14d (0Eh)	GetAttributeSingle	
05d (05h)	Reset	Restarts the system

Assembly object

Object class: 4

Instances: 3

The assembly object bundles together attributes of various objects, to allow exchange of data with the objects to take place via a single connection.

Flag words 1...32 are represented in the input assembly. They can be read by the controller. The individual flag words are compressed, each as a single bit.

The output assembly, in contrast, covers flag words 1...16, as only these can also be written to. They, too, are in compressed form.

Instances

Instance	Remarks
769d	Input data
770d	Output data
771d	Configuration data

Attributes

Attr. no.	Inst. no.	Description	Type	Remarks
3	769d	Data (input)	DWORD	
3	770d	Data (output)	WORD	

TCP/IP interface object

Object class: 245

Instances: 1

The TCP/IP interface object is used for configuring the network settings of a device.

Attributes

Attr. no.	Access	Description	Type	Remarks
1	Get	TCP status	DWORD	
2	Get	Configuration capability	DWORD	
3	Get/Set	Configuration control	DWORD	
4	Get	Physical link object	ARRAY	.
5	Get/Set	Interface configuration	ARRAY	
6	Get/Set	Host name	STRING	

Services

Service code	Service name	Remarks
01d (01h)	GetAttributeAll	
14d (0Eh)	GetAttributeSingle	
16d (10h)	SetAttributeSingle	

Ethernet link object

Object class: 246

Instances: 1

The Ethernet link object is used for making further settings for the Ethernet connection.

Attributes

Attr. no.	Access	Description	Type	Remarks
1	Get	Interface speed	UDINT	
2	Get	Interface flags	DWORD	
3	Get	Physical MAC address	ARRAY	

Services

Service code	Service name	Remarks
01d (01h)	GetAttributeAll	
14d (0Eh)	GetAttributeSingle	

A.7.2 Address table for EtherNet/IP – SBO-specific objects

The SBO-specific objects support the services below. Depending on the attribute, the value can also be set/changed using “SetAttibuteSingle”.

Services

Service code	Service name	Remarks
01d (01h)	GetAttributeAll	
14d (0Eh)	GetAttributeSingle	
16d (10h)	SetAttributeSingle	

Input register

Object class: 100

Instances: 1

Attributes

Attr. no.	Access	Description	Type	Permitted values	Remarks
1	Get/Set	Trigger signal	UINT	0/1	
2	Get/Set	Apply Inputs signal	UINT	0/1	
3	Get/Set	Acknowledge Error signal	UINT	0/1	
4		not used	UINT	0/1	
5		not used	UINT	0/1	
6		not used	UINT	0/1	
7		not used	UINT	0/1	
8		not used	UINT	0/1	

Attr. no.	Access	Description	Type	Permitted values	Remarks
9	Get/Set	Check program preselection bit 0	UINT	0/1	
10	Get/Set	Check program preselection bit 1	UINT	0/1	
11	Get/Set	Check program preselection bit 2	UINT	0/1	
12	Get/Set	Check program preselection bit 3	UINT	0/1	
13	Get/Set	Check program preselection bit 4	UINT	0/1	
14	Get/Set	Check program preselection bit 5	UINT	0/1	
15	Get/Set	Check program preselection bit 6	UINT	0/1	
16	Get/Set	Check program preselection bit 7	UINT	0/1	

Output register

Object class: 101

Instances: 1

Attributes

Attr. no.	Access	Description	Type	Permitted values	Remarks
1	Get	Ready for operation	UINT	0/1	
2	Get	Output result good	UINT	0/1	
3	Get	Output result reject	UINT	0/1	
4	Get	Output result correctly oriented	UINT		
5	Get	Output result incorrectly oriented	UINT		
6		not used	UINT		

Attr. no.	Access	Description	Type	Permitted values	Remarks
7	Get	Warning (corresponds to LED C flashing red)	UINT		
8	Get	Error status (corresponds to LED C red)	UINT		
9	Get	Recognised parts type bit 0	UINT	0/1	Independent of the system parameter “Output module format”. The recognised parts type is binary coded (bit 0 to bit 7): 00000000 = Parts type 1 00000001 = Parts type 2 ... 11111111 = Parts type 256
10	Get	Recognised parts type bit 1	UINT	0/1	
11	Get	Recognised parts type bit 2	UINT	0/1	
12	Get	Recognised parts type bit 3	UINT	0/1	
13	Get	Recognised parts type bit 4	UINT	0/1	
14	Get	Recognised parts type bit 5	UINT	0/1	
15	Get	Recognised parts type bit 6	UINT	0/1	
16	Get	Recognised parts type bit 7	UINT	0/1	

Rapid access to the input and output registers

Object class: 102

Instances: 1

Attributes

Attr. no.	Access	Description	Type	Permitted values	Remarks
1	Get	Rapid access to recognised parts type	UINT		Corresponds to flag word 24 to 31
2	Get/Set	Rapid access to check program preselection	UINT	0-255 when written	Corresponds to flag word 8 to 15 The Apply Inputs signal must then be set for application

Further system status/system information

Object class: 103

Instances: 1

Attributes

Attr. no.	Access	Description	Type	Permitted values	Remarks
1	Get	Error code of the current error	UINT	0/1	0 = No error x = Error number (➔ table A.5)
2	Get	Error code of the current warning	UINT	0/1	
3	Get	Device type	UINT	0/1	SBOI-Q–R1B: 701 SBOC-Q–R1B: 702 SBOI-Q–R1C: 703 SBOC-Q–R1C: 704 SBOC-Q–R2B: 705 SBOC-Q–R2C: 706 SBOI-Q–R3B-WB: 707 SBOC-Q–R3B-WB: 708 SBOI-Q–R3C-WB: 709 SBOC-Q–R3C-WB: 710
4	Get	Version of the major firmware	UINT		e.g. version 3.2.0.9: high byte = 3 low byte = 2
5	Get	Version of the minor firmware	UINT		e.g. version 3.2.0.9: high byte = 0 low byte = 9
6	Get	Connection to PC	UINT		0 = Device is not connected to the PC 1 = Device is connected to e.g. CheckKon

System time of the device

Object class: 104

Instances: 1

Attributes

Attr. no.	Access	Description	Type	Permitted values	Remarks
1	Get/Set	Date-Year	UINT	2000-9999	The system time must be reset after the device is restarted.
2	Get/Set	Date-Month	UINT	1...12	
3	Get/Set	Date-Day	UINT	1...31	
4	Get/Set	Time-Hours	UINT	0...23	
5	Get/Set	Time-Minutes	UINT	0...59	
6	Get/Set	Time-Seconds	UINT	0...59	

Total tolerance of the type in the current check program

Object class: 105

Instances: 1

Attributes

Attr. no.	Access	Description	Type	Permitted values	Remarks
1	Get/Set	Parts type 0	UINT	0...20	
2	Get/Set	Parts type 1	UINT	0...20	
3	Get/Set	Parts type 2	UINT	0...20	
4	Get/Set	Parts type 3	UINT	0...20	

Attr. no.	Access	Description	Type	Permitted values	Remarks
5	Get/Set	Parts type 4	UINT	0...20	
6	Get/Set	Parts type 5	UINT	0...20	
7	Get/Set	Parts type 6	UINT	0...20	
8	Get/Set	Parts type 7	UINT	0...20	
9	Get/Set	Parts type 8	UINT	0...20	
10	Get/Set	Parts type 9	UINT	0...20	
11	Get/Set	Parts type 10	UINT	0...20	
12	Get/Set	Parts type 11	UINT	0...20	
13	Get/Set	Parts type 12	UINT	0...20	
14	Get/Set	Parts type 13	UINT	0...20	
15	Get/Set	Parts type 14	UINT	0...20	
16	Get/Set	Parts type 15	UINT	0...20	

Basic results of the last check

Object class: 106

Instances: 1

Attributes

Attr. no.	Access	Description	Type	Remarks
1	Get	Check program name	STRING	Name of the check program of the last check
2	Get	Check program used	UINT	1...256
3	Get	Mode used	UINT	0 = Teach, 2 = Auto
4	Get	Recognized parts type	UINT	1...16
5	Get	Recognised orientation	UINT	1...8
6	Get	Auto mode: Teach mode recognition quality: C value	UINT	Auto mode: – Recognition quality 0...999 – Good part 0...100 – Bad part › 100 Teach mode: – Scatter of features C value 0...100
7	Get	Orientation quality	UINT	Orientation quality
8	Get	Total tolerance applied	UINT	Total tolerance applied in checking part
9	Get	Part no. low word (LSW)	UINT	The part no. is binary-coded with 32 bit and divided between 2 FW: FW257 = bit 1...16 (LSW) FW258 = bit 17...32 Example: Part no. 500,000 FW257 = 1010 0001 0010 0000 FW258 = 0000 0000 0000 0111
10	Get	Part no. high word (MSW)	UINT	

Attr. no.	Access	Description	Type	Remarks
11	Get	Date – year of capture	UINT	With trigger signal
12	Get	Date – month of capture	UINT	
13	Get	Date – day of capture	UINT	
14	Get	Time – hours of capture	UINT	
15	Get	Time – minutes of capture	UINT	
16	Get	Time – seconds of capture	UINT	
17	Get	Processing time	UINT	Processing time of the part in ms from trigger signal (max. 32,000 s)
18	Get	Number of features actually used	UINT	1...256 (defined in check program)
19...34	Get	Summary of feature results	UINT	The results of up to 256 features are summarised at bit level in 16 bits (= 1 flag word) each. Feature result: 0 = bad 1 = good

Free flag words (non-remanent)

Object class: 107

Instances: 1

Attributes

Attr. no.	Access	Description	Type	Remarks
1...64	Get/Set	Flag word	UINT	

System parameters

Object class: 108

Instances: 1

Attributes

Attr. no.	Access	Description	Type	Permitted values	Remarks
Offset address (flag word)	Get/Set	System parameter "..."	UINT		➔ dynamic help in the "System parameters" window in CheckKon

Features – results of the last check

Object class: 109

Instances: 256

Addressing of the features values is summarised in blocks. The blocks begin at instance no. 1 and end at instance no. 256. Thus:

– the 1st block of feature values starts at instance no. 1

– the 2nd block of feature values starts at instance no. 2

– etc.

In total there can be up to 256 feature blocks, corresponding to the 256 instance numbers. The number of features actually used can be found under the basic results of the last check (object class = 106, instance no. = 1, attribute no. 18) (➔ table in the chapter "Basic results of the last check").

For each feature, the respective attribute no. 7 (Valid Flag) contains the information about whether the feature was successfully calculated in the check. If the result of the feature is a very large negative number, calculation of the feature has probably failed. A request for the "Valid Flag" is unnecessary.

The features available depend on the check program and the device's firmware version.

Additional information on features and tools can be obtained from your local Festo service.

Attributes

Attr. no.	Access	Description	Type	Remarks
1	Get	Feature value	LREAL	
2	Get	Tolerance	LREAL	Tolerance applied for feature (incl. tolerance factor)
3	Get	Feature value, alternative	LREAL	Alternative feature value, e.g. after coordinate transformation
4	Get	Feature value as text	STRING	Feature value as character string
5	Get	Tool name	STRING	Name assigned by the user (truncated if necessary)
6	Get	Feature name	STRING	Fixed feature name (truncated if necessary)
7	Get	Valid Flag	UINT	1 = feature was successfully calculated 0 = calculation failed
8	Get	Feature type	UINT	ID of the feature type
9	Get	Deviation	INT	Deviation (-32000...32000) Good part = -100...100

Attr. no.	Access	Description	Type	Remarks
10	Get	Feature value, before decimal point	INT	Pre-decimal places of the feature value as integer – Maximum value: 32767 – Minimum value: –32768
11	Get	Feature value, after decimal point	UINT	Post-decimal places of the feature value x 10,000 as integer
12	Get	Valid Flag, alternative	UINT	1 = feature was successfully calculated 0 = calculation failed (e.g. after coordinate transformation)
13	Get	Feature description	UINT	ID of feature description
14	Get	Feature description, alternative	UINT	ID of alternative feature description (e.g. after coordinate transformation)

Data output

Object class: 768

Instances: 8

Attributes

Attr. no.	Access	Description	Type	Remarks
1	Get	Data output	ARRAY	

You can find a detailed description of how to set up the byte array in the table in the chapter "Data output via EtherNet/IP" (➔ page 4-97).

A.8 Data types

In the following table you will find a comparison of standard data types as per IEC 61131-3 and the data types used in the Compact Vision System.

Data types		Number of bits	Lower limit	Upper limit	Remarks
Compact Vision System	IEC 61131-3				
byte	BYTE	8	0	255	8-bit bit sequence
uint16	WORD	16	0	65535	16-bit bit sequence
uint32	DWORD	32	0	4294967295	32-bit bit sequence
uint64	LWORD [2)]	64	0	$(2^{64}) - 1$	64-bit bit sequence
byte	SINT	8	–128	127	Signed 8-bit integer
int16	INT	16	–32768	32767	Signed 16-bit integer
int32	DINT	32	–2147483648	2147483647	Signed 32-bit integer
int64	LINT [2)]	64	-2^{63}	$(2^{63}) - 1$	Signed 64-bit integer
byte	USINT	8	0	255	Unsigned 8-bit integer
uint16	UINT	16	0	65535	Unsigned 16-bit integer
uint32	UDINT	32	0	4294967295	Unsigned 32-bit integer
uint64	ULINT [2)]	64	0	$(2^{64}) - 1$	Unsigned 64-bit integer
float32	REAL	32	1.17549435e–38	3.40282346e+38	32-bit floating point number, smallest and largest representable number in terms of amount

Data types		Number of bits	Lower limit	Upper limit	Remarks
Compact Vision System	IEC 61131-3				
double64	LREAL [2]	64	2.2250738e–308	1.7976931e+308	64-bit floating point number, smallest and largest representable number in terms of amount
byte[32] char[32]	STRING(31)	n			ASCII string [1] followed by 0x00 → STRING(31) occupies 32 bytes in memory
byte[64] char[64]	STRING(63)				
str	–				Especially for data output via EtherNet/IP; according to IEC 61131-3 corresponds to UDINT + STRING

1) A variable of data type STRING can incorporate any string. The memory space reservation figure in the declaration relates to characters and can be written in round or square brackets. If no value is indicated, the default size is 80 characters. The string length is essentially unlimited, but the string functions can only process lengths of 1-255.

2) In the current version of the CoDeSys run-time system on the Compact Vision System, the 64-bit data types are NOT supported (→ "CoDeSys provided by Festo" online Help [Help] [Content] "Functional overview" section within the "Target system" description).

Tab. A/15: Data types – comparison

The data types for the Compact Vision System are also used in the software packages CheckKon and CheckOpti.

A.9 Programming robot controllers

A.9.1 Telnet communication with an ABB robot

This section shows, on the basis of an example, how the Telnet protocol can be used for communication between an ABB robot controller and the Compact Vision System.

The example shows a program module for communication between an ABB robot control and the Compact Vision System. The basic configurations for the robot controller will not be covered here.

Configuration of the system parameters

– Telnet port = 9999

– Authentication required = No

– Telnet server = On

– Check program preselection via = I/O possibilities

The program code starting from the next page covers:

– Setup of the connection from the robot controller to the Telnet server of the Compact Vision System

– Reading out the Compact Vision System

– Initiation of image capture

– Reading out the result of the 2nd feature (FW 10100)

– Request for a data output

– Switching the current check program

Demo program

```
MODULE CAMERACOM

! DATA DECLARATIONS
! Camera settings
CONST string CameraIPAddress := ”192.168.2.10”;
CONST num CameraTelnetPortNo := 9999;
! General communication declarations
VAR socketdev ComSocket;
VAR string stReceived;
VAR socketstatus Status;

!----------------------------------------------------------------------
! Procedure Main
! Description:
!   This demo program shows a telnet communication between controler
!   an ABB robot and a Festo camera SBO...-Q
!----------------------------------------------------------------------
PROC Main()
  ! Clear the display of the FlexPendant.
  TPErase;
  ! Connect to the cameras telnet server.
  ConnectToCamera;

  ! Get the version from the camera.
  GetVersion;

  ! Let the camera perform a complete evaluation.
  ! This is only possible if the camera is in ready state.
  Wait4Ready;
  ImageTrigger;

  ! Get the first feature result from the camera.
  Wait4Ready;
  GetFlagWord ”10100”;

  ! Get a user defined data collection from the camera.
  ! This data collection has to be defined previously.
  GetDataCollection ”RDC”;

  ! Change check program of the camera.
  Wait4Ready;
  ChangeProgram ”2”;

  ! Let the camera perform a complete evaluation.
  ! This is only possible if the camera is in ready state.
  Wait4Ready;
  ImageTrigger;
```

```
  ! Get the first feature result from the camera.
  Wait4Ready;
  GetFlagWord ”10000”;

  ! Change check program of the camera.
  Wait4Ready;
  ChangeProgram ”1”;

  ! Close telnet connection.
  Wait4Ready;
  SocketClose ComSocket;
ENDPROC

!-------------------------------------------------------------------------
! Procedure Wait4Ready
! Description:
!   Procedure blocks until ’ready’ signal of camera becomes logical high
!  (+24V)
!   The procedure will block infinitely, unless the ’ready’ signal becomes
!  logical high.
!  The answer can be written on the FlexPendant too; uncomment the
!  according line.
!-------------------------------------------------------------------------
PROC Wait4Ready()
  VAR string readyBit := ””;

  WHILE readyBit <> ”1\0D\0A” DO

    CheckConnectionStatus;

    ! Important !
    ! Any camera command must be terminated by ’\0D\0A’
    SocketSend ComSocket \Str := ”RFW 16\0D\0A”;
    SocketReceive ComSocket \Str := readyBit;

    ! TPWrite ”Ready Signal = ” + readyBit;

    IF readyBit <> ”1\0D\0A” THEN
      WaitTime 0.2;
    ENDIF

  ENDWHILE
ENDPROC
```

```
!-------------------------------------------------------------------
! Procedure GetVersion
! Description:
!   Procedure sends a version request to the camera.
!   The answer will be stored in the local variable 'cameraVersion'.
!   The answer will be written on the FlexPendant too.
!-------------------------------------------------------------------
PROC GetVersion()
  VAR string cameraVersion := "";

  CheckConnectionStatus;

  ! Important !
  ! Any camera command must be terminated by '\0D\0A'
  SocketSend ComSocket \Str := "VERSION\0D\0A";
  SocketReceive ComSocket \Str := cameraVersion;

  TPWrite cameraVersion;
ENDPROC

!-------------------------------------------------------------------
! Procedure ImageTrigger
! Description:
!   Procedure sends an IMAGE command to the camera.
!   The command performs a complete evaluation.
!   The telnet server returns '1 OK: operation successful.' if the
!    requested command succeeded, an appropriate error message otherwise.
!-------------------------------------------------------------------
PROC ImageTrigger()
  VAR string triggerReturn := "";

  CheckConnectionStatus;

  ! Important !
  ! Any camera command must be terminated by '\0D\0A'
  SocketSend ComSocket \Str := "IMAGE\0D\0A";

! The 'IMAGE' command sends return value only after image processing has
! finished and the time required depends on the check program
! configuration.
 SocketReceive ComSocket \Str := triggerReturn;

  ! Important !
  ! Any camera response is terminated by '\0D\0A'
  IF triggerReturn <> "1 OK: operation successful.\0D\0A" THEN
    TPWrite triggerReturn;
    stop;
  ENDIF
ENDPROC
```

```
!-----------------------------------------------------------------------
! Procedure ChangeProgram
! Description:
!   Procedure sends an CHANGEPRG command to the camera.
!   The argument progNum is the requested program number.
!   The telnet server returns '1 OK: operation successful.'
!   if the requested command succeeded, an appropriate error message
!   otherwise.
!-----------------------------------------------------------------------
PROC ChangeProgram( string progNum )
  VAR string changeProgReturn := "";

  CheckConnectionStatus;

  ! Important !
  ! Any camera command must be terminated by '\0D\0A'
  SocketSend ComSocket \Str := "CHANGEPRG " + progNum + "\0D\0A";

  SocketReceive ComSocket \Str := changeProgReturn;

  ! Important !
  ! Any camera response is terminated by '\0D\0A'
  IF changeProgReturn <> "1 OK: operation successful.\0D\0A" THEN
    TPWrite changeProgReturn;
    stop;
  ENDIF
ENDPROC

!-----------------------------------------------------------------------
! Procedure GetFlagWord
! Description:
!   Procedure sends a RFW command to the camera.
!   The argument Offset is the flagword address of the data which
!   should be returned.
!   The telnet server returns '1 OK: operation successful.' if the
!   requested command succeeded, an appropriate error message
!   otherwise.
!-----------------------------------------------------------------------
PROC GetFlagWord( string Offset )
  VAR string flagWord := "";
  VAR num Value := 0;
  VAR bool ok := FALSE;

  CheckConnectionStatus;

  ! Important !
  ! Any camera command must be terminated by '\0D\0A'
  SocketSend ComSocket \Str := "RFW " + Offset + "\0D\0A";
```

```
  SocketReceive ComSocket \Str := flagWord;

  ! Important !
  ! Any camera response is terminated by '\0D\0A'
  IF flagword = "-30 ERROR: read offset not valid.\0D\0A" THEN
    TPWrite flagWord;
    stop;
  ENDIF
  TPWrite flagWord;

  ! Converts the answer into a numeric value and writes it on the
FlexPendant.
  ok := StrToVal( flagWord, Value );
  IF ok = TRUE THEN
    TPWrite "Feature 1="\Num := Value;
  ELSE
    TPWrite "StrToVal failed";
    stop;
  ENDIF
ENDPROC

!---------------------------------------------------------------------
! Procedure GetDataCollection
! Description:
!   Procedure sends a RDO (read data output) command to the camera.
!   The first argument DataCollectionName is a user defined name of
!   the data collection.
!   The data collection can be defined by CheckOpti.
!   The telnet server returns '1 OK: operation successful.' if the
!   requested command succeeded, an appropriate error message
!   otherwise.
!       This function needs firmware version 3.5 or greater
!---------------------------------------------------------------------
PROC GetDataCollection( string DataCollectionName )
  VAR rawbytes dataCollectionRaw;

  VAR num float_x;
  VAR num float_y;
  VAR num float_angle;

  CheckConnectionStatus;

  ! Important !
  ! Any camera command must be terminated by '\0D\0A'
  SocketSend ComSocket \Str := "RDO [" + DataCollectionName +
"]\0D\0A";
```

```
  ! Important !
  ! Here we don’t receive ASCII text, but, instead we get a binary
data stream.
  SocketReceive ComSocket \RawData := dataCollectionRaw;

  ! Error handling has been omitted here.

  ! Extract all floating point values from the data container
  ! Offsets of each data element are determined by CheckOpti.
  UnpackRawBytes dataCollectionRaw, 31, float_x, \Float4;
  UnpackRawBytes dataCollectionRaw, 35, float_y, \Float4;
  UnpackRawBytes dataCollectionRaw, 39, float_angle, \Float4;

  TPWrite ”X coordinate = ”, \Num := float_x;
  TPWrite ”Y coordinate = ”, \Num := float_y;
  TPWrite ”Angle = ”, \Num := float_angle;

ENDPROC

!----------------------------------------------------------------------
! Procedure CheckConnectionStatus
! Description:
!   Procedure checks that the socket is still connected, if not, then
!   it will attempt to re-connect. This procedure can be used before
!   communications.
!   An alternative would be to use error handlers.
!----------------------------------------------------------------------
PROC CheckConnectionStatus()
  status := SocketGetStatus(ComSocket);
  TPWrite ”Connection Status: ”\Num:=status;
  IF status <> SOCKET_CONNECTED THEN
    TPWrite ”Camera Not Connected, re-connecting”;
    ConnectToCamera;
  ENDIF
  ERROR
    Stop;
ENDPROC
```

```
!-------------------------------------------------------------------
! Procedure ConnectToCamera
! Description:
!   This procedure establishes an initial connection with a Camera
!   Telnet Server.
!   Telnet servers may perform option negotiation initially with the
!   client
!   this must be detected and a response given before the session is
!   opened.
!   This procedure can be used in other procedures' error handlers.
!-------------------------------------------------------------------
PROC ConnectToCamera()
  VAR num retry_no := 0;
  !   If the socket was already open, close it.
  SocketClose ComSocket;
  SocketCreate ComSocket;
  SocketConnect ComSocket, CameraIPAddress, CameraTelnetPortNo;

  ! Receive first bytes after establishing a connection
  SocketReceive ComSocket \Str := stReceived;

  TPWrite "Connection Established. Received: " + stReceived;

  ! Check for Telnet negotiation, "Interpret As Command", IAC = 0xFF
  ! in the first byte of the first data received after a connection is
  ! made.
  IF StrPart(stReceived, 1, 1) = "\ff" THEN
    ! Send Telnet:  IAC WILL BINARY =  0xFF, 0xFB, 0x00, which is
    !normal for modern Telnet servers and clients.
    SocketSend ComSocket \Str:="\ff\fb\00";
    ! After negotiation, Telnet servers send Server information
    ! strings and often login prompts if authentication is required.
    SocketReceive ComSocket \Str := stReceived;
    TPWrite "Negotiation complete. Received: " + stReceived;
  ENDIF

  ! Test authentication settings for this camera
  IF stReceived <> "20 SBOx-Q-ZEWAS: no authentication requi-
red.\0D\0A" THEN
    TPWrite "Not a Camera response! Check CheckKon settings!";
    Stop;
  ENDIF
  ! Error handler if the Socket times out or is closed by the server
  ERROR
    IF ERRNO = ERR_SOCK_TIMEOUT THEN
    ! Retry the above Socket call that timed out
      IF retry_no < 3 THEN
        TPWrite "Connection timeout. Retry = "\Num:=retry_no;
```

```
        WaitTime 1;
        retry_no := retry_no + 1;
        RETRY;
      ELSE
        ! Retry failed, log and raise the error
        TPWrite "Connection to camera failed after retry";
        RAISE;
      ENDIF
    ELSEIF ERRNO = ERR_SOCK_CLOSED THEN
      ! If the socket has been closed by the server then one can only
      ! return
      RETURN;
    ENDIF
  ENDPROC

ENDMODULE
```

A.9.2 Telnet communication with a KUKA robot using XML

This section shows, on the basis of an example, how XML and the Telnet protocol can be used for communication between a KUKA robot controller and the Compact Vision System.

In this example, a “Telnet – XML” data output is defined in CheckOpti that contains the x and y coordinates and the rotational angle of a detected object.

Preparations:

– In the CheckOpti data output settings, the communication name “Positions” is assigned (➔ section “Structure specification when receiving data”).

– The corresponding structure file is saved as “SBO_Sensor.xml” and then transmitted to the KUKA robot controller.

Configuration of the system parameters:

– XML Telnet port = 9997

– XML authentication required = No

– XML Telnet server = On

– Check program preselection via = I/O possibilities

➔

Note
On the Kuka robot controller the “KUKA.Ethernet KRL XML” package must be installed in order for XML-formatted data packages to be received from the Compact Vision System.

KUKA.Ethernet KRL XML is a technology package that is loadable at runtime and has the following functions:

– Transfer of data between a robot controller and an external system

– Transmitting and receiving data within a KRL program

Two configuration files need to be installed on the robot controller for communication between the KUKA robot controller and the Festo Compact Vision System:

– XmlApiConfig.xml

– [Channel or sensor name].xml
  e.g. “SBO_Sensor.xml”

Because the Compact Vision System cannot evaluate any commands in XML format, the file [Channel or sensor name]+.xml is not needed on the robot controller.

Communication parameters (IP address, port, channel name)

The file “XmlApiConfig.xml” is for configuration of the parameters for communication with the Compact Vision System.

– IP address
  e.g. IP = “192.168.2.10”

– Port number of the XML Telnet server
  e.g. Port = “9997”

– Channel or sensor name
  e.g. SensorName = “SBO_Sensor”

Note
- Assign a unique channel or sensor name.

The name of the structure file will be derived from this (➔ section “Structure specification when receiving data”).

- Ensure that the file “XmlApiConfig.xml” is saved on the Kuka controller in the directory C:KRCRoboterInit.

Example of a “XmlApiConfig.xml” file

```
<?xml version=”1.0”?>
<!-- KUKA Roboter GmbH                                                    -->
<!--                                                                      -->
<!-- ’InitOnce’   use false only                                          -->
<!-- ’Channel’    represents a connection to a Sensor. Every              -->
<!--              channel has the following parameters                    -->
<!-- ’SensorName’ is the name you give to a sensor. In KRL it             -->
<!--              serves as a handle to this channel. If the              -->
<!--              sensor name, for example, is ’StackCam’ there           -->
<!--              has to be a file called ’StackCam.XML’ which            -->
<!--              holds the information of the associated ring            -->
<!--              buffers (XML tags)                                      -->
<!-- ’SensorType’ is the type or the model name of a sensor.              -->
<!-- ’TCP_IP’     holds information about the type of connection,         -->
<!--              TCP/IP. No other then TCP_IP is realized yet.           -->
<!-- ’IP’         holds the IP of the sensor.                             -->
<!-- ’Port’       holds the sensor port number of the port you want-->
<!--              to connect to.                                          -->
<!-- ’Route’      if set to ’true’ the connection will use                -->
<!--              ROUTE.EXE. If set to ’false’ it will establish          -->
<!--              a direct connection.                                    -->
<!-- ’MapPort’    up to now should be always the same as ’Port’.          -->
<!--                                                                      -->
<XmlApiConfig xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
xsi:
  noNamespaceSchemaLocation=”XMLCommunicationSetup.xsd”>
<!--   -->
<!--   -->
<XmlApiParam InitOnce=”false”/>
<!--   -->
<!-- USE DEMOSERVER -->
<Channel SensorName=”SBO_Sensor” SensorType=”SBO”>
  <TCP_IP IP=”192.168.2.10” Port=”9997” Route=”false” MapPort=”9997”/>
</Channel>
<!-- END DEMO -->
</XmlApiConfig>
```

Structure specification when receiving data

For the robot controller to receive data from an external system, a structure file (XML format) must be defined for each channel in the directory C:KRCRoboterInit for data reception.

The structure file can be created in CheckOpti:

- Click on the “Create” button in the “Result” register of the “Data output” register.
- Use the channel or sensor name as the file name, e.g. SBO_Sensor.xml (➔ section “Communication parameters”). Use correct capitalisation.

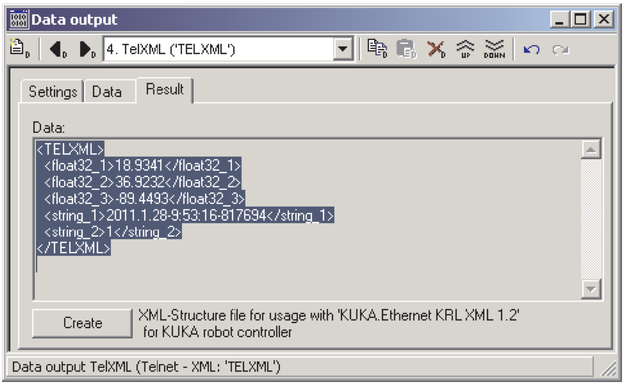


Fig. A/2: Creating a structure file

This file on the KUKA robot controller contains all of the structure specifications necessary in order for the Compact Vision System to receive data.

Example of the “SBO_Sensor.xml” structure file

```
<Elements>
  <Element Tag=”Camera”                        Type=”STRUCTTAG”
   Stacksize=”5”/>
  <Element Tag=”Camera.CommandResultCode”   Type=”INTEGER”
   Stacksize=”5”/>
  <Element Tag=”Camera.CommandResultText”   Type=”STRING”
   Stacksize=”5”/>
  <Element Tag=”Camera.DataResult_RFW”      Type=”STRING”
   Stacksize=”5”/>
  <Element Tag=”Camera.DataResult_RSTR”     Type=”STRING”
   Stacksize=”5”/>
  <Element Tag=”Camera.DataResult_RNV”      Type=”STRING”
   Stacksize=”5”/>
  <Element Tag=”Camera.DataResult_RDO”      Type=”STRUCTTAG”
   Stacksize=”5”/>
<Element Tag=”Camera.DataResult_RDO.Positions” Type=”STRUCTTAG”
   Stacksize=”5”/>
<Element Tag=”Camera.DataResult_RDO.Positions.float32_1” Type=”REAL”
   Stacksize=”5”/>
<Element Tag=”Camera.DataResult_RDO.Positions.float32_2” Type=”REAL”
   Stacksize=”5”/>
<Element Tag=”Camera.DataResult_RDO.Positions.float32_3” Type=”REAL”
   Stacksize=”5”/>
</Elements>
```

Demo program

The following program code shows as an example a function library for communication between the KUKA robot controller and the Compact Vision System via XML Telnet.

Note
Commands from the robot controller to the Compact Vision System must NOT be XML-formatted.

Responses from the Compact Vision System to the robot controller, conversely, are always XML-formatted (➔ chapter 4.10.10, Structure specification file).

The program code here is limited to the content relevant for the demo.

```
DEF u_CamComm( )
;=====================================================================
; Program: Camera control library
;=====================================================================
END

;=====================================================================
; Image acquisition and position evaluation
;=====================================================================
GLOBAL DEFFCT CamCommData CamGetPartPos(sProgNumber[]:IN)
DECL CHAR sProgNumber[]

;---------------------------------------------------------------------
  ; General variable initialization
;---------------------------------------------------------------------
  m_bNew = TRUE

  m_Result.bValid = FALSE
  m_Result.nX  = 0.0
  m_Result.nY  = 0.0
  m_Result.nAngle = 0.0

;---------------------------------------------------------------------
  ; Camera control commands
;---------------------------------------------------------------------
  m_bOK = StrClear(m_sendProg[])
  m_sendProg[] = ”CHANGEPRG ”

  i = StrAdd(m_sendProg[],sProgNumber[])

  m_sendImage[] = ”image”
  m_sendRDO[] = ”RDO [Positions]”

;---------------------------------------------------------------------
  ; Channel name (sensor name, camera name)
  ; (see “XmlApiConfig.XML” file)
;---------------------------------------------------------------------
  m_sensorname[]=”SBO_Sensor”
```

```
;---------------------------------------------------------------------
  ; Element names with prefixed channel name
  ; (see "SBO_Sensor.xml" file)
;---------------------------------------------------------------------
  m_rcvCmdResultCamera[] = "SBO_Sensor.Camera"
  m_rcvCmdResultCode[] = "SBO_Sensor.Camera.CommandResultCode"
  m_rcvCmdResultText[] = "SBO_Sensor.Camera.CommandResultText"
  m_rcvCmdResultRFW[] = "SBO_Sensor.Camera.CommandResult_RFW"
  m_rcvCmdResultRSTR[] = "SBO_Sensor.Camera.CommandResult_RSTR"
  m_rcvCmdResultRNV[] = "SBO_Sensor.Camera.CommandResult_RNV"
  m_rcvCmdResultPosX[] = "SBO_Sensor.Camera.DataResult_RDO.Posi-
tions.float32_1"
  m_rcvCmdResultPosY[] = "SBO_Sensor.Camera.DataResult_RDO.Posi-
tions.float32_2"
  m_rcvCmdResultAngle[] = "SBO_Sensor.Camera.DataResult_RDO.Posi-
tions.float32_3"

;---------------------------------------------------------------------
  ; Initially close XML telnet channel
;---------------------------------------------------------------------
  m_bOk = EKX_close(m_sensorname[])
;---------------------------------------------------------------------
  ; Open XML telnet channel
;---------------------------------------------------------------------
  m_nErrCode = EKX_open(m_sensorname[])
  EKX_handleerror(m_nErrCode)

;---------------------------------------------------------------------
  ; Wait for camera data
  ; If XML telnet channel could be opened successfully a
status message should be received here
;---------------------------------------------------------------------
  m_bOk = EKX_WaitForSensorData(0, m_rcvCmdResultCamera[], 10000)
  IF m_bOk == FALSE THEN
    HALT
    wait for false
  ENDIF


;---------------------------------------------------------------------
  ; Evaluate command result code from camera
;---------------------------------------------------------------------
  m_bOk = EKX_GetIntegerElement(0, m_rcvCmdResultCode[],
m_nResultCode, m_bNew)
  IF m_bOk == FALSE THEN
    HALT
    wait for false
  ENDIF
```

```
  ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  ; Check if received command result code is equal to 20
  ; (no authentication required)
  ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  IF m_nResultCode <> 20 THEN
    HALT
    wait for false
  ENDIF

;-------------------------------------------------------------------
  ; Switch check program
;-------------------------------------------------------------------
  m_nErrCode = EKX_writeline(m_sensorname[], m_sendProg[])
  IF m_nErrCode == eioc_error THEN
    HALT
    wait for false
  ENDIF

;-------------------------------------------------------------------
  ; Wait for response (switch check program)
;-------------------------------------------------------------------
  m_bOk = EKX_WaitForSensorData(0, m_rcvCmdResultCamera[], 10000)
  IF m_bOk == FALSE THEN
    HALT
    wait for false
  ENDIF

;-------------------------------------------------------------------
  ; Check response message (switch check program)
;-------------------------------------------------------------------
  m_bOk = EKX_GetIntegerElement(0, m_rcvCmdResultCode[],
m_nResultCode, m_bNew)
  IF m_bOk == FALSE THEN
    HALT
    wait for false
  ENDIF

  ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  ; Check if received command result code is equal to 1
(operation successful)
  ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  IF m_nResultCode <> 1 THEN
    HALT
    wait for false
  ENDIF
```

```
;---------------------------------------------------------------------
  ; Trigger image acquisition
;---------------------------------------------------------------------
  m_nErrCode = EKX_writeline(m_sensorname[], m_sendImage[])
  IF m_nErrCode == eioc_error THEN
    HALT
    wait for false
  ENDIF

;---------------------------------------------------------------------
  ; Wait for response (trigger image acquisition)
;---------------------------------------------------------------------
  m_bOk = EKX_WaitForSensorData(0, m_rcvCmdResultCamera[], 10000)
  IF m_bOk == FALSE THEN
    HALT
    wait for false
  ENDIF

;---------------------------------------------------------------------
  ; Check response message (trigger image acquisition)
;---------------------------------------------------------------------
  m_bOk = EKX_GetIntegerElement(0, m_rcvCmdResultCode[],
m_nResultCode, m_bNew)
  IF m_bOk == FALSE THEN
    HALT
    wait for false
  ENDIF

  ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  ; Check if received command result code is equal to 1
(operation successful)
  ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  IF m_nResultCode <> 1 THEN
    HALT
    wait for false
  ENDIF

;---------------------------------------------------------------------
  ; Retrieve RDO-data from camera
;---------------------------------------------------------------------
  m_nErrCode = EKX_writeline(m_sensorname[], m_sendRDO[])
  IF m_nErrCode == eioc_error THEN
    HALT
    wait for false
  ENDIF
```

```
;--------------------------------------------------------------------
  ; Wait for response
;--------------------------------------------------------------------
  m_bOk = EKX_WaitForSensorData(0, m_rcvCmdResultCamera[], 10000)
  IF m_bOk == FALSE THEN
    HALT
    wait for false
  ENDIF

;--------------------------------------------------------------------
  ; Check if received RDO-data are valid
;--------------------------------------------------------------------
  m_bOk = EKX_GetIntegerElement(0, m_rcvCmdResultCode[],
m_nResultCode, m_bNew)
  IF m_bOk == FALSE THEN
    HALT
    wait for false
  ENDIF

  IF m_nResultCode < 0 THEN
    HALT
    wait for false
  ENDIF

;--------------------------------------------------------------------
  ; Read coordinates of detected part
;--------------------------------------------------------------------
  m_bOk = EKX_GetRealElement(0, m_rcvCmdResultPosX[], m_Result.nX,
m_bNew)
  IF m_bOk == FALSE THEN
    HALT
    wait for false
  ENDIF

  m_bOk = EKX_GetRealElement(0, m_rcvCmdResultPosY[], m_Result.nY,
m_bNew)
  IF m_bOk == FALSE THEN
    HALT
    wait for false
  ENDIF

  m_bOk = EKX_GetRealElement(0, m_rcvCmdResultAngle[],
m_Result.nAngle, m_bNew)
  IF m_bOk == FALSE THEN
    HALT
    wait for false
  ENDIF
```

```
;----------------------------------------------------------------------
  ; Check if coordinates are valid
;----------------------------------------------------------------------
  m_Result.bValid = TRUE            ; Default

  ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  ; Is x-coordinate within limits?
  ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  IF m_Result.nX < XCAM_MIN.x THEN
    m_Result.bValid = FALSE
  ENDIF

  IF m_Result.nX > XCAM_MAX.x THEN
    m_Result.bValid = FALSE
  ENDIF

  ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  ; Is y-coordinate within limits?
  ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  IF m_Result.nY < XCAM_MIN.y THEN
    m_Result.bValid = FALSE
  ENDIF

  IF m_Result.nY > XCAM_MAX.y THEN
    m_Result.bValid = FALSE
  ENDIF

  ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  ; Normalize angle
  ; - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
  IF m_Result.nAngle < -180 THEN
    m_Result.nAngle = m_Result.nAngle + 360
  ENDIF

  IF m_Result.nAngle > 180 THEN
    m_Result.nAngle = m_Result.nAngle - 360
  ENDIF
```

```
;---------------------------------------------------------------------
  ; Close XML telnet channel
;---------------------------------------------------------------------
  m_bOk = EKX_close(m_sensorname[])
  IF m_bOk == FALSE THEN
    HALT
    wait for false
  ENDIF

;---------------------------------------------------------------------
  ; Return results to calling function
;---------------------------------------------------------------------
  RETURN m_Result

ENDFCT
```

Index

Appendix B

Table of contents

B. Index B-1

A

Abbreviations XIII
Accessories 1-13
Activity LED (C) 5-4
Adapter kit 2-6
Amplification XIII
Aperture . . . . XIII, 4-17
Assign IP address automatically 4-13
Auto MDI-X XIII

B

Back light 1-19

C

C-Mount 1-14
Cable 3-12
Calculation time 4-20
Camera image 4-19
CAN interface 3-15
CANopen XIII
CE mark VII
Certification VII
Characteristic 4-110
Check area 4-16
Check feature 4-19
Check program . . . . XIII, 4-110

CheckKon XIII
CheckOpti . . . . XIII, 1-12
CMOS sensor XIII
Coaxial vertical light 1-19
CoDeSys pbF XIII
Commissioning 4-3
Condition-controlled XIII
Connection example 3-9
Contrast 4-18
CP cable XIII
CP connection XIII
CP master XIV
CP modules XIV
CP modules address assignment 4-60
CP node XIV
CP string XIV
CP system XIV
CP valve terminal XIV
CPI modules XIV
CPI system XIV
CPX modules XIV
CPX Terminal XIV
CS-Mount 1-14

D

Data output XIV
Depth of focus XIV
Device as CPI module 4-57, 4-63
DHCP server 4-13
Diagnostics 5-3
Digital input (I) XV
Digital output (O) XV
Dimensions 2-5
Display and connecting elements 1-9, 1-10
Dome light 1-19
Dovetail guide 2-4

E

EasyIP 4-65
Edge-controlled XIV
Electrical connections 3-6
EMERGENCY STOP concept 3-4
Error elimination 5-8
Ethernet XIV
Ethernet address A-4
Ethernet connection 3-12
Ethernet interface 3-11, 3-13
Ethernet traffic LED (B) 5-4
EtherNet/IP XIV
Evaluation mode 4-21
Exposure time . . . . XV, 4-16
External lighting 1-18
Extraneous light 1-18

F

Falling edge XV
Feature XV
Fieldbus nodes XV
Fixed focal length 1-16
Fixed frame rate 4-21, 4-39
Flat angle light 1-19
Focal depth XV
Focal length .... XV, 1-15
Focus 4-17
Free run 4-21, 4-28

G

Gateway A-5

H

Higher-order controller (PLC) 4-44

I

I module XV
I/O expansion 4-52
I/O modules XV
I/Os XV
Image field area 4-16
Image optimisation 4-20
Image sharpness 4-17
Input “Acknowledge Error signal” 4-49
Input “Apply Inputs signal” 4-47

Input “Trigger signal” 4-47
Inputs with signal function 4-46
Intended use VI
Internal lighting 1-18
IP address A-4

L

LEDs 5-3
Lens 1-14, 2-7
Light source 1-18
Lighting 4-18
Lighting control 4-16
Live image 4-19
Logic 0 XV
Logic 1 XV

M

MAC ID A-4
Main level 1-15
Modbus TCP XV
Mounting 2-3, 2-6
Multicast 4-11

N

Net classes A-5
Net mask A-5
Network settings 4-5, 4-11
Networking and control 1-4

O

O module XV
Object distance 1-15
Operating status LED (A) 5-3
Operating voltage supply 3-6
Operating voltage supply and digital I/Os 3-8
Optics 4-17
Other inputs 4-46
Output duration 4-23
Output LED (D) 5-5
Output time 4-23

P

Pictograms XI
PLC/IPC XV
Port enable 4-9
Power supply unit 3-5
Preprocessing 4-19
Protective barrel 2-7
Protective foil 2-8
Protective glass 2-8

R

Rising edge XV
Rod light 1-19

S

Safety instruction VIII
SBO-DeviceManager . . . . XV, 1-12, 4-11
Search function 4-11
Sensor amplification . . . . XVI, 4-16
Sensor size 1-15
Service IX
Setting the aperture 4-18
Shield tube 2-7
Siemens star A-7
Signal cables 3-4
Signal sequence 4-25, 4-31, 4-41
Size of field of view 1-15
Software packages 1-12
Specifying IP address 4-13
Status display 5-3
String allocation XVI
Supply cables 3-4

T

Target group IX
TCP/IP XVI
Technical data A-8
Telnet 4-66
Telnet under Windows Vista 4-68
Text designations XI
Tools XVI
Triggered 4-21, 4-22
TSP XVI

U

User instructions X

V

Variants of the Compact Vision System 1-6, 1-7
Versions XII

W

Working distance 1-15

X

XML XVI

Z

Zoom 1-15