

 SENSOPART

VISOR[®]
Communications manual
Software version 2.2

Copyright (English)

No part of this document may be reproduced, published, or stored in databases or information retrieval systems in any form – even in part – nor may illustrations, drawings, or the layout be copied without prior written permission from SensoPart Industriesensorik GmbH.

We accept no responsibility for printing errors or mistakes which occurred in drafting these document. Subject to delivery and technical alterations.

First publication 01 / 2019

SensoPart Industriesensorik GmbH
Nägelseestr. 16
79288 Gottenheim

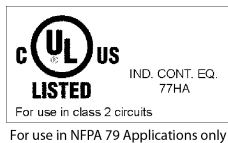


Table of contents

1 Information on this document	5
1.1 Explanation of symbols	5
1.2 Additional documents	6
1.3 Document version	6
2 Network connection	7
2.1 Integrating the VISOR® into the network / gateway	7
2.2 Network connection: Direct connection	8
2.3 Network connection: Connection via network	9
2.4 Used Ethernet ports	10
2.5 Access to VISOR® through network	11
2.6 Access to VISOR® through the Internet / World Wide Web	12
2.7 Electrical connection of VISOR® in the network	13
3 Configuration VISOR® vision sensor	15
4 Ethernet TCP/IP, port 2005 / 2006	19
4.1 Example: Data output from VISOR® to PC / PLC	19
4.2 Example: Commands (requests) from PC / PLC VISOR®	23
4.3 Example: Job change from PC / PLC to VISOR®	24
4.4 Example Beckhoff CX 1020	27
4.5 Example Siemens S7	27
5 Service / Visualization	29
5.1 Backup creation	29
5.2 Visualization	29
6 VISOR® telegrams for PROFINET and EtherNet/IP	31
6.1 Module 1: "Control" (From PLC to VISOR®)	31
6.2 Module 2: "Status" (from VISOR® to PLC)	33
6.3 Module 3: "Data" (from VISOR® to PLC)	36
6.4 Module 4: "Request" (From PLC to VISOR®)	37
6.5 Module 5: "Response" (from PLC to VISOR®)	38
6.6 Start / end criteria for each telegram	39
7 Timing diagrams for VISOR® communication	41
8 Request sequences	45
8.1 Trigger Request Sequence	46
8.2 Change job request sequence	47
8.3 Switch to Run sequence	48
8.4 Sequence for requests via request/response module	49
9 PROFINET	51
9.1 Siemens S7-1200 TIA 12 configuration example	51

9.1.1 Create new project	51
9.1.2 Selecting the GSD file	51
9.1.3 Adding the VISOR® vision sensor to the project	52
9.1.4 Writing a name to VISOR®	57
9.1.5 Load the project onto the PLC	58
9.1.6 Mapping of output data	58
9.2 PLC example programs	60
10 EtherNet/IP	65
10.1 Rockwell CompactLogix™ configuration example	65
10.2 Installation of EDS file	67
10.3 Create module	73
10.3.1 Selection via hardware catalog (with EDS file)	73
10.3.2 Using a Generic Device (without EDS file)	77
10.4 Load the project onto the PLC	81
10.5 Mapping of output data	83
10.6 PLC example programs	85
11 Telegrams and data output	87
11.1 Overview telegrams	87
11.2 Telegrams: Availability and supported interfaces	91
11.3 Error codes	94
11.4 Description Telegrams ASCII	96
11.4.1 General	96
11.4.2 Control	97
11.4.3 Job settings	105
11.4.4 Calibration	136
11.4.5 Visualization	159
11.4.6 Service (available only on port 1998 and in ASCII format)	161
11.4.7 Data output ASCII	169
11.5 Description Telegrams BINARY	185
11.5.1 General	185
11.5.2 Control	186
11.5.3 Job settings	195
11.5.4 Calibration	224
11.5.5 Visualization	244
11.5.6 Data output BINARY	246

1 Information on this document

1.1 Explanation of symbols

Warnings



CAUTION / WARNING / DANGER

This symbol is used to indicate a potentially hazardous situation that, if not avoided, could result in death or serious injury.



WARNING

This symbol is used to indicate potentially hazardous situations arising from laser beams.



ATTENTION:

This symbol is used to indicate text that must be observed without fail. Failure to do so may result in bodily injury or property damage.



NOTE:

This symbol is used to highlight useful tips and recommendations, as well as information intended to help ensure efficient operation.

Detectors



Pattern matching



Contour



Contrast



Brightness



Gray



Caliper



BLOB



Contour 3D



Barcode



Datacode



OCR



Color Value



Color List



Color Area



Result processing

Alignment



Alignment

Includes the position detectors: Contour matching, Pattern matching, and Edge detector

1.2 Additional documents

The following documents for the VISOR® vision sensor are available for download in the Download area of the SensoPart website.

- VISOR® User Manual
- VISOR® Communications manual
- VISOR® Operating manual

Furthermore, these documents are part of the software installation and can be found in the subfolder "\\Documentation\\", as well as via the Windows Start menu.

1.3 Document version

This manual describes the VISOR® software version 2.2.

Documents for the previous software versions (< 2.2) can be found in the download area of the SensoPart homepage (www.sensopart.com).

2 Network connection

2.1 Integrating the VISOR® into the network / gateway

SensoFind/Active sensors will show a list with all the VISOR® vision sensors that are found on the same network segment on the PC on which is running SensoFind. To update the list, press the "Find" button, e.g. for sensors that were only activated after viewing SensoFind.

For sensors which are installed in the network but are located in a different network segment via a gateway, please enter the corresponding sensor IP address under "Add active sensor" and press the button "Add". The corresponding sensor will now also appear in the "Active sensors" list, and you will be able to access it and work with it.

2.2 Network connection: Direct connection

Establishing a direct Ethernet connection between the VISOR® vision sensor and the PC

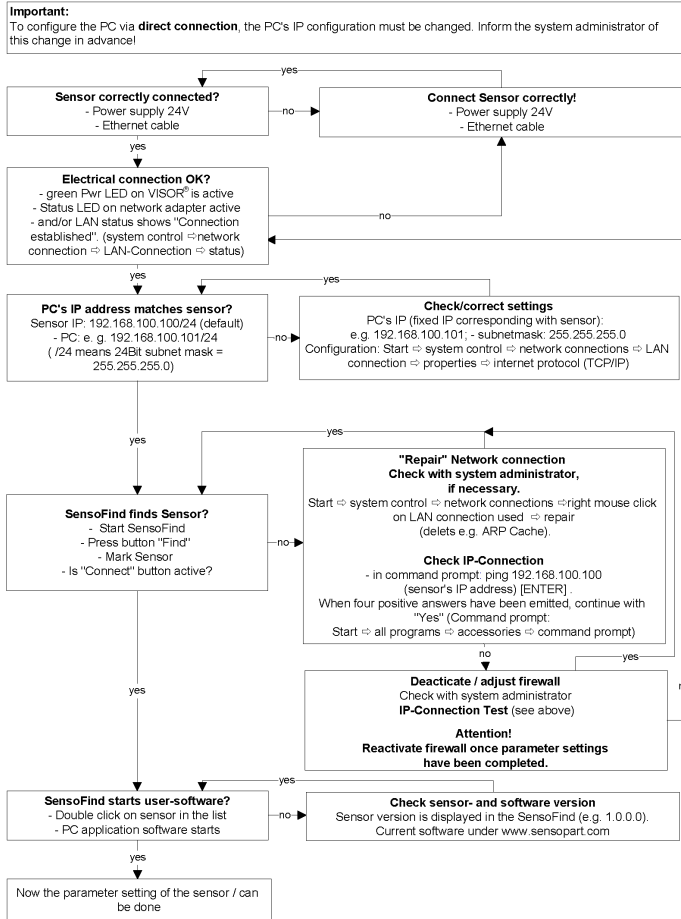


Fig. 1: Direct connection sensor / PC, procedure and troubleshooting

2.3 Network connection: Connection via network

Establishing an Ethernet connection between the VISOR® vision sensor and the PC through a network.

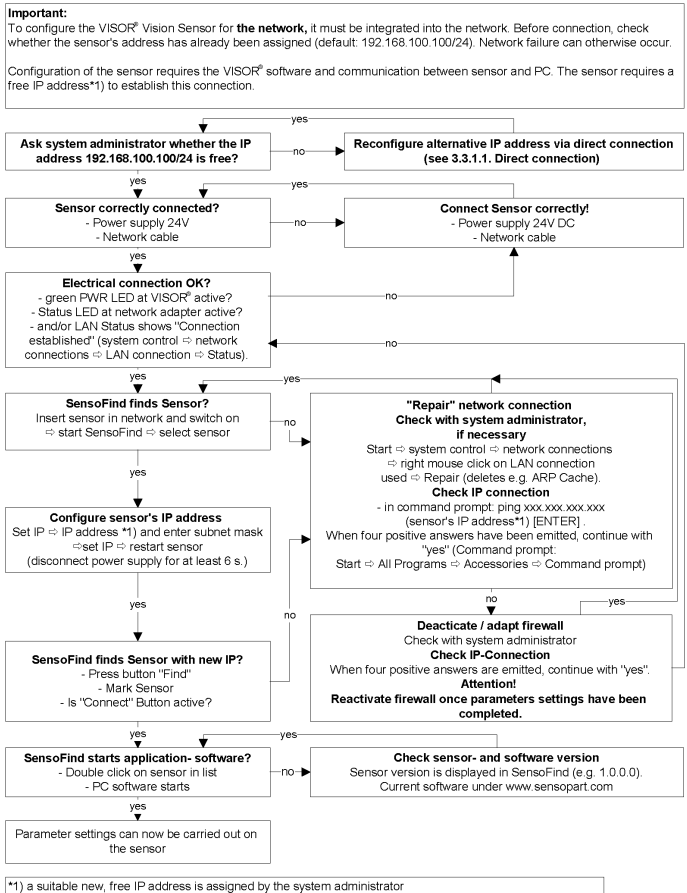


Fig. 2: Connection via network sensor / PC, procedure and troubleshooting

2.4 Used Ethernet ports

If you are integrating the VISOR® into a network, make sure that an admin opens the following ports if necessary. This is only the case if these ports were previously explicitly blocked in the company network or by a firewall installed on the PC.

The following ports are used for communications between the VISOR® software (PC) and the VISOR®:

- Port 2000, TCP
- Port 2001, UDP Broadcast (to find sensors via SensoFind)
- Port 2002, TCP
- Port 2003, TCP
- Port 2004, TCP

The following ports are used for communications between the PLC (PLC or control PC) and VISOR® vision sensor:

Process interfaces:

- Ethernet
 - Port 2005, TCP (Implicit results, i.e. user-configured result data)
 - Port 2006, TCP (Explicit requests, e.g. trigger or job switch)
- EtherNet/IP:
 - Port 2222, UDP
 - Port 44818, TCP
- PROFINET:
 - Port 161, UDP
 - Port 34962, UDP
 - Port 34963, UDP
 - Port 34964, UDP
- Service:
 - Port 22, TCP
 - Port 1998, TCP
- SensoWeb:
 - Port 80



NOTE:

If Ports 2005 or 2006 are changed in the configuration software, they must also be changed accordingly in the firewall by an administrator.

2.5 Access to VISOR® through network

Exemplary values for IP, etc.

Access to VISOR® 1 from PC 1, if on the same subnet

- Via SensoFind (/find)

Access to VISOR® 2 from PC 1, if on a different subnet

Only if:

- Gateway is set correctly in Sensor 2 (here to 192.168.30.1) - and
- in SensoFind via Add IP, the sensor IP of Sensor 2 is set correctly
> after this, VISOR® 2 will also appear in the "Active sensors" list in SensoFind!

PC 1

IP: 192.168.20.x
Subnetmask: 255.255.255.0
Gateway: 192.168.20.1

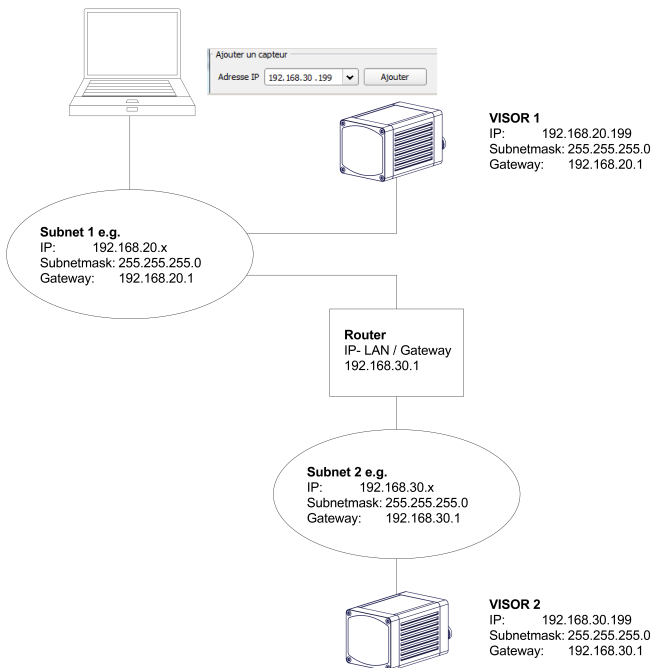


Fig. 3: Access to VISOR® through network, same or other subnet

2.6 Access to VISOR® through the Internet / World Wide Web

Exemplary values for IP, etc.

Access from PC 1 (company network 1), through the Word Wide Web, to company network 2 to VISOR® 1.

1. On PC 1 (company network 1 SensoFind) enter and add the IP WAN of Router 2 (company network 2) under "Add active sensor" in (here in this example: 62.75.148.101)
2. On router 2, open the ports that the sensor will be using (please refer to section: [Used Ethernet ports](#)). See Chapter:

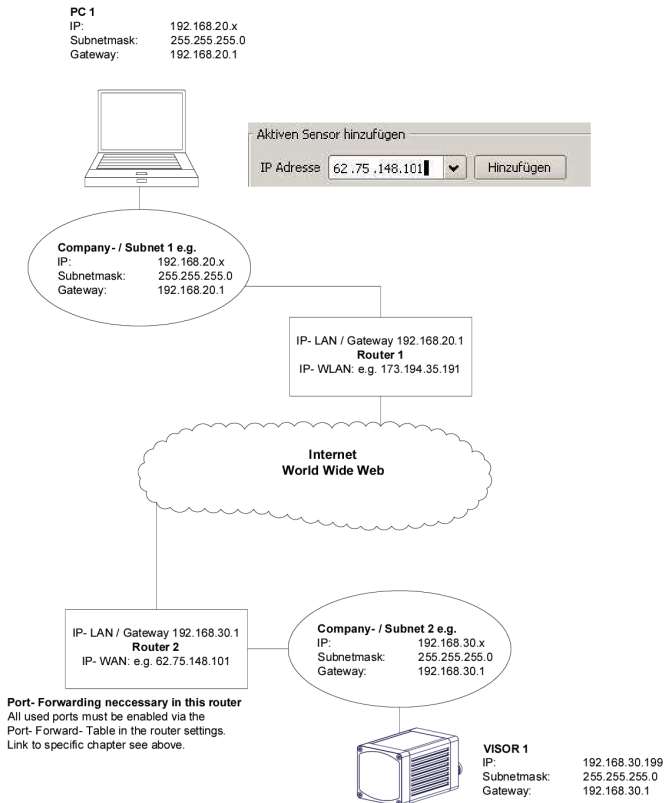


Fig. 4: Access to VISOR® through the Internet / World Wide Web

2.7 Electrical connection of VISOR® in the network

The VISOR® vision sensor is connected to the network through a switch.

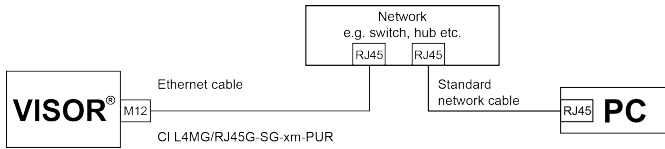


Fig. 5: Electrical connection of VISOR® in the network

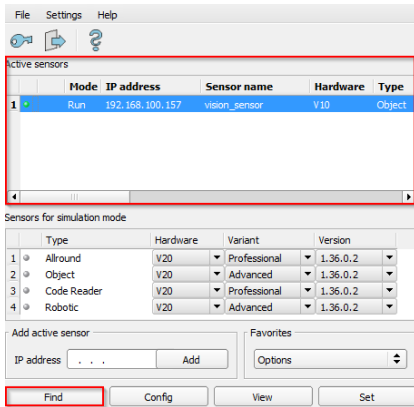
3 Configuration VISOR® vision sensor

In order to configure the vision sensor, follow the steps below.

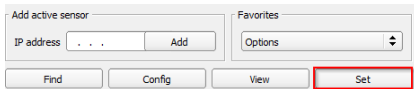
Settings in SensoFind



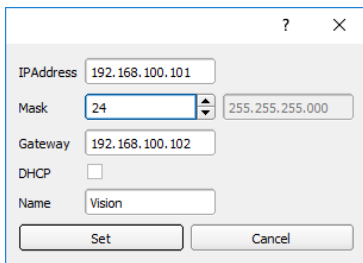
1. Start the VISOR® software. SensoFind is opened.



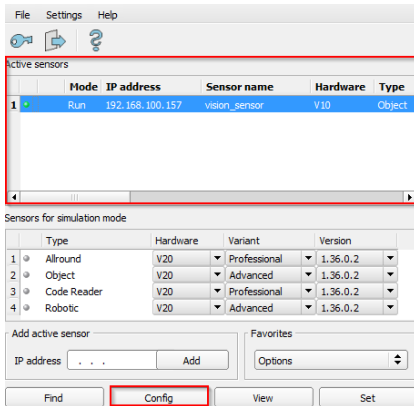
2. Click on the "Find" button. The vision sensor will be listed in the "Active sensors" window.



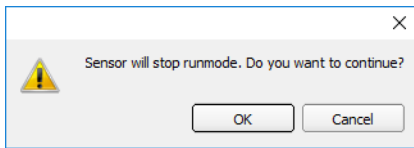
3. Click on the "Set" button. The dialog box for configuring the IP address and the sensor name will appear.



4. Assign an IP address and a name to the sensor.
5. Click on the "Set" button. The IP address and the name have now been updated.

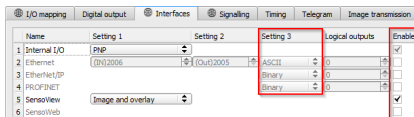


- Open SensoConfig by selecting the sensor you want and then clicking on the "Config" button.



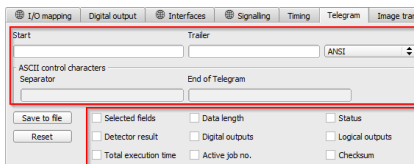
- Confirm the following dialog box with "OK" to stop SensoFind and start the configuration in SensoConfig.

Select an interface in SensoConfig

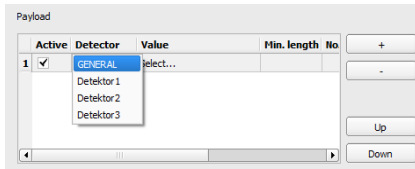


- Use the "Output" setup step to open the "Interfaces" tab.
- Enable the interface by enabling the corresponding checkbox in the "Enabled" column.
- In the "Setting 3" column, select the format for the data output.

Defining telegrams / data output in SensoConfig



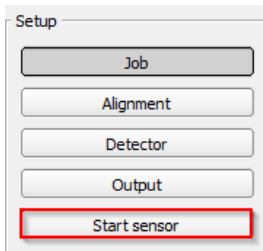
- Use the "Output" setup step to open the "Telegram" tab.
- Set the control characters you want for the data output.
- Select the Checkboxes you want.



- Configure the data you want to be output. Use the "+" button to generate new entry. What the buttons do:
 - "+": Insert new entry
 - "-": Delete marked entry
 - "Up", "Down": Displace marked entry
- Select the detector you want in the "Detector" column.
- Select the detector value you want in the "Value" column so that this value will be output through the enabled interface.

Additional information:: Data output ([ASCII](#) / [binary](#))

Start sensor



- Click on the "Start sensor" setup step. The data will be transferred to the vision sensor and the vision sensor will be started.



NOTE:

Detector must be generated.

4 Ethernet TCP/IP, port 2005 / 2006

Numerical data, which has been configured under Output/Telegram, can be output in a separate ASCII/BINARY format.

The sensor here is the (socket) "server", and provides the data via a "server socket" interface. This is mainly a "programming interface".

To read / process the data, a "socket client" (PC, PLC, etc.) must establish a (socket) connection (active) to the sensor, and then receives the data.

Handling, Settings

4.1 Example: Data output from VISOR® to PC / PLC

Step 1:

After the job with all necessary detectors, Alignment, etc. is set, the Ethernet interface for data output is activated and, if necessary, parameterized.

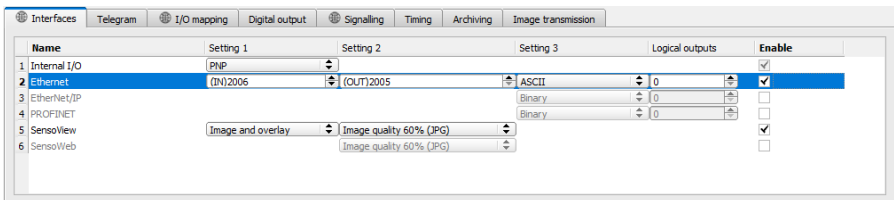


Fig. 6: Data output, Ethernet

In the example, the Ethernet interface is activated in the parameter field in the tab "Interfaces" by marking the checkbox "Enable". The default settings for input port (IN) = 2006 and output port (OUT) = 2005 are adopted in this way. Any other settings can be made here to adapt the data output to your network environment. If necessary, contact your network administrator.

Step 2:

The "Telegram" tab configures the payload to be output via Ethernet Port 2005.

In this example, it is the:

- Start "010"
- Overall result of Detector 1
- Trailer "xxx"

"ASCII" is defined as a data format, which facilitates the traceability of this example. The function with other data or in binary is analogous to settings made here by way of example.

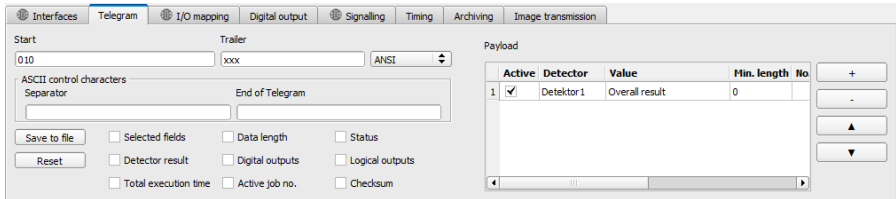


Fig. 7: Telegram, configure output data

Step 3:

After opening the Hercules Ethernet tool, you will need to open the "TCP-Client" tab to communicate with the VISOR® socket server via Ethernet.

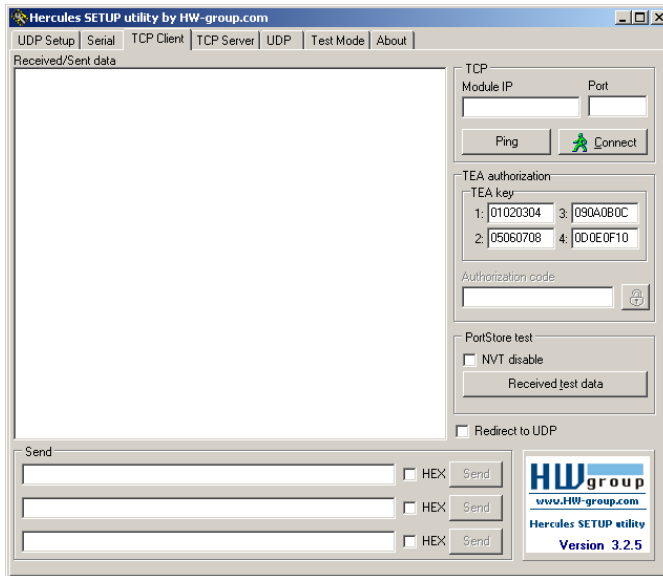


Fig. 8: Data output, Ethernet Tool / 1

You will need to enter the IP address of the VISOR® and the correct port in order to receive data. The IP address of the VISOR® can be found in SensoFind. See the first line in the window "Active sensors" = 192.168.60.199

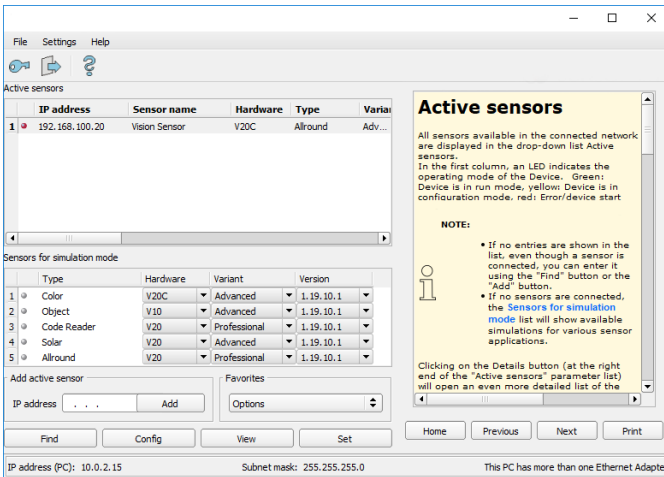


Fig. 9: SensoFind, IP address ...

The port number for the output port was adopted under Step 1 with Port 2005.

Step 4:

Therefore, the following settings are made in Hercules: Module IP = 192.168.60.199, Port = 2005. All other settings remain in the default values. Clicking on the "Connect" button will establish a connection to the VISOR® and the connection will be shown in green letters in the main window.

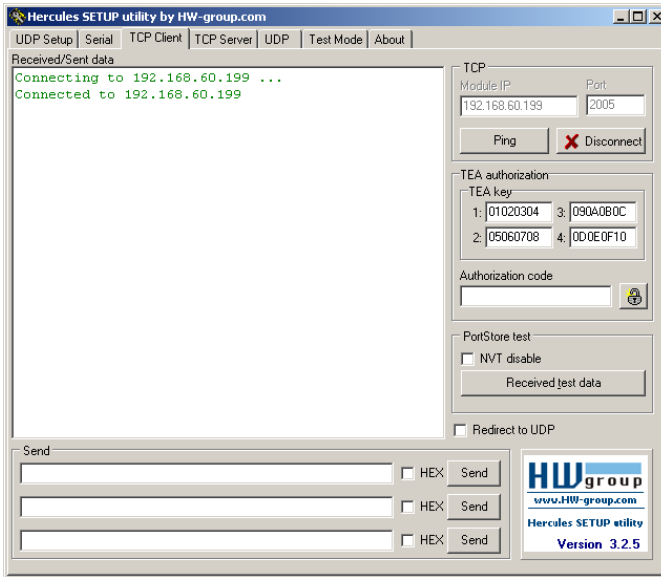


Fig. 10: Data output, Ethernet Tool / 2

Step 5:

You will now need to start the VISOR® from the PC application with "Start sensor" (later during operation, the VISOR® will run normally after being turned on and will transmit data if configured). In this example, Trigger mode = continuous is set, i.e. evaluations are made continuously and data is sent. These are only visible in the main window of Hercules.

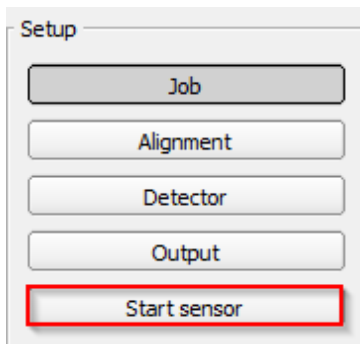


Fig. 11: Start sensor

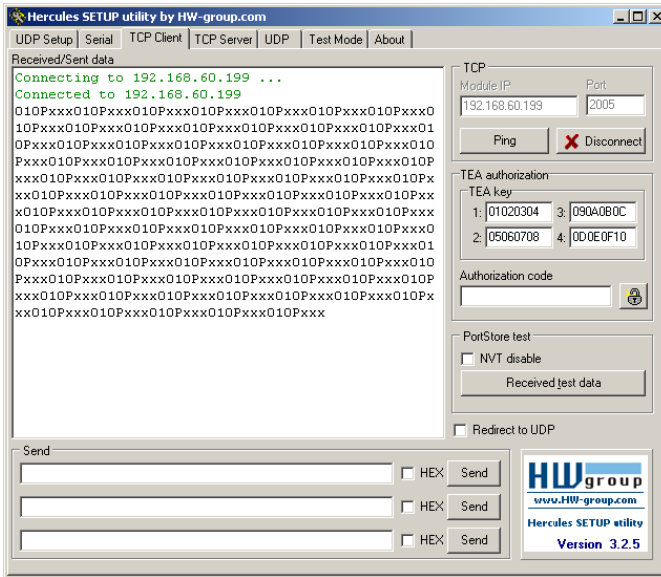


Fig. 12: Data output, Ethernet, Tool/3

The data visible here are set under "Telegram":

- Start "010"
- Overall result of Detector 1 (here, a "P" for positive, since test condition: brightness fulfilled)
- Trailer "xxx"

4.2 Example: Commands (requests) from PC / PLC VISOR®

With acknowledgement / data output from VISOR®

Step 1

For better clarity, the triggered operation is switched to here for Example 2. This can be done as follows: In SensoConfig under Job/Image Acquisition/Trigger mode = Set "Trigger". All other settings from Ethernet example 1 in the VISOR® remain unchanged.

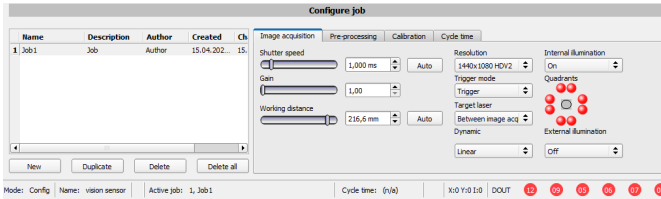


Fig. 13: Data output, Ethernet, Trigger

Step 2

In order to transmit commands to the VISOR®, the Hercules application needs to be opened again. This time with port 2006 as the VISOR® input port through which it can receive commands. All telegrams (commands and response strings) to and from the VISOR® are described in section [Overview telegrams \(Page 87\)](#).

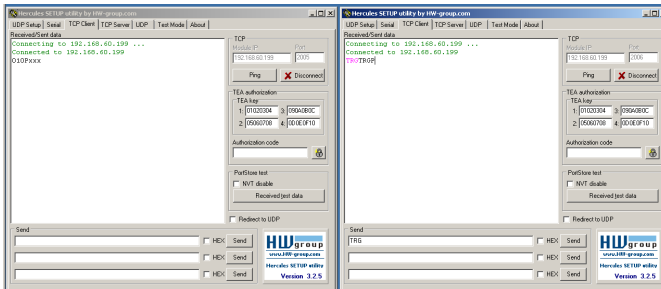


Fig. 14: Data output, Ethernet Tool / 4

In the right window, the "TRG" command (for Trigger; see first line on the bottom for command) was sent from port 2006 to the VISOR® by clicking on the corresponding "Send" button. This command is shown in the main window in red letters when being sent. The VISOR® responds to port 2006 with an acknowledge to the "TRG" command and, in this case, "P" for a positive detector 1 result (black letters in right pane).

In the left window, the VISOR® uses output port 2005 to send the "010Pxxx" value defined in Data output the same way as in the Ethernet 1 example.

4.3 Example: Job change from PC / PLC to VISOR®

With acknowledgement / data output from VISOR®

Function of both Ethernet ports for in- and output:

*A: Port 2005, only one direction: Sensor >> PC, all payload, defined under "Data output"

*B: Port 2006, both directions: Sensor <> PC, commands to VISOR® with acknowledge, + all response data to commands (no payloads)

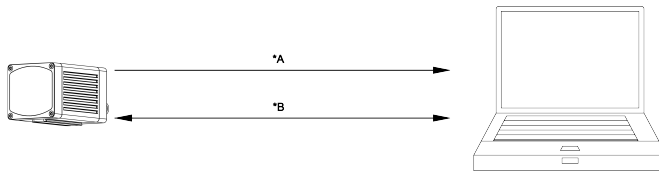


Fig. 15: Ethernet ports

Step 1

For better clarity, it is switched here to the triggered mode. This can be done as follows: In SensoConfig under Job/Image Acquisition/Trigger mode = Set "Trigger". All other settings from Ethernet example 1 in the VISOR® remain unchanged. All data output definitions are made here in "ASCII" for better traceability of the examples.

For this example, at least two jobs must be created on the VISOR® vision sensor. To create a new job based on an existing job, you can use the "Duplicate" function. Adjust the following parameters to easily check the job change. Later you can freely define the output.

For this example, Job 1 was defined with the data output:

- Start: "010" and
- Trailer: "xxx"

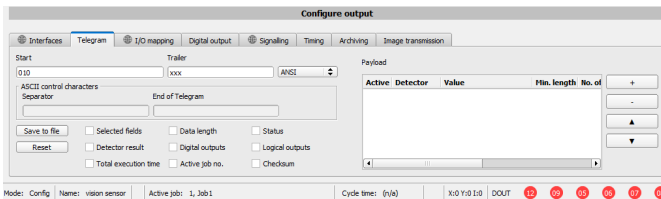


Fig. 16: Data output, Ethernet, Job switch Job 1

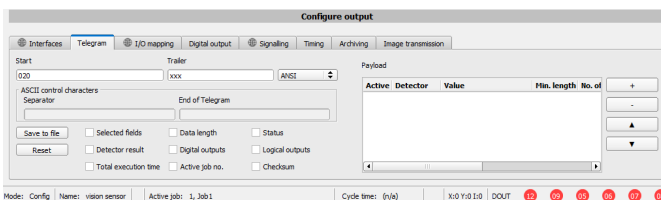


Fig. 17: Data output, Ethernet, Job switch, Job 2

Step 2

Here, the application Hercules was opened twice. Once with port 2005 (receiving of results as defined in "Data output") and port 2006 (commands + acknowledge) as VISOR® input port through which it can receive commands.

All telegrams (commands and response strings) to and from the VISOR® are described in section [Overview telegrams \(Page 87\)](#).

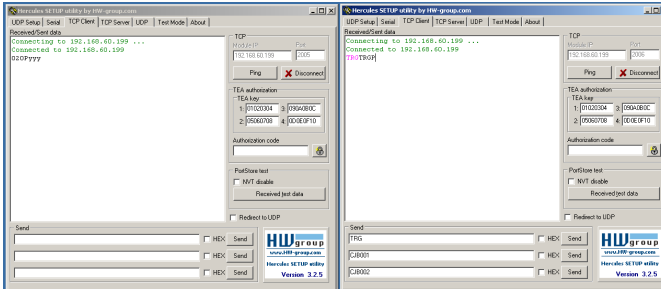


Fig. 18: Data output, Ethernet, Job switch, Tool / 1

In the right window (Port 2006), the command TRG (Trigger, see "Send" below, first line) was issued. This is displayed in the main window in red letters with "TRG". The VISOR® responds immediately with the "TRGP" acknowledge (repetition of "TRG" command and "P" for positive, in black letters in the right pane)

In the left window (Port2005), the VISOR® on which Job 2 is currently active sends the corresponding result string, which is defined in Data output in Job 2 with "020Pyyy".

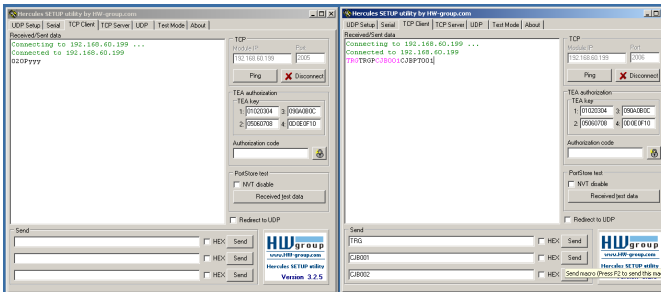


Fig. 19: Data output, Ethernet, Job switch, Tool / 2

Nun wurde im rechten Fenster (Port2006) das Kommando CJB001 (Change Job 001, 001 = Job Nr. 1, siehe unten bei „Send“, zweite Zeile) abgesetzt. This is displayed in the main window in red letters with "CJB001". The VISOR® responds immediately with the "CJBPT001" acknowledge (repetition of "CJB" command, "P" for positive, "T" = Triggered, 001 job number to which the change was made)

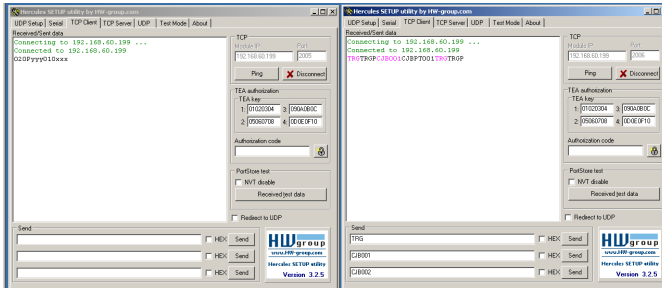


Fig. 20: Data output, Ethernet, Job switch, Tool / 3

After the next Trigger command TRG (see "Send" below, third line) is displayed again in the main window in red letters. The VISOR® immediately responds again with the "TRGP" acknowledge (repetition of "TRG" command and "P" for positive)

In the left window (Port2005), after the job has changed to Job 1, the VISOR® sends the corresponding result string, which is defined in Data output in Job 1 with "010xxx"!

4.4 Example Beckhoff CX 1020

The connection to a Beckhoff CX 1020 and the corresponding configuration is described in the Beckhoff Operating instructions in:

Start menu/SensoPart/VISOR vision sensor/Tools/SPS PLC/...

4.5 Example Siemens S7

The connection to a Siemens S7 PLC and the corresponding configuration is described in the Siemens S7 Operating instructions in:

Start menu/SensoPart/VISOR vision sensor/Tools/SPS PLC/...

5 Service / Visualization

There is a service port (Ethernet TCP/IP port 1998) available for the VISOR® vision sensor. This port will be available regardless of how you configure the various steps.

5.1 Backup creation

The following telegrams can be used for automatic backups and restores

- **Read job set (ASCII)**

The "Set job set" telegram can be used to change the VISOR® vision sensor's job set. The job set file must first be loaded onto the VISOR®.

- **Save job set (ASCII)**

The "Save job set" telegram can be used to read the VISOR® vision sensor's job set.

5.2 Visualization

The VISOR® vision sensor provides all data for the visualization of the applications via the service port.

Additional information: [Service \(available only on port 1998 and in ASCII format\) \(Page 161\)](#)

6 VISOR® telegrams for PROFINET and EtherNet/IP

6.1 Module 1: "Control" (From PLC to VISOR®)

Name in PLC "CTRL (3 bytes)"


Byte Offset	Bit Adr..	Name	Data type	Meaning
0	0	Reset error	1 bit	Reset Error clears the 4 bit error code in the "Status" module. Rising edge (False → True) clears error code.
	1	HW trigger Disable	1 bit	This bit is used to disable the trigger. Valid for Trigger mode Trigger and Free run. <ul style="list-style-type: none"> False (0): Trigger activated. True (1): Trigger disabled. If the digital input "Trigger enable" is used, both conditions (digital input "Hardware Trigger" and "Trigger Disable Bit") must be set to "Enable" to accept a trigger.
	2	Trigger	1 bit	Rising edge (false → true): Trigger is executed immediately. If the trigger could not be executed, the Trigger acknowledge Bit stays false and "Error status" module has the error code "1: Failure trigger request". See also Timing diagram, Chapter Case: Trigger not possible (not ready)
	3	Change job	1 bit	Rising edge (false → true): Switch to job with number "Job number" from Control module. When executing this request, delays may occur. After a successful job change, the "Job number" byte in the "Status" module shows the same value as in the Control module. If the job change could not be executed due to error (due to an error, e.g. wrong job number), the "Error status" module has the error code "2: Failure change job" (and Ready stays false!). See also Timing diagram, Chapter Case: Job change not possible (e.g. wrong job number)

Byte Offset	Bit Adr..	Name	Data type	Meaning
	4	Switch-to-Run	1 bit	Rising edge (false → true) "Switch to Run" is executed. Success or failure of Switch to Run request is shown in the "Error status" module (error code "3: Failure Switch to run request") and bit "Operation Mode". See also Timing diagram, Chapter Case: Switch to run not possible
	5-7	Reserve		
1		Reserve	1 byte	
2		Job number	U8	Job number to be switched to, on the rising edge of the change job bit. Binary value 1-255 for "Job number change". 0 stands for "No switching", even if the Change Job Bit changes.

[Timing diagrams for VISOR® communication \(Page 41\)](#)

6.2 Module 2: "Status" (from VISOR® to PLC)

Name in PLC "STAT (6 bytes)"

Byte Offset	Bit Addr.	Name	Data type	Description
0	0	Ready	1 bit	VISOR® ready for next evaluation. Ready=1.  Attention: The Ready bit is exclusively reserved for indicating the readiness of the VISOR® vision sensor for the next evaluation. It is not suitable for indicating that an evaluation has been completed or the results of an evaluation are available!
	1	Reserve	1 bit	
	2	Trigger acknowledge	1 bit	Acknowledge (confirmation) for successful trigger request (via Trigger Bit in Control module). Acknowledge is deleted as a response to the deletion of the trigger bit. If the trigger could not be executed, the Trigger Acknowledge Bit stays false.
	3	Change Job acknowledge	1 bit	Acknowledge (confirmation) for the Change Job Request (via Change Job Bit in Control module) – independent of its success. Acknowledge is deleted as soon as the Change Job Request Bit has been deleted. Success or failure of Change Job Request is shown in the bitfield "Error" (error code "2: Failure change job") and in the byte "Job number" in the Status module. If there are delays in executing the job change, this acknowledge bit can also be set with a delay.

Byte Offset	Bit Addr.	Name	Data type	Description
	4	Switch to run acknowledge	1 bit	Acknowledge (confirmation) for the Switch to Run Request (via Switch to Run Request Bit in the Control module). Acknowledge is deleted as soon as the Request Bit is deleted. Success or failure of Switch to Run Request is shown in the bitfield "Error" (error code "3: Failure Switch to run request") and bit "Operation Mode". Acknowledge is set after SensoConfig is closed and the job has been loaded from the flash or if an error has occurred.
	5-7	Reserve		
1		Reserve	1 byte	
2	0	Digital Results	1 bit	12 RDBU
	1		1 bit	09 RD
	2		1 bit	05 PK
	3		1 bit	06 YE
	4		1 bit	07 BK
	5		1 bit	08 GY
	6	Reserve	1 bit	This byte is filled with the results of the digital switching outputs. The bit position is fixed. The value of the output is defined in the tab: Output/Digital output, Column: "Logical expression" in SensoConfig. If not selected as result output pin, or if no valid logical expression is assigned, the value is = 0.
	7	Reserve	1 bit	
3		Job number	U8	Number of current job: Job number 1-255
4		Image ID	U8	Image ID (0 - 255) is incremented by 1 with each job execution, independent of the trigger source.

Byte Offset	Bit Addr.	Name	Data type	Description
5	0-3	Error	4 bit	<p>4 bit error code (decimal). Used to indicate errors in requests via the control module or VISOR® system errors. The error code can be reset by "Reset error" or is overwritten by the next error. In case of an archiving error (8), you can continue without a "Reset error".</p> <p>0: No error 1: Error: Trigger request error (sensor not Ready) 2: Error: Change job 3: Error: Switch-to-Run 5: Error: Interface not active in job 7: Focus lock time 8: Error: Archiving 15: System error</p>
	4	Trigger Mode	1 bit	<p>1 = Free run 0 = Trigger</p>
	5	Reserve	1 bit	
	6	Operation mode	1 bit	<p>1 = Run 0 = Config</p>
	7	Reserve	1 bit	

6.3 Module 3: "Data" (from VISOR® to PLC)

Name in PLC "DATA (2 + 8 / 16 / ... / 192 / 252 Bytes)"

Byte Offset	Bit Addr.	Name	Data type	Description
0		Image ID	U8	Image ID (0 - 255) is incremented by 1 with each job execution, independent of the trigger source.
1	0	Result data overrun	1 bit	Result data has been truncated. 1: Data overrun = truncated 0: No overrun
	1 - 7	Reserve	7 Bit	
2		Result data	Byte array	Data as defined in SensoConfig in "Output/Data Output/Detector-Specific payload". When using PROFINET "binary" must be enabled in the Interfaces tab.

6.4 Module 4: "Request" (From PLC to VISOR®)

Name in PLC "REQU (4 + 8 / 16 / ... / 192 / 250 Bytes)"

Byte Offset	Bit Addr.	Name	Data type	Meaning
0	1	Key	1 byte	Request key (Request counter)
1	1	Reserve	1 byte	Reserve
2	1	Reserve	1 byte	Reserve
3	1	Reserve	1 byte	Reserve
4		Request Data	Byte array	Additional information: Overview telegrams (Page 87)

6.5 Module 5: "Response" (from PLC to VISOR®)

Name in PLC "RESP (4 + 8 / 16 / ... / 192 / 250 Bytes)"

Byte Offset	Bit Addr.	Name	Data type	Description
0		Key	U8	Response key = mirrored from request
1	0	Result Data overrun	1 bit	Response data has been truncated
	1-7	Reserve	7 Bit	
2		Reserve	1 byte	
3		Reserve	1 byte	
4		Result Data	Byte array	Additional information: Overview telegrams (Page 87)

6.6 Start / end criteria for each telegram

Telegram ("Control" module)	Start condition ("Status" module)	Acceptance confirmation ("Status" module)	Execution confirmation ("Status" module)
Trigger	Ready = True	Trigger acknowledge = True	Image ID changed
Change job	/	Change Job acknowledge = True	Job number changes
Switch-to-Run	Operation Mode = False	Switch-to-Run acknowledge = True	Operation Mode = True

7 Timing diagrams for VISOR® communication

Case: Trigger ok

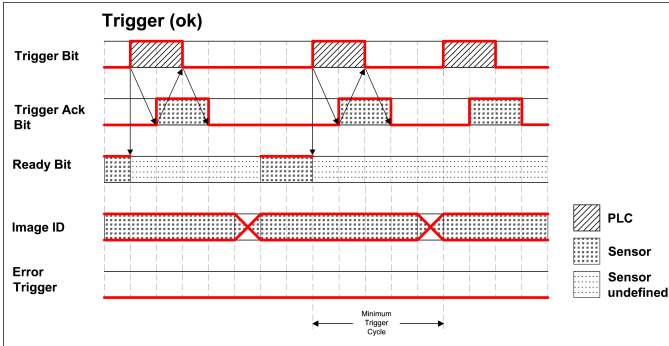


Fig. 21: Timing Trigger ok

Case: Trigger not possible (not ready)

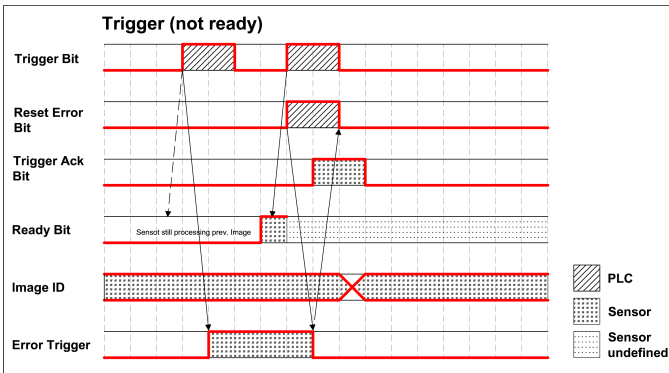


Fig. 22: Timing Trigger not ready

Case: Job change ok

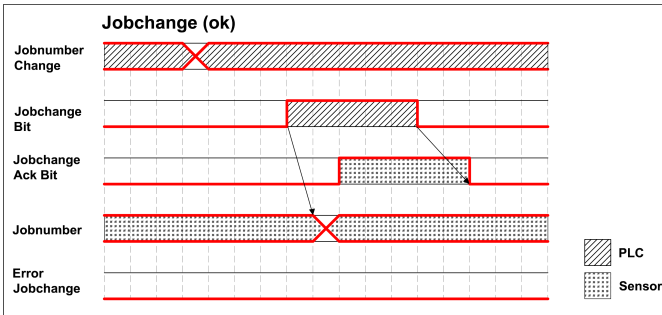


Fig. 23: Timing Job change ok

Case: Job change delayed

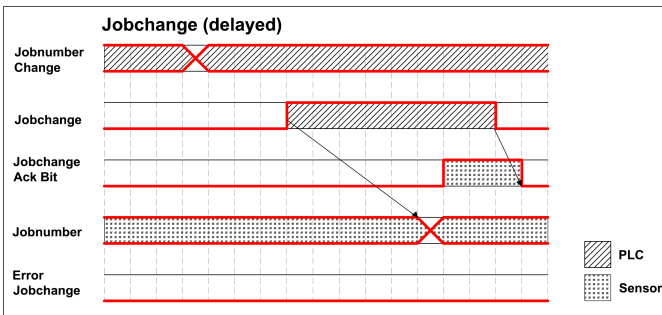


Fig. 24: Timing Job change delayed

Case: Job change not possible (e.g. wrong job number)

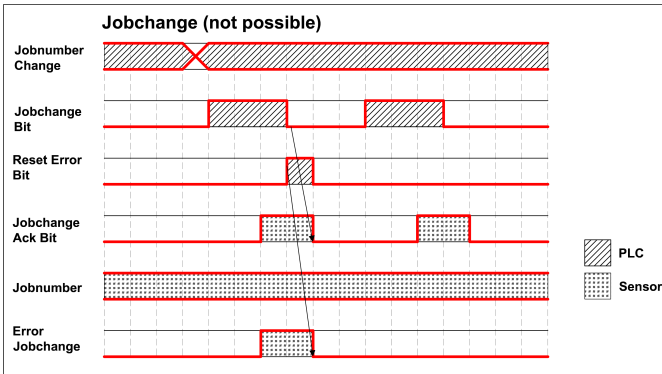


Fig. 25: Timing Job change not possible

Case: Switch to run ok

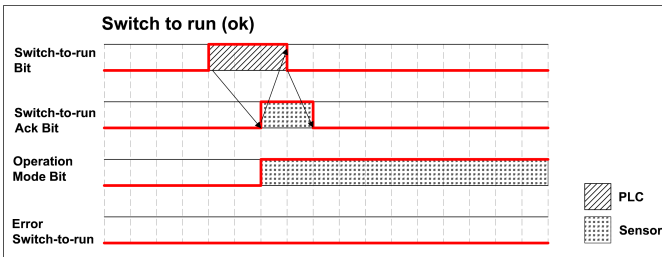


Fig. 26: Timing Switch to run ok

Case: Switch to run not possible

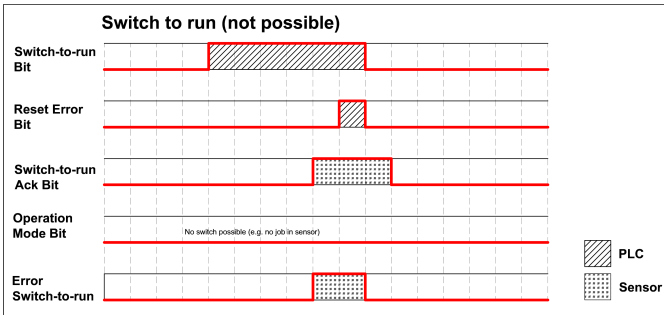


Fig. 27: Switch to run not possible

8 Request sequences

Important recommendations for PLC programmers

1. Follow the sequence of requests
2. Wait for complete execution of an action before sending the next one. Complete execution takes place when the image ID changes in the trigger request, or the corresponding acknowledge bit is set for the other requests.



NOTE:

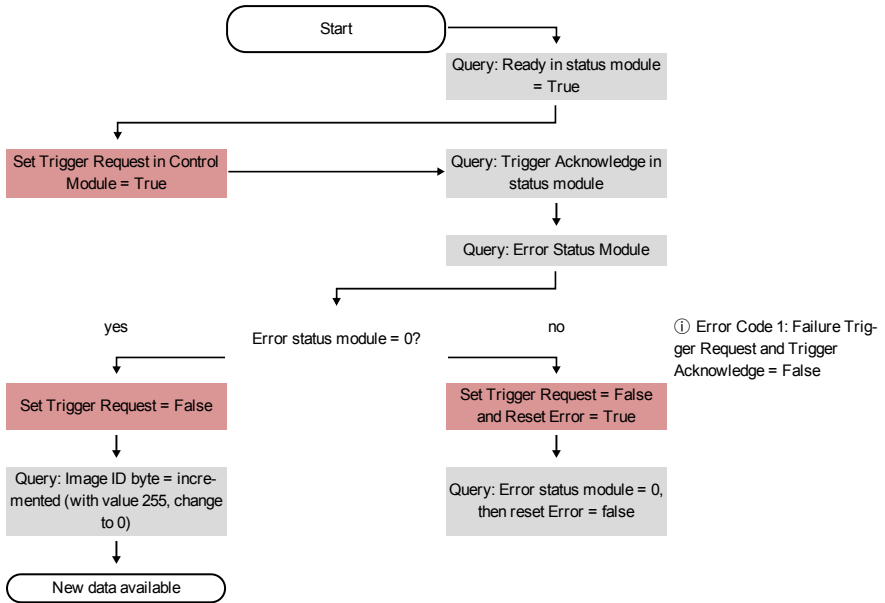
The complete execution of an action cannot be recognized as safe due to the low/high change of READY, since due to possibly long cycle times between PLC and VISOR® (e.g. 32ms), READY may never become low.

3. READY should always be high before a trigger request is sent

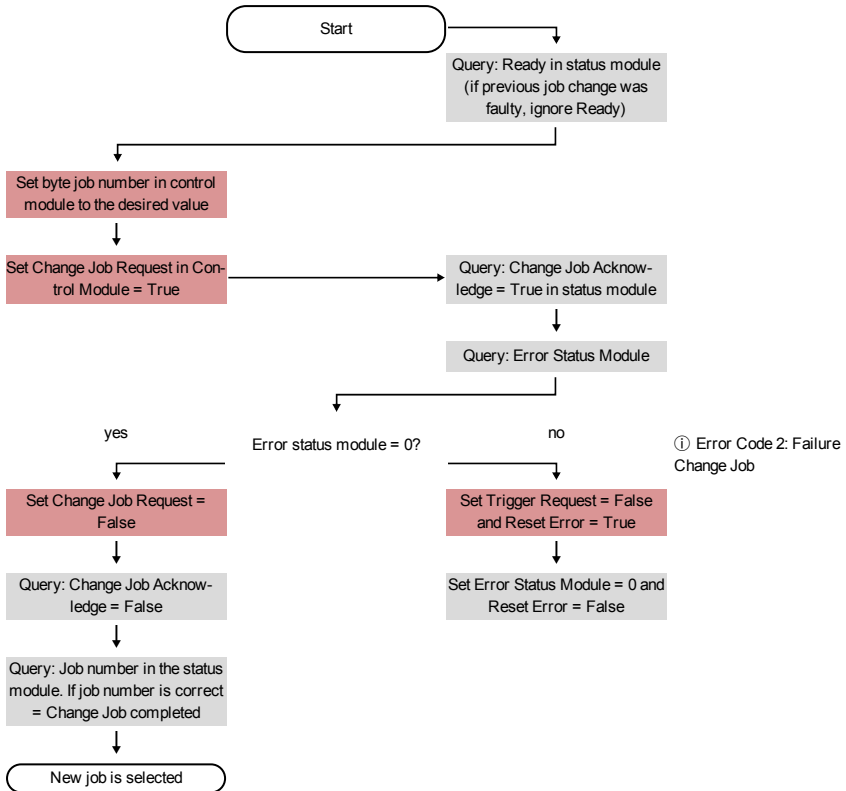
Accepting / discarding of requests of the control module

1. Request is accepted with an increasing acknowledge bit
2. Request is discarded if the error bit is set.
3. Request is discarded without an error bit and acknowledge bit if the sensor is still processing the previous request and no acknowledgment has yet been set for it. (i.e. not following the recommended handshake)

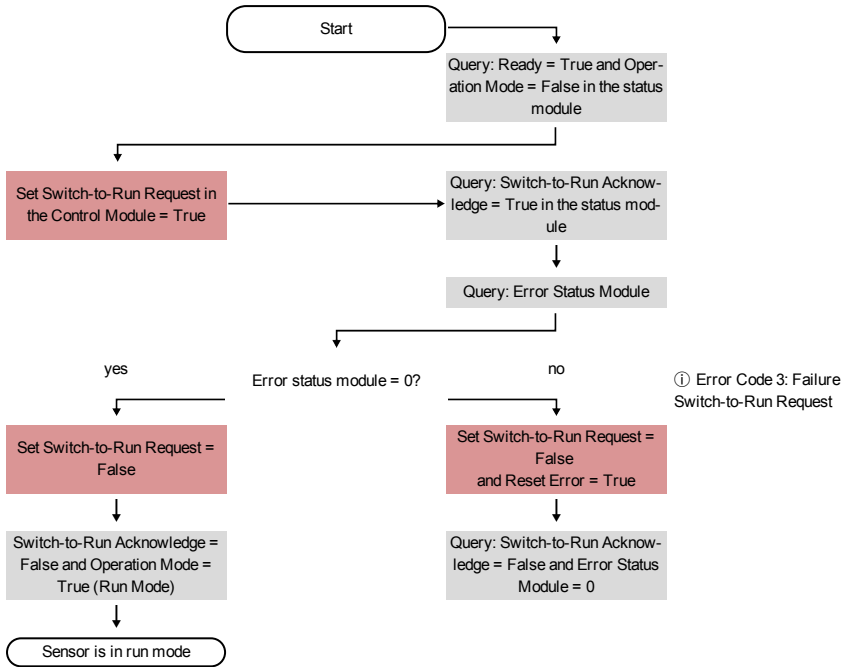
8.1 Trigger Request Sequence



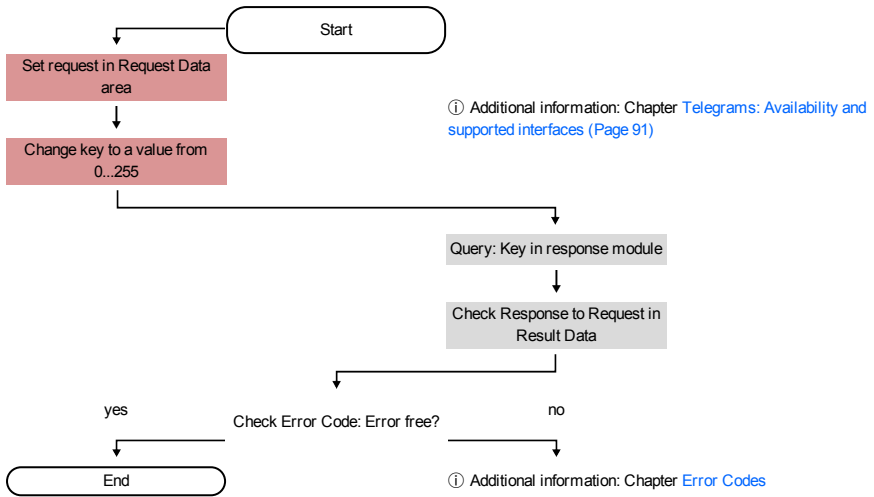
8.2 Change job request sequence



8.3 Switch to Run sequence



8.4 Sequence for requests via request/response module



Additional information:

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Error codes \(Page 94\)](#)

Error Reset (depicted in the use case "Job change not possible")

1. Reset with "Reset Error Bit"
2. Error bits are overwritten by new error bits.

9 PROFINET

This section explains how to operate the VISOR® vision sensor with PROFINET.

9.1 Siemens S7-1200 TIA 12 configuration example

This description shows all PLC screenshots in English; switch the TIA software to English if necessary.

9.1.1 Create new project

New project with: Project / Create new project

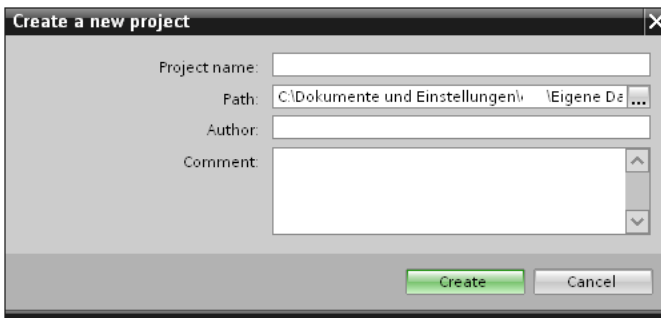


Fig. 28: PROFINET Create new project

9.1.2 Selecting the GSD file

First a PROFINET PLC must be added to the project.

In order to be able to use the PROFINET functions of the VISOR® vision sensor, the latest version of the corresponding VISOR® GSD file must be installed. This is done at: Options/Install general station description file. The EDS file can be found in the installation path for the VISOR® in: ... \SensoPart\VISOR vision sensor\Tools\PROFINET and is also available for download at www.sensopart.com.

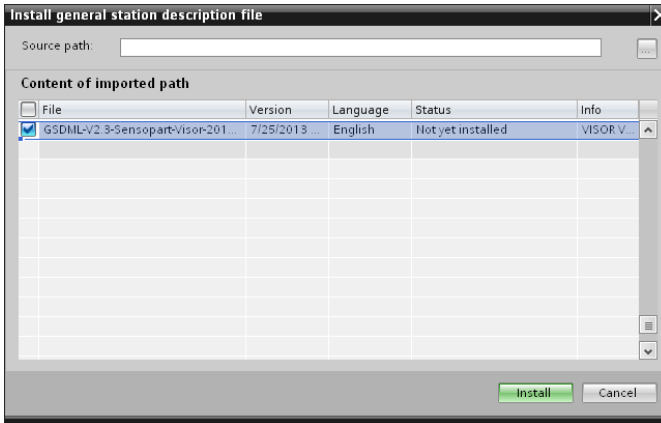


Fig. 29: Selecting and installing the GSD file

9.1.3 Adding the VISOR® vision sensor to the project

The VISOR® modules are added in the hardware catalog: Other field devices/PROFINET IO/sensors/SensoPart Industriesensorik GmbH.

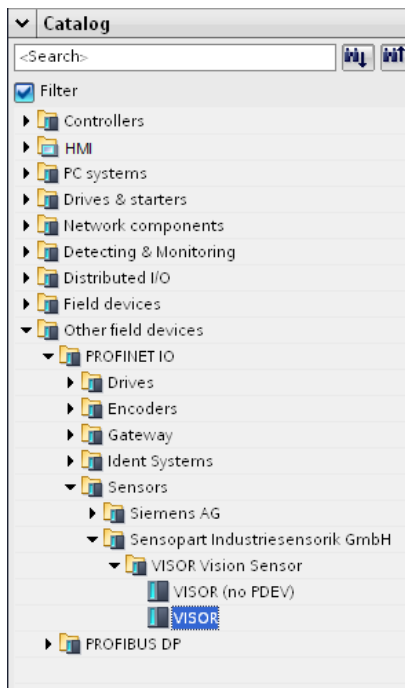


Fig. 30: Adding the VISOR® to the project

Connecting the VISOR® to the PLC

You can now drag a VISOR® module from the catalog and drop it in the Network View. The VISOR® is connected to the PLC via PROFINET (Network View tab).

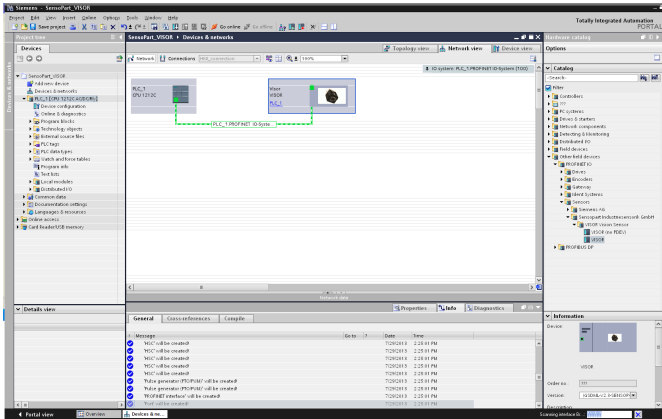


Fig. 31: Connecting the VISOR® to the PLC

Definition of I/O data

In the tab "Device view", the modules CTRL (Control) and STAT (Status) are active by default. As an option, the module DATA (Data module) can be added with a certain use size.

In this example: 2 bytes + 16 bytes of payload (1 byte: Image ID, 1 byte: Result data overrun (see [Module 3: "Data" \(from VISOR® to PLC\) \(Page 36\)](#)), + 16 bytes of data). If the data are longer than the defined range, these are truncated (in this case: Result data overrun = 1); if it's shorter, the rest of the 16 bytes are filled with 00h.

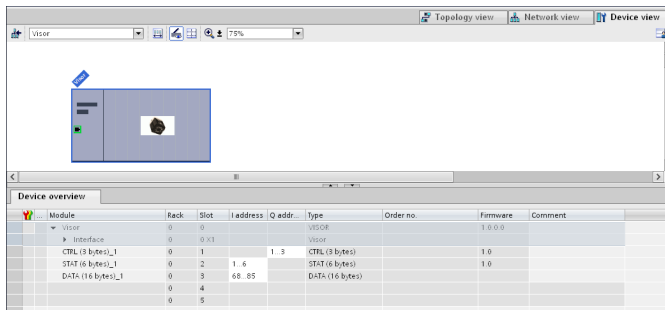


Fig. 32: Enter I/O data

Configuring the VISOR[®] IP address

Option 1: In the project

The IP address for the VISOR[®] can be assigned through the project in the PLC. Select option "Set IP address in the project" and enter IP address. The address from the "IP address" field will be written to the VISOR[®]. The IP addresses of the PLC and the VISOR[®] must be different from each other but correspond to each other, i.e., fall within the same address space.

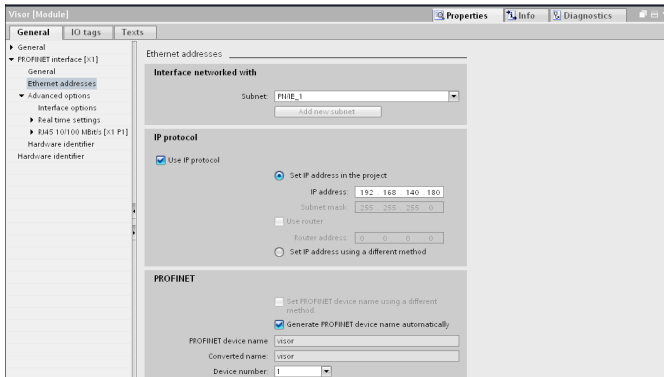


Fig. 33: Configuring the VISOR[®] IP address in the project

The VISOR[®] can also be used without a running PROFINET, and accordingly can be configured via SensoFind. If the IP address of the VISOR[®] does not match the one in the TIA project, the PLC will configure the IP address instead. In this case, the original configuration in the VISOR[®] will be overwritten with 0.0.0.0. This means that the IP address is set correctly but the IP configuration is deleted (this is important for a restart, possibly without a connected PLC).

Option 2: In SensoFind

The IP address of the VISOR[®] can also be configured via SensoFind. Select option "Set IP address using a different method" in the PLC / TIA interface. Configure the IP address via SensoFind (See Chapter: [Settings in SensoFind \(Page 15\)](#)).

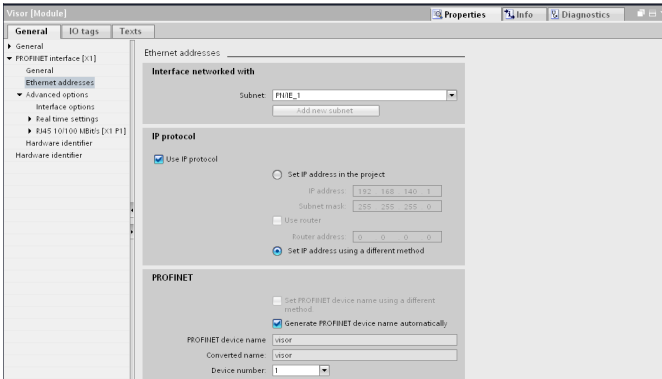


Fig. 34: Configure the IP address for the VISOR® in SensoFind®; the corresponding settings can be found in the PLC/TIA interface

Set the name with TIA interface

There are two ways to configure the name for the VISOR® from the TIA Portal.

Generate name automatically

The PROFINET name for the VISOR® can be generated automatically in the PLC. Option: "Generate PROFINET device name automatically" takes the name from the project.

Set name manually

If the option "Set PROFINET device name using a different method" is selected any name can be set.

Information: In the field "Converted name", a different name than entered is displayed, which is then also used. Da im PROFINET nicht alle Zeichen genutzt werden können ist eventuell eine Konvertierung notwendig und wird hier automatisch gemacht (Namen müssen DNS kompatibel sein, s. dazu auch Kap. [Settings in SensoFind \(Page 15\)](#)).

If the VISOR®'s name is configured using the TIA Portal, it must be written to the sensor with the "PROFINET device name" tool (as described in section [Writing a name to VISOR® \(Page 57\)](#)).

The PROFINET name in the project and in the VISOR® must match.

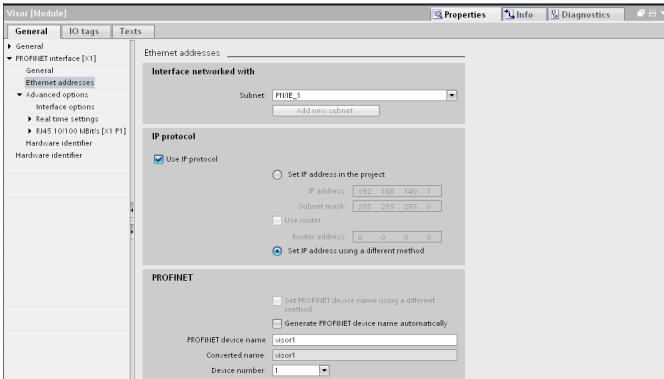


Fig. 35: Set name in project

9.1.4 Writing a name to VISOR®

In order to be able to establish communications, the PROFINET name must be written to the VISOR® in case it needs to be updated.

This is done with the tool: Online/Assign PROFINET device name. Select the corresponding device (VISOR®) and apply the name with "Assign name."

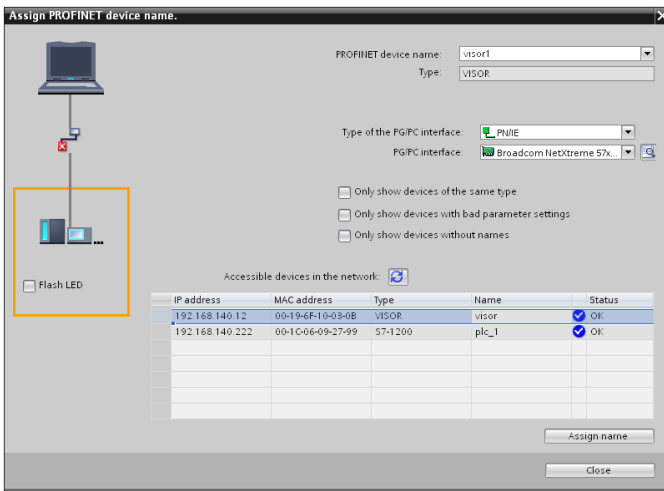


Fig. 36: Writing a name to VISOR®

9.1.5 Load the project onto the PLC

To finish the configuration and save changes of the project: 1. translate and 2. transfer / write to the PLC



Fig. 37: Translate project and write to PLC

9.1.6 Mapping of output data

The VISOR® vision sensor's output data can be mapped to the data in the PROFINET log as follows:

Step 1) The start address for an input variable can be taken from "Device Overview".

	Name	Address	Display format	Monitor value	Modify value
1		%I868	Hex	16#00	
2		%I869	Hex	16#00	
3	*Data 1*	%I870	Hex		
4	*Data 2*	%I871	Hex		
5	*Data 3*	%I872	Hex		
6	*Data 4*	%I873	Hex		
7	*Data 5*	%I874	Hex		
8	*Data 6*	%I875	Hex		
9	*Data 7*	%I876	Hex		
10	*Data 8*	%I877	Hex		
11	*Data 9*	%I878	Hex		
12	*Data 10*	%I879	Hex		
13	*Data 11*	%I880	Hex		
14	*Data 12*	%I881	Hex		
15	*Data 13*	%I882	Hex		
16	*Data 14*	%I883	Hex		
17	*Data 15*	%I884	Hex		
18	*Data 16*	%I885	Hex		
19		<Add new>			

Fig. 38: Table of variables

Step 2) Creating a tag table in the PLC

Module	Rack	Slot	I address	Q address	Type	Order no.	Firmware	Comment
VISOR	0	0			VISOR			
Interface	0	0 X1			VISOR			
CPU (3 bytes)_1	0	1		1..3	CPU (3 bytes)			
STAT (6 bytes)_1	0	2	1..6		STAT (6 bytes)			
DATA (2 + 16 bytes)_1	0	3	68..85		DATA (2 + 16 bytes)			
REQU (4 + 16 bytes)_1	0	4		64..83	REQU (4 + 16 bytes)			
RESP (4 + 16 bytes)_1	0	5	86..105		RESP (4 + 16 bytes)			

Fig. 39: Device overview

Step 3) Creating the configuration in SensoFind and saving the configured log as a CSV file.

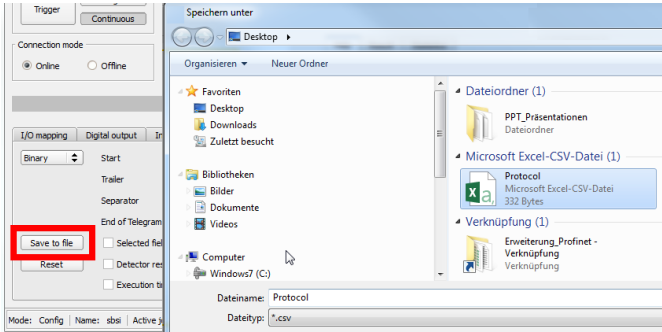


Fig. 40: Output format saved as CSV file

Step 4) Opening the file with the text program

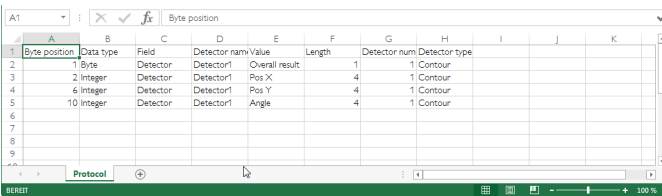


Fig. 41: Output protocol in Excel representation

For a description of the format of the PROFINET Data module, please refer to [Module 3: "Data" \(from VISOR® to PLC\) \(Page 36\)](#)

Step 5) The result is the following assignment between the input data of the PLC

Name	Address	Display format	Monitor value	Modify value	Comment
	%B68	Hex	16#01		
	%B69	Hex	16#00		
"Data1"	%B70	Hex	16#01		
"Data2"	%B71	Hex	16#00		
"Data3"	%B72	Hex	16#03		
"Data4"	%B73	Hex	16#98		
"Data5"	%B74	Hex	16#C6		
"Data6"	%B75	Hex	16#00		
"Data7"	%B76	Hex	16#05		
"Data8"	%B77	Hex	16#88		
"Data9"	%B78	Hex	16#85		
"Data10"	%B79	Hex	16#FF		
"Data11"	%B80	Hex	16#FF		
"Data12"	%B81	Hex	16#FF		
"Data13"	%B82	Hex	16#78		
"Data14"	%B83	Hex	16#00		
"Data15"	%B84	Hex	16#00		
"Data16"	%B85	Hex	16#00		
	<Add new>				

Fig. 42: Input data PLC

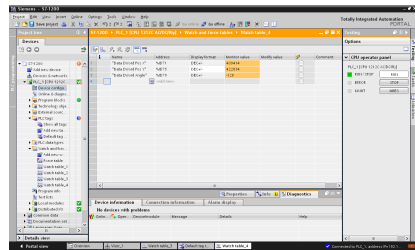
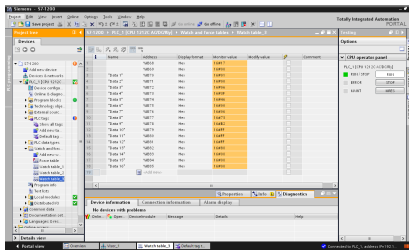
... and the configured protocol:

Byte position	Data type	Field	Detector name	Value	Length	Detector num	Detector type
1	Byte	Detector	Detector1	Overall result	1	1	Contour
2	Integer	Detector	Detector1	Pos X	4	1	Contour
4	Integer	Detector	Detector1	Pos Y	4	1	Contour
5	Integer	Detector	Detector1	Angle	4	1	Contour

Fig. 43: In the vision sensor configured protocol

Conversion of binary values

All detector-specific payloads with decimal places will be transmitted as integers multiplied by 1000, and accordingly must be divided by 1000 after the data is received. The values are transferred in the format "Big-endian". The length is based on the value, e.g., score 32 bits (DWord).



Detector	Score	Time	Detector
1	99.8	27ms	Contour

Score	Position X [µm]	Position Y [µm]	Angle [°]	Scale	Circle pos.x [µm]	Circle pos.y [µm]	Circle angle	Position centered	
1	99.8	403.4	402.8	-0.1°	1	0.0	0.2	0.1°	Off

Count	Pass
1	100.00%
Fail	0.00%
Minimum	42ms
Maximum	42ms
Average evaluation time	42ms

9.2 PLC example programs

The following PLC example programs show some basic functions.

PLC example 1: Trigger when VISOR® Ready

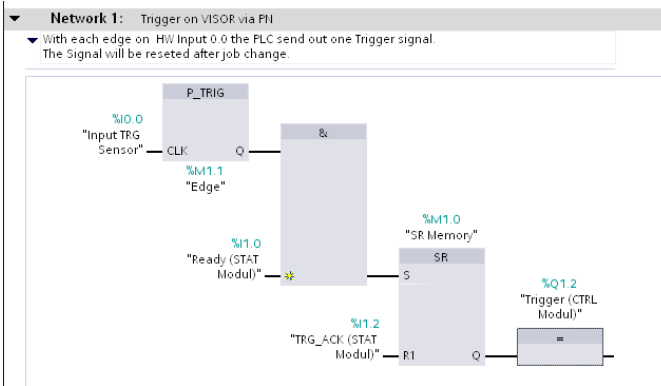


Fig. 44: Trigger when VISOR® Ready, (without error handling)

PLC example 2: Send job number to VISOR®

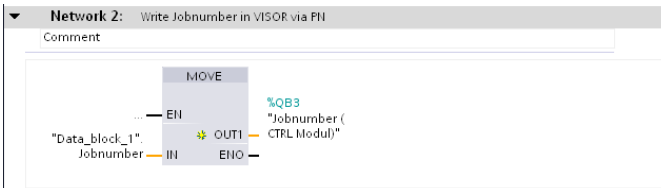


Fig. 45: Send job number

PLC example 2.1: Job change when VISOR® Ready

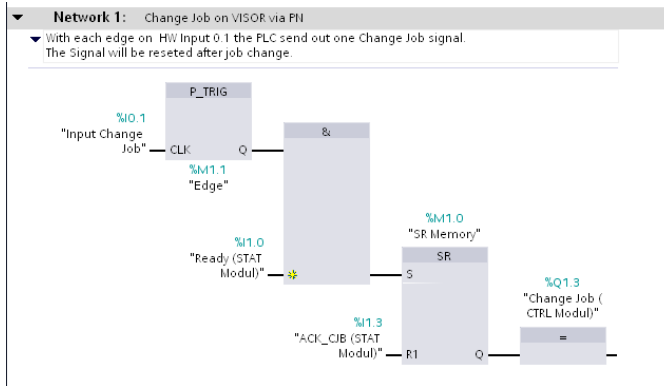


Fig. 46: Job change when VISOR® Ready, (without error handling)

PLC example 3: Switch to Run when VISOR® in configuration mode

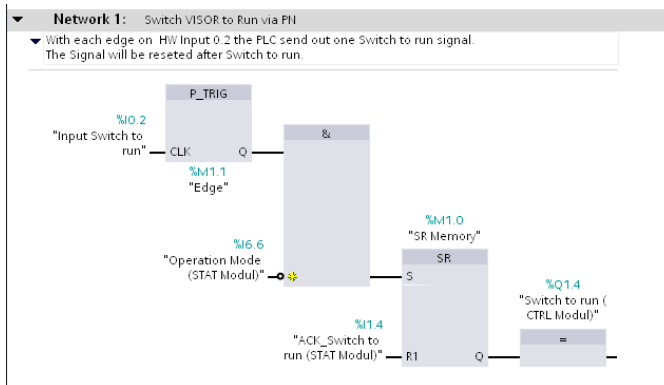


Fig. 47: Switch to Run when VISOR® in configuration mode (without error handling)

PLC example 4: Data transfer, data block on PLC, creating tags

Variable "Data Array" (type: Array of Byte) Length (34 bytes) = Payload (32) + 2 bytes (header)
 (Module "Data" with 32 bytes: User data + 1 byte: Image ID + 1 byte: Result data overrun = 34 bytes)

Data_block_1			
	Name	Data type	Start value
1	Static		
2	Jobnumber	Byte	1
3	Data Array *1	Array [0..33] of Byte	
4	Example String *2	String	

Fig. 48: Data block for data transfer

PLC example 4.1: Data transfer

Data transfer from input memory to data block with function DRPD_DAT. Access to diagnosis address via "PLC_Tags". Conversion of data of the read codes into a string with variable data length.

Name	Details
ACT_CB (START Modul)	%I 3
ACT_Switch to run (CTA)	%I 4
AI2_1[4]	267
Change Job (CTRL Modul)	%Q1 3
CTA_3_bytes[0]	277
CTA_3_bytes[1]	276
DATA_32_bytes[0]	281
DATA_32_bytes[1]	280
DIB_D06_1[DI00]	266
Edge	%M1 1
HSC_1	258
HSC_2	259
HSC_3	260

Fig. 49: Data transfer

PLC example 4.2, VISOR® telegram settings

Active	Detector	Value	Min. length	Ro
<input checked="" type="checkbox"/>	Detector 1	Datacode: -1: String length	0	
<input checked="" type="checkbox"/>	Detector 1	Datacode: -1: String	0	

Fig. 50: Settings for sample telegram in VISOR®

10 EtherNet/IP

This section explains how to operate the VISOR® vision sensor with EtherNet/IP.

10.1 Rockwell CompactLogix™ configuration example

Following is a description of the PLC settings required for data transfers between the VISOR® vision sensor and the PLC via EtherNet/IP (using Rockwell CompactLogix™ as an example).

Rockwell Studio 5000

This description shows all PLC screenshots (Studio 5000, version 30 under Windows 7) in English language. Switch Rockwell software to English if necessary.

1. Create a new project: "Create" / "New Project

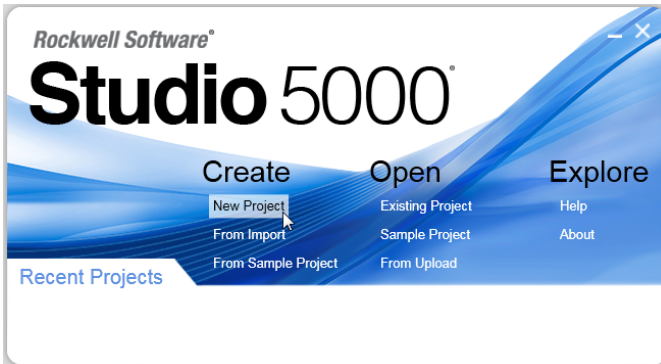


Fig. 51: EtherNet/IP Create new project

2. Select the appropriate PLC type and assign a name.

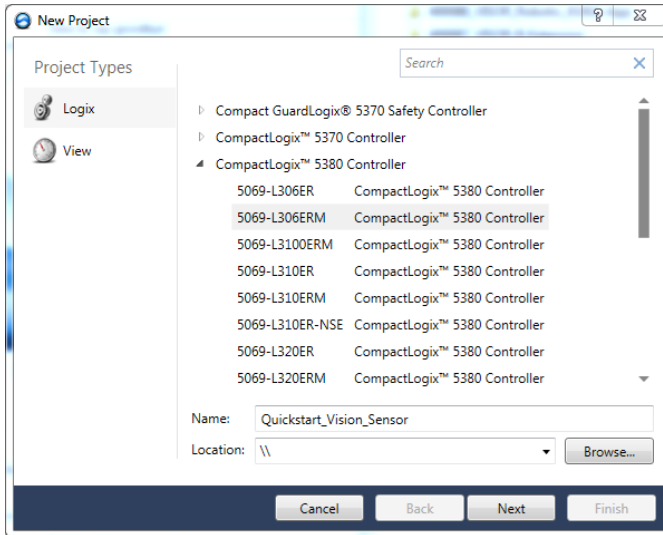


Fig. 52: EtherNet/IP Select the PLC type.

3. Apply the default settings. Click on "Finish" to create the project.

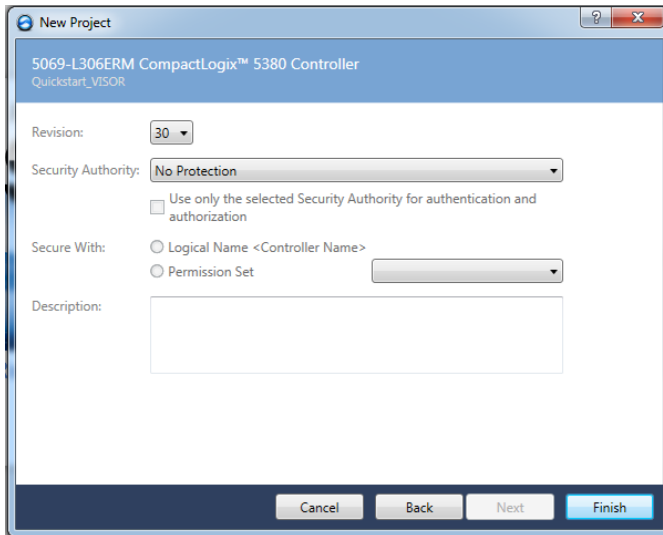


Fig. 53: EtherNet/IP Apply the default settings.

10.2 Installation of EDS file

The project view opens. In order to be able to use the EtherNet/IP functions of the VISOR® vision sensor, the latest version of the corresponding VISOR® EDS file must be installed.

If the controller does not support EDS file, follow instructions in chapter [Create module/Using a Generic Device \(without EDS file\)](#).

1. Install EDS file under "Tools" / "EDS Hardware Installation Tool".

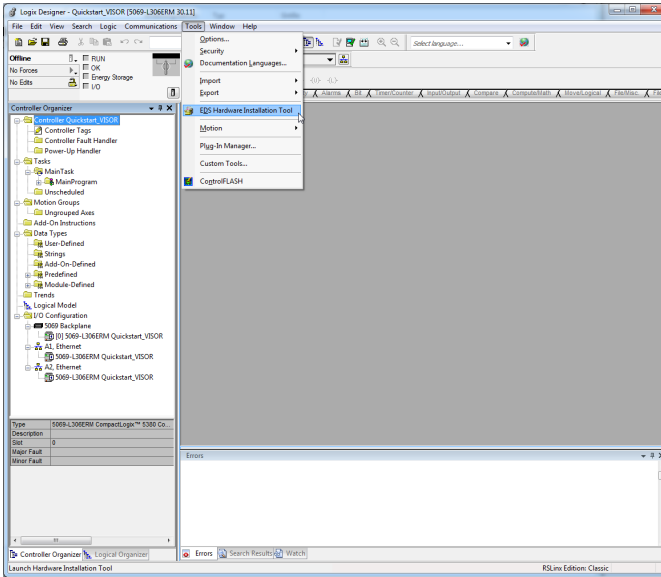


Fig. 54: Project view, Tool, EDS Hardware Installation Tool

2. Confirm information with "Next".



Fig. 55: Confirming information

3. Select "Register to EDS file(s)" in the options

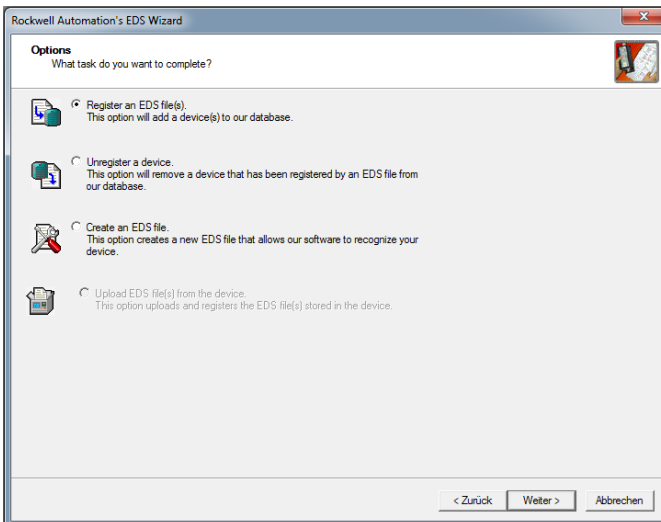


Fig. 56: Register an EDS File(s)

4. Select "Register a single file"



NOTE:

The exact same EDS file can be used for all VISOR® vision sensors.

5. Specify the path to the EDS file.

The EDS file can be found in the installation path of the VISOR® under: \SensoPart\VISOR Vision Sensor\Tools\EtherNet/IP and is also available for download at www.sensopart.com

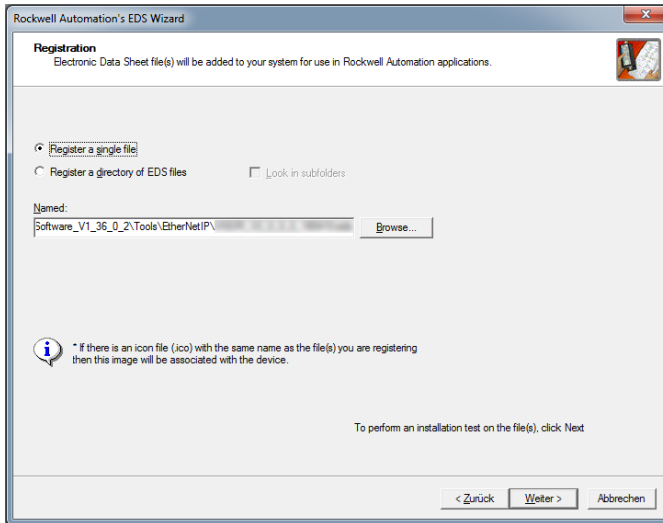


Fig. 57: Select EDS file

6. Confirm EDS file test.

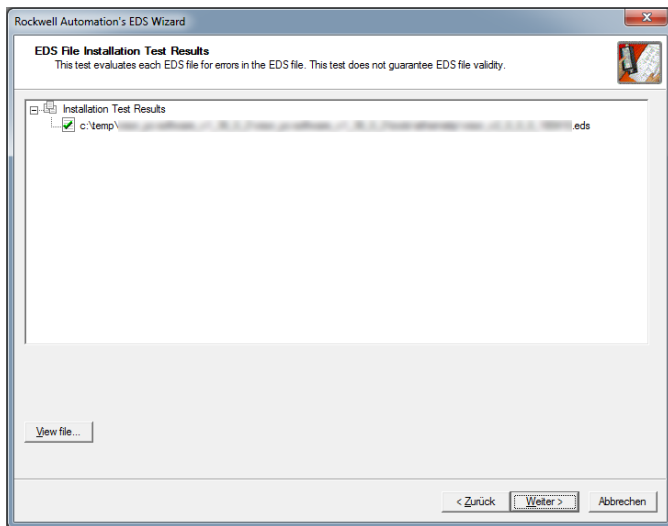


Fig. 58: EDS file test

7. Select icon if required or continue with standard icon.

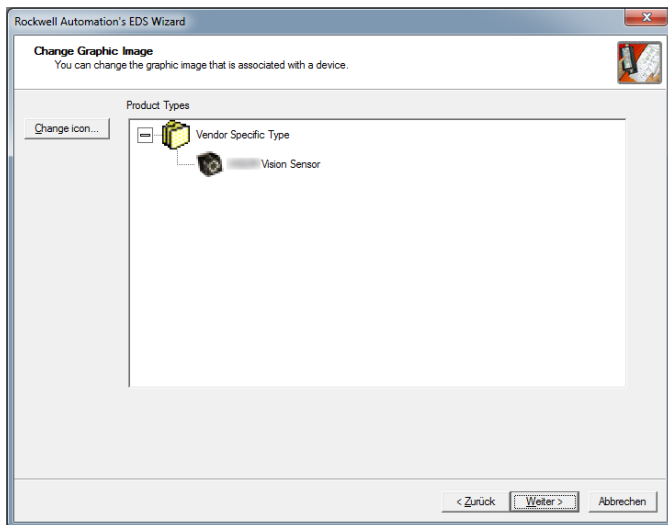


Fig. 59: Icon

8. Confirm the installation.

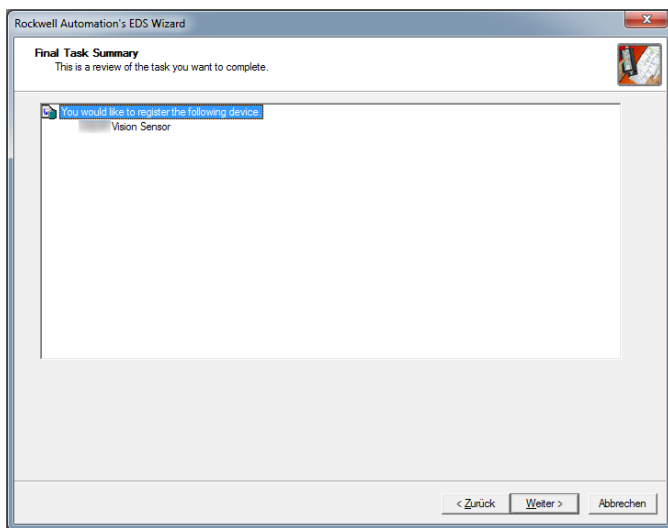


Fig. 60: Confirming the installation

9. Complete the installation with "Finish".

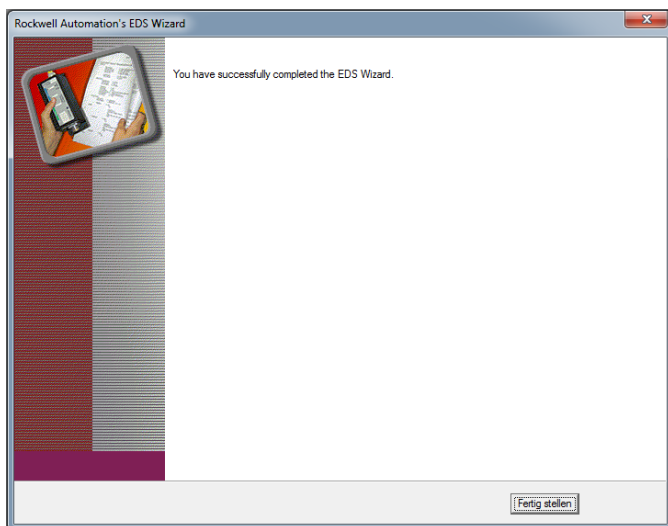


Fig. 61: Finishing the installation

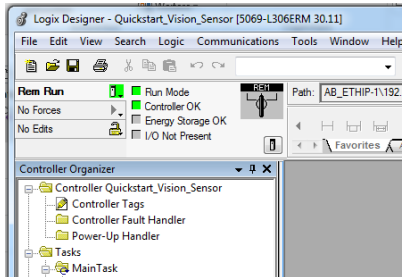
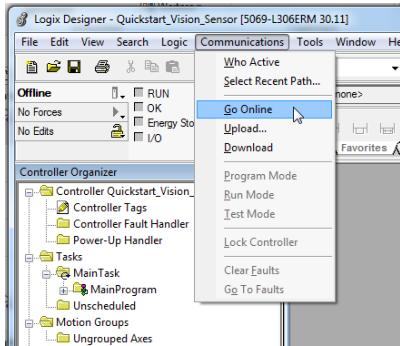
10.3 Create module

10.3.1 Selection via hardware catalog (with EDS file)

1 To go online with the project, select Communications / "Go Online".

**NOTE:**

Before this, the project path must be configured correctly.



2. Create a new module by right clicking on the desired network connection.

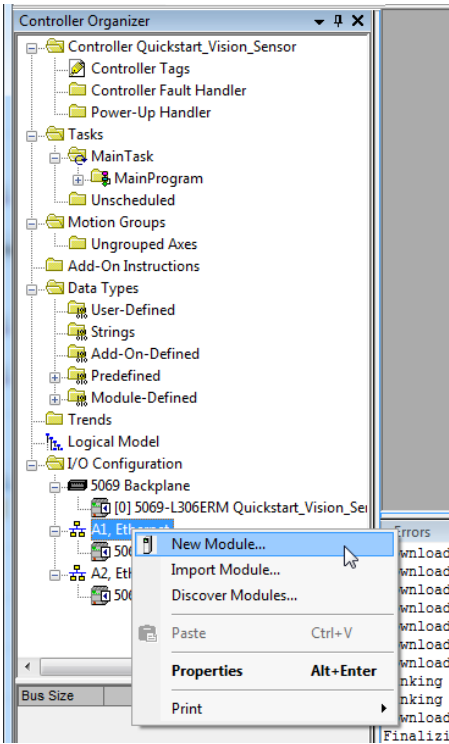


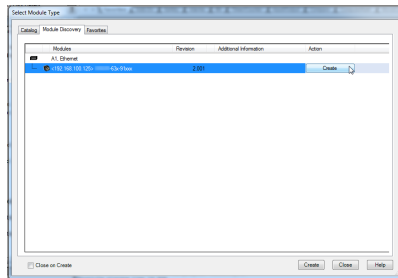
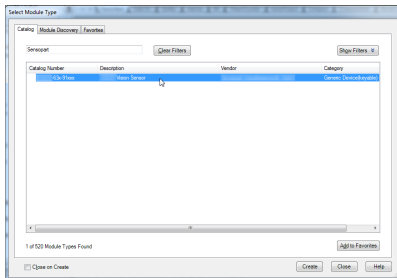
Fig. 62: Creating a new module

3. Select VISOR® from the catalog or search for available devices online.



NOTE:

For the option "Search online" the software must already be online ([see Create module / step 1](#)).



You can search for "SensoPart" in the hardware catalog. The corresponding devices are listed. Alternatively, the "Module Discovery" tab can be used to search for accessible participants.

4. Assign device name and IP address of the VISOR®.

- The device name will be used as a variable name for the data later on.
- The IP address can be read out via SensoFind.

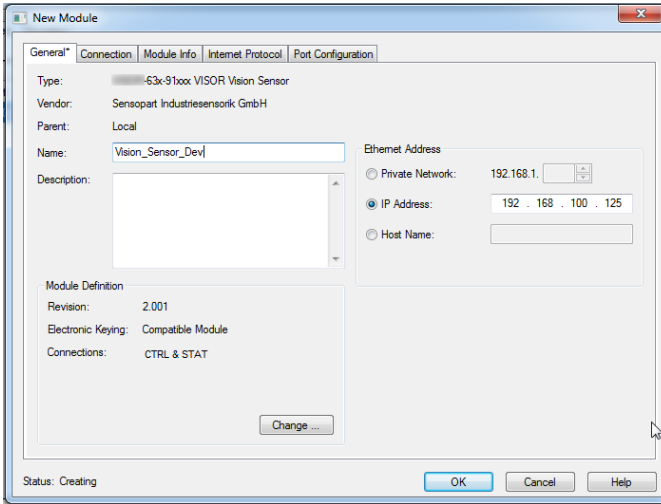
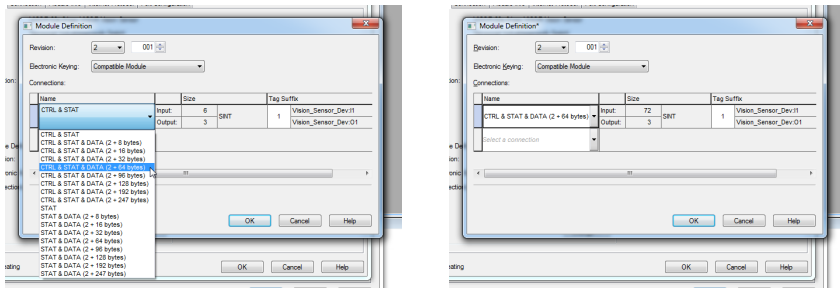


Fig. 63: Configure the device name and IP address

5. the desired modules and module sizes can be selected via "Change ...".



6. Set the desired refresh rate (RPI) in the "Connection" tab.

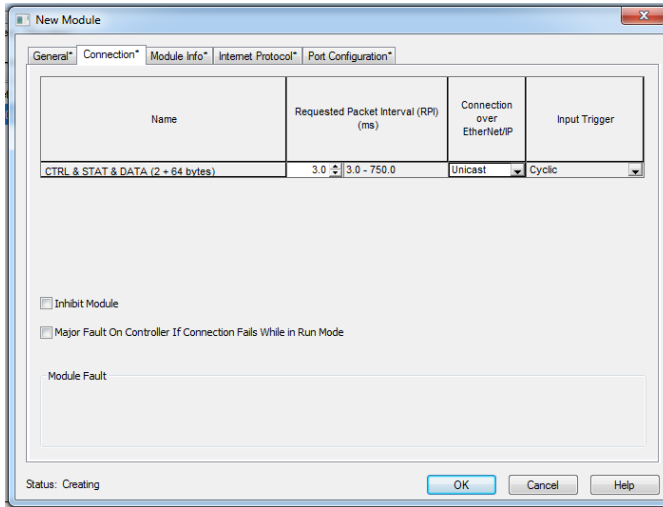


Fig. 64: Set the refresh rate.

7. Complete the participant's settings via "OK".

10.3.2 Using a Generic Device (without EDS file)

If the controller does not support EDS files, continue with the following steps.

1. Create a new module by right-clicking on the desired network connection.

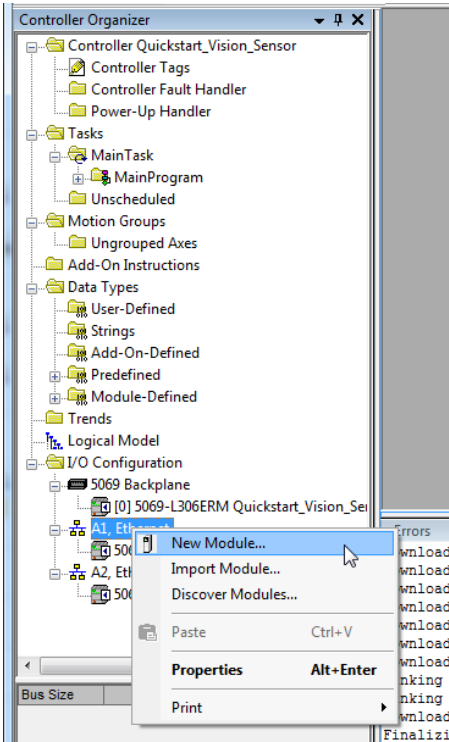


Fig. 65: Creating a new module

2. Select a module of type Ethernet Module - "Generic Ethernet Module" from the catalog

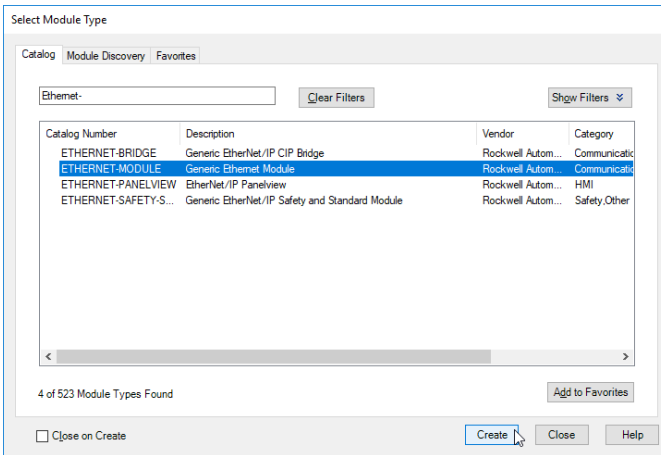


Fig. 66: Selection of "Generic Ethernet Module"

3. Assign device name and IP address of the VISOR® (A).

- The device name will be used as a variable name for the data later on.
- The IP address can be read out via SensoFind.

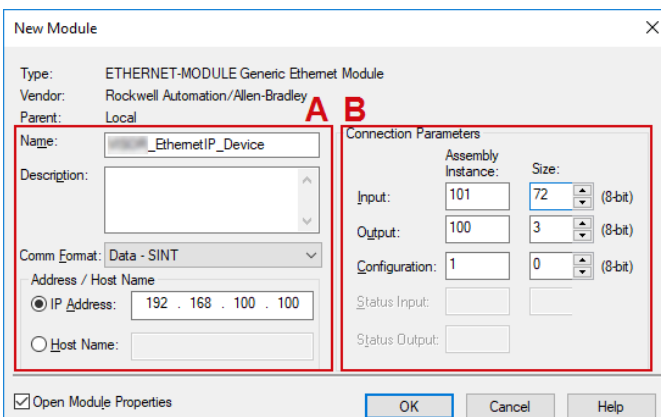


Fig. 67: Assignment of the device name and IP address

4. Change the data format to "Data - SINT" (8 bit format) with the "Comm Format" parameter (A).

5. Enter connection parameters (B) (see following table).

	Assembly instance (dec)	Size (dec)	Assembly instance (hex)	Size (hex)
Control + Status				
Input	101	6	0x65	0x06
Output	100	3	0x64	0x03
Configuration	1	0	0x01	0x00
Control + Status + Data (2+8)				
Input	102	16	0x66	0x10
Output	100	3	0x64	0x03
Configuration	1	0	0x01	0x00
Control + Status + Data (2+16)				
Input	103	24	0x67	0x18
Output	100	3	0x64	0x03
Configuration	1	0	0x01	0x00
Control + Status + Data (2+32)				
Input	104	40	0x68	0x28
Output	100	3	0x64	0x03
Configuration	1	0	0x01	0x00
Control + Status + Data (2+64)				
Input	105	72	0x69	0x48
Output	100	3	0x64	0x03
Configuration	1	0	0x01	0x00
Control + Status + Data (2+96)				
Input	105	104	0x69	0x68
Output	100	3	0x64	0x03
Configuration	1	0	0x01	0x00
Control + Status + Data (2+128)				
Input	105	136	0x69	0x88
Output	100	3	0x64	0x03
Configuration	1	0	0x01	0x00
Control + Status + Data (2+192)				
Input	105	200	0x69	0xCB
Output	100	3	0x64	0x03

	Assembly instance (dec)	Size (dec)	Assembly instance (hex)	Size (hex)
Configuration	1	0	0x01	0x00
Control + Status + Data (2+247)				
Input	105	255	0x69	0xFF
Output	100	3	0x64	0x03
Configuration	1	0	0x01	0x00

10.4 Load the project onto the PLC

1. Download the project to the PLC via "Communications" / "Download".



NOTE:

For this the software must already be online (see [Create module / step 1](#)).

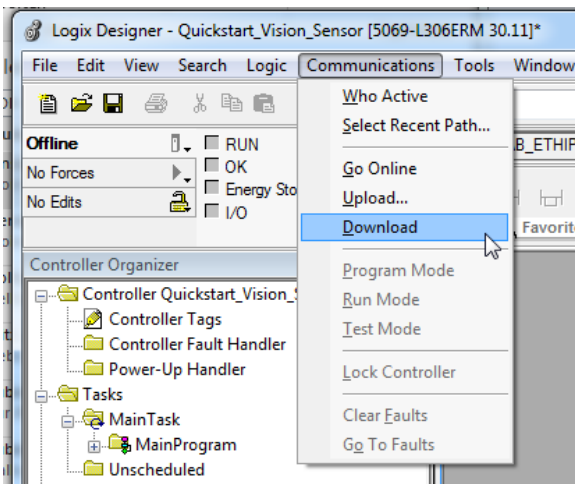


Fig. 68: Download

2. Check the notes and confirm with "Download".

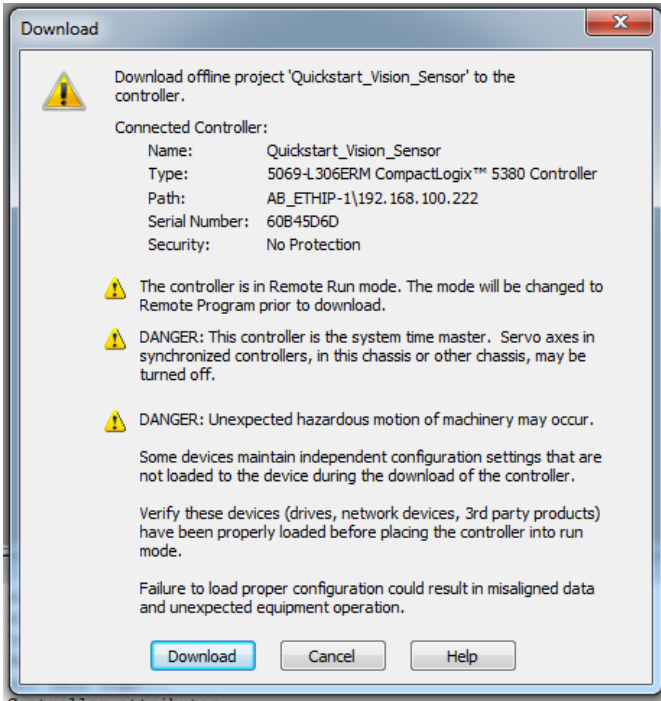


Fig. 69: Information

3. After a successful download, the VISOR® status is "Running".

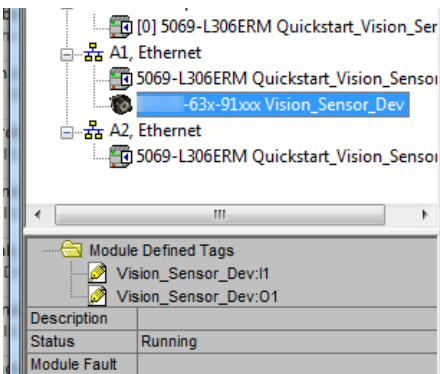


Fig. 70: Status "Running"

10.5 Mapping of output data

The input data is assigned as follows: (select module CNTL + STAT + Data (2+128))

.... I1.Data[0] – I1.data [5] "Status" module (see description [Module 2: "Status" \(from VISOR® to PLC\) \(Page 33\)](#))

e.g. ... I1.Data [3] = Job number

.... I1.Data[4] = Image_ID

The data module is appended directly. Start of Data module from ... I1.Data[6] - I1.Data[135]

Here the data is inserted as indicated in SensoConfig under "Output" / "Telegram".

Additional information: [Defining telegrams / data output in SensoConfig \(Page 16\)](#)

Name	Value	Force Mask	Style	De
- Vision_Sensor_Dev:11.Data	{...}	{...}	Decimal	SII
+ Vision_Sensor_Dev:11.Data[0]	1		Decimal	SII
+ Vision_Sensor_Dev:11.Data[1]	0		Decimal	SII
+ Vision_Sensor_Dev:11.Data[2]	0		Decimal	SII
+ Vision_Sensor_Dev:11.Data[3]	1		Decimal	SII
+ Vision_Sensor_Dev:11.Data[4]	6		Decimal	SII
+ Vision_Sensor_Dev:11.Data[5]	0		Decimal	SII
+ Vision_Sensor_Dev:11.Data[6]	6		Decimal	SII
+ Vision_Sensor_Dev:11.Data[7]	0		Decimal	SII
+ Vision_Sensor_Dev:11.Data[8]	0		Decimal	SII
+ Vision_Sensor_Dev:11.Data[9]	0		Decimal	SII
+ Vision_Sensor_Dev:11.Data[10]	0		Decimal	SII
+ Vision_Sensor_Dev:11.Data[11]	0		Decimal	SII
+ Vision_Sensor_Dev:11.Data[12]	0		Decimal	SII
+ Vision_Sensor_Dev:11.Data[13]	0		Decimal	SII
+ Vision_Sensor_Dev:11.Data[14]	0		Decimal	SII
+ Vision_Sensor_Dev:11.Data[15]	0		Decimal	SII
+ Vision_Sensor_Dev:11.Data[16]	0		Decimal	SII

Fig. 71: Output data

Conversion of binary values

All detector-specific payloads with decimal places will be transmitted as integers multiplied by 1000, and accordingly must be divided by 1000 after the data is received. The values are transferred in the format "Big-endian". The length is based on the value, e.g., score 32 bits (DWord).

Controller Organizer

- Controller Quickstart_Vision_Sensor
 - Controller Tags
 - Controller Fault Handler
 - Power-Up Handler
 - Tasks
 - MainTask
 - MainProgram
 - Parameters and Local Tags
 - MainRoutine
 - Motion Groups
 - Ungrouped Axes
 - Add-On Instructions
 - Data Types
 - User-Defined
 - Strings
 - Add-On-Defined
 - Predefined
 - Module-Defined
 - Trends
 - Logical Model
 - I/O Configuration
 - 5069 Backplane
 - 5069-L306ERM Quickstart_Vision_Ser
 - A1, Ethernet
 - 5069-L306ERM Quickstart_Vision_Ser
 - 63x-91xxx Vision_Sensor_Dev
 - A2, Ethernet

Watch

| Name | Scope | Value | Force Mask | Description |
|---------------|-------------|--------|------------|-------------|
| Angle | MainProgram | -116 | | |
| Pos_X | MainProgram | 409395 | | |
| Pos_Y | MainProgram | 422919 | | |
| Vision_Sensor | Controller | [...] | [...] | |
| Vision_Sensor | Controller | [...] | [...] | |

Fig. 72: Swapping the byte order

Results

| Detector | Score | Time | Detector t |
|----------|-------|------|------------|
| 1 | 99.8 | 27ms | Contour |

No. objects: 1 No. of valid objects: 1

| Score | Position X [px] | Position Y [px] | Angle | Scale | Delta pos-X [px] | Delta pos-Y [px] | Delta angle | Position control | |
|-------|-----------------|-----------------|-------|-------|------------------|------------------|-------------|------------------|-----|
| 1 | 99.8 | 409.4 | 422.8 | -0.1° | 1 | -0.0 | -0.2 | -0.1° | Off |

Statistics

| | | |
|------------------------|-----|---------|
| Count | 1 | Reset |
| Pass | 1 | 100.00% |
| Fail | 0 | 0.00% |
| Minimum execution time | 4ms | |
| Maximum execution time | 4ms | |
| Average execution time | 4ms | |

10.6 PLC example programs

The following PLC example programs show some basic functions.

PLC example 1: Trigger when VISOR® Ready

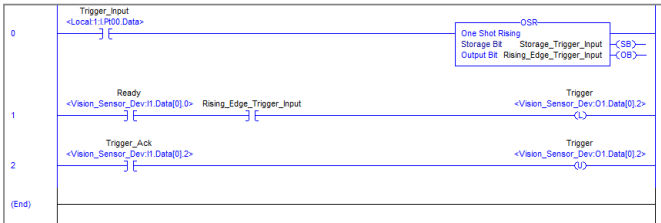


Fig. 73: Trigger when VISOR® Ready, (without error handling)

PLC example 2: Job change when VISOR® Ready

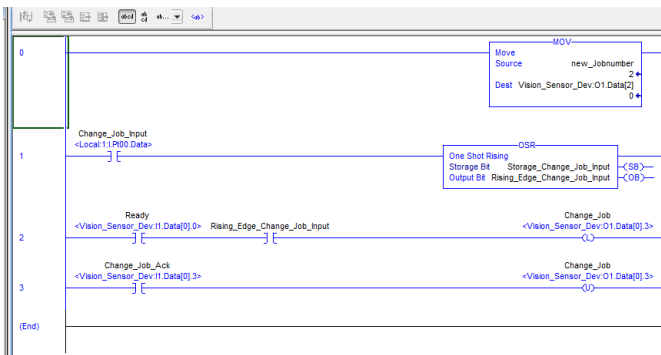


Fig. 74: Job change when VISOR® Ready, (without error handling)

PLC example 3: Switch to Run when VISOR® in configuration mode

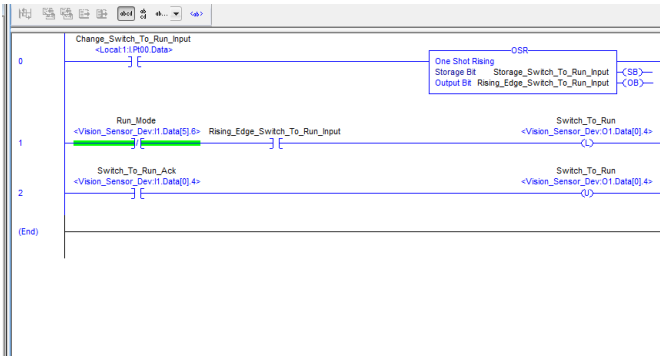


Fig. 75: Switch to Run when VISOR® in configuration mode (without error handling)

11 Telegrams and data output

This section describes the telegrams available for the VISOR® vision sensor. These telegrams can be sent to the VISOR® vision sensor through various interfaces.

- Ethernet TCP/IP
- PROFINET (Request / Response module)

The telegrams are available in ASCII and Binary format. The format is defined in the module "SensoConfig", in the tab "Telegram" of the setup "Output".

The following settings are possible:

| Communication | TCP / IP | EtherNet/IP | PROFINET |
|-----------------|----------------|-------------|----------|
| Telegram format | ASCII / Binary | Binary | Binary |

11.1 Overview telegrams

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

VISOR® General

- **Reset statistics (RST) (ASCII / Binary)**
The "Reset statistics" telegram can be used to reset the VISOR® vision sensor's internal statistics counter.

VISOR® Control

- **Trigger (TRG) (ASCII / binary)**
With the telegram "Trigger", an image can be acquired. Some commands need additional image acquisition. The result data of the evaluation are output via the "Out" port.
- **Extended trigger (TRX) (ASCII / binary)**
This telegram "Extended trigger" is an expansion of the "trigger" telegram. Besides the result data, there is also the option to assign an ID or to receive information about the operating mode (run/config). Unlike the "trigger" telegram, the result data of the "Extended trigger" telegram are also transferred via the "In" port.
- **Trigger Robotics (TRR) (ASCII / Binary)**
With the telegram "Trigger", an image can be acquired. In addition to image acquisition, the robot tool center point (TCP) can be transferred. The TCP is used to calculate the position values.

- **Set Trigger ID (STI)** (ASCII / Binary)
With the telegram "Set Trigger ID", a Trigger ID can be set. Der Identifier wird für die nächste Bildaufnahme verwendet und kann bspw. als Dateiname gesetzt werden.
- **Job change (CJB)** (ASCII / binary)
The "Job change" telegram will trigger a job change on the VISOR® vision sensor.
- **Job change permanent (CJP)** (ASCII / Binary)
The "Job change permanent" telegram will trigger a permanent job change on the VISOR® vision sensor. The job is run again after restarting.
- **Job change by job name (CJN)** (ASCII / Binary)
The "Job change by job name" telegram will trigger a job change on the VISOR® vision sensor. The job will be run by job name. You can read the job names by using the "Read job list" telegram, for example.

VISOR® Job settings

- **Auto Working distance (AFC)** (ASCII / Binary)
The "Auto operating distance" telegram can be used to have the working distance for the job be automatically determined.
- **Set working distance (SFC)** (ASCII / Binary)
The "Set working distance" telegram can be used to change the working distance for the job.
- **Read working distance (GFC)** (ASCII / binary)
The "Read working distance" telegram can be used to read the current working distance for the job.
- **Auto shutter speed (ASH)** (ASCII / Binary)
The "Auto shutter speed" telegram can be used to have the shutter time for the job be automatically determined.
- **Set shutter speed (SSP/SST)** (ASCII / Binary)
With the telegram "Set shutter speed", the shutter speed of the job can be changed. This telegram can, for example, be used for brightness compensation.
- **Read shutter speed (GSH)** (ASCII / Binary)
With the telegram "Read shutter speed", the set shutter speed of the job can be read.
- **Set gain (SGA)** (ASCII / binary)
With the telegram "Set gain", the gain of the job can be changed. This telegram can, for example, be used for brightness compensation.
- **Read gain (GGA)** (ASCII / binary)
With the telegram "Read gain", the set gain of the job can be read.
- **Set parameters (SPP/SPT)** (ASCII / binary)
With the telegram "Set parameter", the detector parameters can be adjusted, e.g. reference strings, detector thresholds.

- **Read parameter (GPA) (ASCII / binary)**
With the telegram "Read parameter", the set parameters of the detectors can be read.
- **Set search range / ROI (SRP/SRT) (ASCII / binary)**
With the telegram "Set ROI", the position of the selected detector can be changed.
- **Read search range / ROI (GRI) (ASCII / Binary)**
With the telegram "Read ROI", the position of the selected detector can be read.
- **Read job list (GJL) (ASCII / binary)**
The "Get job list" telegram can be used to output a list of all available jobs on the VISOR® vision sensor.
- **Read detector list (GDL) (ASCII / binary)**
With the telegram "Read detector list", a list of all detectors in the current job will be displayed.
- **Teach-in detector (TED) (ASCII / binary)**
The "Teach detector" telegram will result in the specified detector being re-taught (available only for Pattern matching and Contour).
- **Set trigger delay (STD) (ASCII / Binary)**
With the telegram "Set trigger delay", a delay for starting a trigger can be set (in time (ms) or encoder steps).
- **Read trigger delay (GTD) (ASCII / Binary)**
With the telegram "Read trigger delay", the set delay for starting a trigger can be read.
- **Save Job Permanently (SJP) (ASCII / binary)**
The "Save job permanently" telegram will take all the parameters that were previously set temporarily and copy them to a job set.

VISOR® Calibration

- **Calibration: Initialize (CCD) (ASCII / binary)**
The point pair list is initialized with the telegram "Calibration: Initialize point pair list".
- **Calibration: Add world point (CAW) (ASCII / binary)**
With the telegram "Calibration: Add world point" a world point (fiducial or point pair) is added to the point pair list. The telegram can be used for the calibration method Point pair list (Robotics) and Calibration plate (Robotics).
- **Calibration: Point pair list (CCL) (ASCII / binary)**
With the telegram "Calibration: Point pair list" the calibration is carried out using the point pair list in the current job.
- **Calibration: Validate point pair list (ASCII / binary)**
With the telegram "Calibration: Validate point list", the calibration is validated using the point list.

- **Calibration: Calibration Plate (CCP)** (ASCII / Binary)
With the telegram "Calibration: Calibration plate", the calibration is carried out using the calibration plate.
- **Set fiducial (CSF)** (ASCII / binary)
With the telegram "Set fiducial", the fiducials are set using the point list in the current job.
- **Calibration: Add Image (CAI)** (ASCII / Binary)
The "Add image" telegram triggers an image acquisition and if a calibration plate is found, an image is added to the calibration object. The telegram can be used for calibration method Multi-image calibration and calibration method Calibration plate (Robotics).
- **Calibration: Multi-image (CMP)** (ASCII / binary)
With the telegram "Calibration: Multi-image" a calibration is carried out and an existing calibration object is accessed.
- **Calibration: Robotics Multi-image (CRP)** (ASCII / Binary)
With the telegram "Multi-image, robot" a calibration is carried out using the calibration plate.
- **Calibration: Copy calibration (CCC)** (ASCII / binary)
With the telegram "Calibration: Copy calibration", the calibration of the current job is copied to the selected destination.
- **Calibration: Set parameters (CSP)** (ASCII / binary)
With the telegram "Calibration: Set parameter", the parameter values for the calibration can be set.
- **Calibration: read parameters (CGP)** (ASCII / binary)
With the telegram "Calibration: Read parameter", the set parameter values of the calibration can be read.

VISOR® Visualization

- **Get image (GIM)** (ASCII / binary)
The "Get image" telegram can be used to get the image from the VISOR® vision sensor.

VISOR® Service (available only on port 1998 and in ASCII format)

- **Update visualization data (UVR)** (ASCII)
The "Update visualization data" telegram is used to update visualization data such as image, detector information and results.
- **Read sensor identity (GSI)** (ASCII)
With the telegram "Read sensor identity", the current firmware status as well as the hardware type can be queried.

- **Update firmware (UFW) (ASCII)**

With the telegram "Update firmware", a firmware update is started. The firmware file must first be loaded onto the VISOR® vision sensor.

- **Read jobset (SJS) (ASCII)**

The "Set job set" telegram can be used to change the VISOR® vision sensor's job set. The job set file must first be loaded onto the VISOR®.

- **Save jobset (GJS) (ASCII)**

The "Save jobset" telegram can be used to read the VISOR® vision sensor's job set.

Data output

This section contains information about the data output (e.g. which format the individual results will have).

Data output ASCII

- [General](#)
- [Base values](#)
- [Position](#)
- [Measurement](#)
- [Identification](#)
- [Identification - quality](#)
- [Color](#)
- [Counting / number](#)
- [Extended](#)

Data output Binary

- [General](#)
- [Base values](#)
- [Position](#)
- [Measurement](#)
- [Identification](#)
- [Identification - quality](#)
- [Color](#)
- [Counting / number](#)
- [Extended](#)

11.2 Telegrams: Availability and supported interfaces

Device variants

| | | | |
|------------|-------------|----------|--------------|
| ALL | Allround | S | Standard |
| OB | Object | A | Advanced |
| CR | Code reader | P | Professional |
| RO | Robotic | | |

✓ available

[] Limited availability: differences between versions < 2 and ≥ 2

Interfaces

| | |
|----------|------------------------|
| 1 | Ethernet TCP IN (2006) |
| 2 | PROFINET |
| 3 | EtherNet/IP |
| 4 | Service Port (1998) |

| Telegram | ALL | | OB | | CR | | | RO | | Interfaces | | | | From version |
|-----------------------------|-----|---|----|---|----|---|---|----|---|------------|---|---|---|--------------|
| | A | P | S | A | S | A | P | A | P | 1 | 2 | 3 | 4 | |
| VISOR® General | | | | | | | | | | | | | | |
| Reset statistics (RST) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.18 |
| VISOR® Control | | | | | | | | | | | | | | |
| Trigger (TRG) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |
| Extended trigger (TRX) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.6 |
| Trigger Robotics (TRR) | | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.2 |
| Set Trigger ID (STI) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.2 |
| Job change (CJB) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |
| Job Change Permanent (CJP) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.18 |
| Job change by name (CJN) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.0 |
| VISOR® Job settings | | | | | | | | | | | | | | |
| Auto working distance (AFC) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.0 |
| Set working distance (SFC) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.0 |
| Read working distance (GFC) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.0 |
| Auto Shutter Speed (ASH) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.0 |
| Set Shutter Speed (SSP/SST) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |
| Read shutter speed (GSH) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |
| Set gain (SGA) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.6 |
| Read gain (GGA) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.6 |
| Set Parameter (SPP/SPT) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |
| Read Parameter (GPA) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |

| Telegram | ALL | | OB | | CR | | | RO | | Interfaces | | | | From version |
|--------------------------------------|-----|---|----|-----|----|---|---|----|---|------------|---|---|---|--------------|
| | A | P | S | A | S | A | P | A | P | 1 | 2 | 3 | 4 | |
| Set ROI (SRP/SRT) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |
| Read ROI (GRI) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |
| Read job list (GJL) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.18 |
| Read Detector List (GDL) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.18 |
| Teach detector (TED) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |
| Set trigger delay (STD) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.22 |
| Read Trigger Delay (GTD) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.22 |
| Save Job Permanently (SJP) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.0 |
| VISOR® Calibration | | | | | | | | | | | | | | |
| Initialization (CCD) | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.18 |
| Add world point (CAW) | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.22 |
| Calibration: Point Pair List (CCL) | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.18 |
| Validate calibration (CVL) | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.18 |
| Calibration: Calibration Plate (CCP) | [] | ✓ | | [] | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.19 |
| Set fiducials (CSF) | | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.22 |
| Add image (CAI) | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.2 |
| Multi-Image (CMP) | ✓ | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.2 |
| Robotics Multi-Image (CRP) | | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 2.2 |
| Copy calibration (CCC) | | ✓ | | | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.19 |
| Set parameters (CSP) | [] | ✓ | | [] | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.22 |
| Read parameters (CGP) | [] | ✓ | | [] | | | | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.22 |
| VISOR® Visualization | | | | | | | | | | | | | | |
| Get Image (GIM) | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | | 1.0 |

| Telegram | ALL | | OB | | CR | | | RO | | Interfaces | | | | From version |
|---------------------------------|-----|---|----|---|----|---|---|----|---|------------|---|---|---|--------------|
| | A | P | S | A | S | A | P | A | P | 1 | 2 | 3 | 4 | |
| VISOR® Service | | | | | | | | | | | | | | |
| Update visualization data (UVR) | ✓ | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | | | | ✓ | 1.22 |
| Read sensor identity (GSI) | ✓ | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | | | | ✓ | 1.19 |
| Update firmware (UFW) | ✓ | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | | | | ✓ | 1.19 |
| Read job set (SJS) | ✓ | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | | | | ✓ | 1.19 |
| Save job set (GJS) | ✓ | ✓ | | ✓ | | ✓ | ✓ | ✓ | ✓ | | | | ✓ | 1.19 |

Please refer to the following as well: [Overview telegrams \(Page 87\)](#)

11.3 Error codes

| Error code | Error code HEX | Description |
|------------|----------------|---|
| 000 | 0x00 | Successful |
| 001 | 0x01 | Error |
| 003 | 0x03 | Invalid parameter data |
| 005 | 0x05 | Invalid telegram |
| 006 | 0x06 | Input parameters with invalid size or invalid value |
| 007 | 0x07 | File does not exist |
| 008 | 0x08 | Recorder off |
| 009 | 0x09 | Matching image of requested type not found |
| 010 | 0x0A | Invalid file name or length |
| 011 | 0x0B | Invalid data length |
| 012 | 0x0C | Not allowed due to job set mismatch |
| 013 | 0x0D | Failed to start new job from job set |
| 016 | 0x10 | Firmware version mismatch |
| 018 | 0x12 | Calibration plate data not available |
| 020 | 0x14 | More than one vis file present |
| 021 | 0x15 | Sensor type does not match for vis file |
| 030 | 0x1E | Calibration not activated / Calibration not supported |

| Error code | Error code HEX | Description |
|------------|----------------|--|
| 031 | 0x1F | Calibration copy error |
| 032 | 0x20 | Mismatched input conditions for destination job |
| 033 | 0x21 | Calibration / validation error |
| 034 | 0x22 | Invalid number of points |
| 035 | 0x23 | Calibration error: Add point, e.g. last job result failed |
| 036 | 0x24 | Invalid fiducial |
| 037 | 0x25 | Job set protection error: "Permanent" job change is not allowed |
| 038 | 0x26 | Parameter values are not available to write / read |
| 039 | 0x27 | Sensor is in configuration mode. The telegram was rejected |
| 040 | 0x28 | Write / read error for parameter value |
| 041 | 0x29 | No matching job found |
| 042 | 0x2A | Formatting error |
| 043 | 0x2B | Job set / Job saving error |
| 044 | 0x2C | Focus lock time exceeded |
| 045 | 0x2D | Error with multiple files |
| 046 | 0x2E | Working distance could not be determined |
| 047 | 0x2F | "Min. processing time per image" was not observed |
| 048 | 0x30 | Search range size (ROI) does not match |
| 049 | 0x31 | Search range (ROI) Freeform not selected |
| 050 | 0x32 | Calibration method does not match |
| 051 | 0x33 | No calibration plate found |
| 052 | 0x34 | Number of images too small |
| 053 | 0x35 | No calibration possible: distance between tool positions not plausible |
| 054 | 0x36 | Rotation between images not sufficient |
| 055 | 0x37 | Tilt between the images not sufficient |

11.4 Description Telegrams ASCII

11.4.1 General

Reset statistics (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Reset Statistics (RST) Request string to sensor (ASCII) | | |
|---|--|--------------------------------------|
| Byte no. | Content | Meaning |
| 1 | R | Reset statistics |
| 2 | S | |
| 3 | T | |
| Example: | RST | |
| Reset Statistics (RST) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | R | Reset statistics |
| 2 | S | |
| 3 | T | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| Example: | RSTP | |
| Additional information: | | |
| Accepted in run mode: | Yes | |
| Accepted in configuration mode: | No | |
| Accepted when Ready is low: | Yes | |
| Status of Ready signal during processing: | No change | |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) | |
| End of telegram: | Max. 4 bytes (optional) | |

11.4.2 Control

Trigger (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Trigger (TRG) Request string to sensor (ASCII) | | |
|--|--|---|
| Byte no. | Content | Meaning |
| 1 | T | Trigger (simple trigger, in-port) |
| 2 | R | |
| 3 | G | |
| Example: | TRG | |
| Trigger (TRG) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | T | Trigger (response to command trigger without index, via port 2006. If defined: Result data without index via port 2005) |
| 2 | R | |
| 3 | G | |
| 4 | P | P: (Pass) Success |
| | F | F: (Fail) Error |
| Example: | TRGP | |
| Additional information: | | |
| Accepted in run mode: | Yes | |
| Accepted in configuration mode: | Yes | |
| Accepted when Ready is low: | No | |
| Status of Ready signal during processing: | Low | |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) | |
| End of telegram: | Max. 4 bytes (optional) | |

Extended trigger (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Extended Trigger (TRX) Request string to sensor (ASCII) | | |
|---|----------------------|--|
| Byte no. | Content | Meaning |
| 1 | T | Extended trigger, (trigger with index, for correlation of trigger to corresponding result data, via port 2006) |
| 2 | R | |
| 3 | X | |
| 4 - 5 | X | Length of following data (n) |
| 6 ... n | X | Data |
| Example: | TRX06MyPart | |
| Extended Trigger (TRX) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | T | Extended trigger, (response to trigger with index and result data, via port 2006, for correlation of trigger to corresponding result. Result data without index via port 2005) |
| 2 | R | |
| 3 | X | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 6 | X | Length of following data (n) |
| 7 ... n | X | Data of sending command |
| n+1 | C
R | C = Config
R = Run |
| n+2 ... n+9 | X | Length of following result data (n) |
| n+9 ... m | X | Result data |
| Example: | TRX06MyPartR00000000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | Yes |
| Accepted when Ready is low: | | No |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Trigger Robotics (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Trigger Robotics (TRR) Request string to sensor (ASCII) | | |
|--|--|---|
| Byte no. | Content | Meaning |
| 1 | T | Trigger Robotics |
| 2 | R | |
| 3 | R | |
| 4 | 1 | Request version |
| 5-6 | X | Length of trigger identifier |
| 7-n | X | Trigger Identifier |
| n+1...n+8 | X | Pose_TCP Pos. X
(in user unit * 1000) |
| n+9...n+16 | X | Pose_TCP Pos. Y
(in user unit * 1000) |
| n+17...n+24 | X | Pose_TCP Pos. Z
(in user unit * 1000) |
| n+25...n+32 | X | Pose_TCP Angle X
(in degrees * 1000) |
| n+33...n+40 | X | Pose_TCP Angle Y
(in degrees * 1000) |
| n+41...n+48 | X | Pose_TCP Angle Z
(in degrees * 1000) |
| Example: | TRR104Part000040040000500500006006000070070000
800800009009 | |
| Trigger Robotics (TRR) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | T | Trigger (response to command trigger without index, via port 2006. If defined: Result data without index via port 2005) |
| 2 | R | |
| 3 | R | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5-7 | X | Error codes (Page 94) |
| 7-8 | X | Length of trigger identifier |

| | | |
|---|------------------------|--|
| 9-n | X | Trigger Identifier |
| n+1 | X | Operation Mode
C = Config
R = Run |
| n+2...n+9 | X | Length of result data |
| n+10...m | X | Result data |
| Example: | TRRP00004PartR00000000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | Yes |
| Accepted when Ready is low: | | No |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | |

Note: For "Calibration plate (Robotics)" and "Point pair list (Robotics)" only the X and Y position are taken into account. The other values (position Z and rotations) must be 0.

Set Trigger ID (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Set Trigger ID (STI) Request string to sensor (ASCII) | | |
|---|--------------|--|
| Byte no. | Content | Meaning |
| 1 | S | Set Trigger ID |
| 2 | T | |
| 3 | I | |
| 4 | 1 | Request version |
| 5-6 | x | Length of the following data (max 99) |
| 7-n | x | Trigger ID |
| Example: | STI106MyPart | |
| Set Trigger ID (STI) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | S | Set Trigger ID |
| 2 | T | |
| 3 | I | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5-7 | x | Error codes (Page 94) |
| Example: | STIP000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | Yes |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | |

Job change (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Job change (CJB) Request string to sensor (ASCII) | | |
|--|----------|--|
| Byte no. | Content | Meaning |
| 1 | C | Job change |
| 2 | J | |
| 3 | B | |
| 4 - 6 | X | Job number |
| Example: | CJB005 | |
| Job change (CJB) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | C | Job change |
| 2 | J | |
| 3 | B | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 | T
F | Triggered
Freerun |
| 6 - 8 | X | Job number |
| Example 1: | CJBPT005 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |



NOTE:

If an error occurs during the job change, it is possible to change to Job 1.

Job Change Permanent (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Job Change Permanent (CJP) Request string to sensor (ASCII) | | |
|--|----------|--|
| Byte no. | Content | Meaning |
| 1 | C | Job change permanent (Change Job Permanently) |
| 2 | J | |
| 3 | P | |
| 4 - 6 | X | Job number |
| Example: | CJP005 | |
| Job Change Permanent (CJP) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | C | Job change permanent (Change Job Permanently) |
| 2 | J | |
| 3 | P | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 | T
F | Triggered
Freerun |
| 6 - 8 | X | Job number |
| Example 1: | CJPPT005 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |



NOTE:

If an error occurs during the job change, it is possible to change to Job 1.

Job change by job name (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)


| Job change by job name (CJN) Request string to sensor (ASCII) | | |
|---|--------------|--|
| Byte no. | Content | Meaning |
| 1 | C | Job change by name |
| 2 | J | |
| 3 | N | |
| 4 | 1 | Request version |
| 5 - 7 | X | Job name length |
| 8 - n | X | Job name |
| Example: | CJN1005Myjob | |
| Job change by job name (CJN) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | C | Job change by name |
| 2 | J | |
| 3 | N | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 | X | Trigger mode
T: Trigger
F: Free run |
| Example: | CJNP000T | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |


11.4.3 Job settings

Auto working distance (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Auto working distance (AFC) Request string to sensor (ASCII) | | |
|---|---|---|
| Byte no. | Content | Meaning |
| 1 | A | Auto Working Distance (Auto Focus) |
| 2 | F | |
| 3 | C | |
| 4 | 1 | Request version |
| 5 | X | 0: Temporary
1: Permanent |
| 6 | X | Step size 1-5 |
| 7 - 9 | X | Focus selection
0: Maximum score
1: Min. Arbeitsabstand
2: Max. working distance
3: Average working distance
4: Median working distance
5: Maximum score and all planes |
| 10 | X | Focus unit
0: Millimeters
1: Steps |
| 11 | X | Working distance selection
0: Default range
1: Specified range |
| |  NOTE:
The following byte sequence is only relevant if "Distance range selection" has been set to 1. | |
| 12 - 19 | X | Start of working area (close) |
| 20 - 27 | X | End of working area (far) |
| Example: | Example 1: AFC11100500
Example 2: AFC111005010001000000100000 | |
| Auto working distance (AFC) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |

| | | |
|---|--|---|
| 1 | A | Auto Working Distance (Auto Focus) |
| 2 | F | |
| 3 | C | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 10 | X | Focus selection = 5 ; Number of Focus selection distances found = 1-4 ; 1 |
| |  NOTE:
The following fields [Distance value / Score value] are repeated for each number of distances found. | |
| 11 - 18 | X | Distance value in mm *1000 or in steps |
| 19 - 26 | X | Score value in %*1000 |
| Example: | AFCP000002000000950000009000009300000089000 | |
| Additional information: | | |
| Accepted in run mode: | Yes | |
| Accepted in configuration mode: | No | |
| Accepted when Ready is low: | Yes | |
| Status of Ready signal during processing: | No change | |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) | |
| End of telegram: | Max. 4 bytes (optional) | |

Set working distance (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Set working distance (SFC) Request string to sensor (ASCII) | | |
|---|------------------|---|
| Byte no. | Content | Meaning |
| 1 | S | Working distance (Set Focus) |
| 2 | F | |
| 3 | C | |
| 4 | 1 | Request version |
| 5 | X | 0: Temporary
1: Permanent |
| 6 | X | Movement
0: Absolute
1: Relative
2: Absolute with reinitialization |
| 7 | X | Unit
0: 1/1000 millimeters
4: Steps |
| 8 - 15 | X | Distance value in mm * 1000 or in steps |
| Example: | SFC111400000010 | |
| Set working distance (SFC) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | S | Working distance (Set Focus) |
| 2 | F | |
| 3 | C | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 15 | X | Distance value in mm * 1000 or in steps |
| Example: | SFCP000000000050 | |
| Additional information: | | |
| Accepted in run mode: | Yes | |
| Accepted in configuration mode: | No | |
| Accepted when Ready is low: | Yes | |

| | |
|---|--|
| Status of Ready signal during processing: | No change |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | Max. 4 bytes (optional) |

Read working distance (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Read working distance (GFC) Request string to sensor (ASCII) | | |
|---|-----------------|--|
| Byte no. | Content | Meaning |
| 1 | G | Read working distance (Get Focus) |
| 2 | F | |
| 3 | C | |
| 4 | 1 | Request version |
| 5 | X | Unit
0 - 1/1000 millimeters
4 - steps |
| Example: | GFC10 | |
| Read working distance (GFC) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | G | Read working distance (Get Focus) |
| 2 | F | |
| 3 | C | |
| 4 | P | P: (Pass) Success |
| | F | F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 15 | X | Distance value in mm *1000 or in steps |
| Example: | GFCP00000092500 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Auto shutter speed (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Auto shutter speed (ASH) Request string to sensor (ASCII) | | |
|---|-------------------------|--|
| Byte no. | Content | Meaning |
| 1 | A | Auto shutter speed |
| 2 | S | |
| 3 | H | |
| 4 | 1 | Request version |
| 5 | X | 0: Temporary
1: Permanent |
| Example: | ASH11 | |
| Auto shutter speed (ASH) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | A | Auto shutter speed |
| 2 | S | |
| 3 | H | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 15 | X | Auto Shutter speed value in ms * 1000 |
| 16 - 23 | X | Score in % * 1000 |
| Example: | ASHP0000000178000057500 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Set shutter speed (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Set shutter speed (SSP/SST) Request string to sensor (ASCII) | | |
|--|----------------------|--|
| Byte no. | Content | Meaning |
| 1 | S | Set Shutter Speed |
| 2 | S | |
| 3 | P
T | Permanent
Temporary |
| 4 - 5 | X | Number of digits of the shutter speed value,
e.g. 04 |
| 6 - 9 | X | New shutter speed value in ms * 1000
e.g. 8000 = 8 ms |
| Example: | SSP048000 | |
| Set shutter speed (SSP/SST) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | S | Set Shutter Speed |
| 2 | S | |
| 3 | P
T | Permanent
Temporary |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| Example: | SSPP | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Read shutter speed value (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Read Shutter Speed Value (GSH) Request string to sensor (ASCII) | | |
|--|-----------|--|
| Byte no. | Content | Meaning |
| 1 | G | Read Shutter Speed value (Get Shutter) (from active job) |
| 2 | S | |
| 3 | H | |
| Example: | GSH | |
| Read Shutter Speed Value (GSH) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | G | Read Shutter Speed Value (Get Shutter) |
| 2 | S | |
| 3 | H | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 | X | Shutter speed value, length |
| 6 ... n | X | Shutter speed value in ms * 1000 |
| Example Run Mode: | GSHP41200 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Set gain (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Set gain (SGA) Request string to sensor (ASCII) | | |
|---|-----------|--|
| Byte no. | Content | Meaning |
| 1 | S | Set Gain |
| 2 | G | |
| 3 | A | |
| 4 | X | 0: Temporary
1: Permanent |
| 5 - 9 | X | New gain value (in value * 1000), e.g. 2.0 = 02000 |
| Example: | SGA102000 | |
| Set gain (SGA) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | S | Set Gain |
| 2 | G | |
| 3 | A | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 9 | X | Current gain value * 1000 |
| Example: | SGAP02000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Read gain value (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Read gain value (GGA) Request string to sensor (ASCII) | | |
|---|-----------|--|
| Byte no. | Content | Meaning |
| 1 | G | Read gain value (Get Gain) |
| 2 | G | |
| 3 | A | |
| Example: | GGA | |
| Read gain value (GGA) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | G | Read gain value (Get Gain) |
| 2 | G | |
| 3 | A | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 9 | X | Current gain value (value *1000), e.g. 1.0 = 01000 |
| Example: | GGAP01000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Set parameter (ASCII)

Telegrams: [Availability and supported interfaces \(Page 91\)](#)

Overview telegrams (Page 87)

| Set parameters (SPP/SPT) Request string to sensor (ASCII) | | |
|---|----------------------|--|
| Byte no. | Content | Meaning |
| 1 | S | Set parameters |
| 2 | P | |
| 3 | P
T | P Permanent
T Temporary |
| 4 - 6 | X | Detector number |
| 7 - 9 | X | Command: Parameter number, see below, table Overview detector parameters |
| 10 - 14 | X | Length of value (max. 512 bytes) |
| 15 ... n | X | Value |
| Example: | SPP0010010000560000 | |
| Set parameters (SPP/SPT) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | S | Set parameters |
| 2 | P | |
| 3 | P
T | P Permanent
T Temporary |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |

| | | |
|---|--|--|
| 5 - 8 | X | SI08 - Signed Integer 08
UI08 - Unsigned Integer 08
SI16 - Signed Integer 16
UI16 - Unsigned Integer 16
SI32 - Signed Integer 32
UI32 - Unsigned Integer 32
SI40 - Signed Integer 40
UI40 - Unsigned Integer 40
FLOT - Float
DOBL - Double
STRG - String
BOOL - Boolean
SP08 - Special Signed 8
UDEF - Undefined
IARR - Integer Array
ZERO - Default Zero Parameter |
| Example: | SPPPSTRG | |
| Additional information: | | |
| Accepted in run mode: | Yes | |
| Accepted in configuration mode: | No | |
| Accepted when Ready is low: | Yes | |
| Status of Ready signal during processing: | Low | |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) | |
| End of telegram: | Max. 4 bytes (optional) | |

Overview Detector Parameters (set / read)

| Detector | Function | Value | Multiplier | Length |
|--------------------------------------|---|-------|------------|----------------------------|
| Alignment | | | | |
| Pattern matching
Contour matching | Threshold value Min. | 1 | 1000 | n |
| | Threshold value Max. | 2 | 1000 | n |
| | Result offset
0: "Off"
1: "Image plane (in pixels)"
2: "Align (2D)"
3: "Robot (3D)" | 30 | 1 | n |
| | Result offset Image plane: Pos. X | 31 | 1000 | n |
| | Result offset Image plane: Pos. Y | 32 | 1000 | n |
| | Result offset Image plane: angle | 33 | 1000 | n |
| | Result offset
Align (2D), Robot (3D): Pos. X,
Pos. Y, Pos. Z, Angle X, Angle Y,
Angle Z | 34 | 1000 | 48 (6 * 8 bytes per value) |
| | Calculate Result offset* with transmitted position <ul style="list-style-type: none"> Align (2D): Pos. X, Pos. Y, 0, 0, 0, Angle Z Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z *A valid position for the detector must be available | 35 | 1000 | 48 (6 * 8 bytes per value) |
| Edge detector | Probe 1: Transition
0: Any
1: Dark to light
2: Light to dark | 101 | 1 | n |
| | Probe 2: Transition
0: Any
1: Dark to light
2: Light to dark | 102 | 1 | n |

| Detector | Function | Value | Multiplier | Length |
|---|---|-------|------------|----------------------------|
| | Probe 3: Transition
0: Any
1: Dark to light
2: Light to dark | 103 | 1 | n |
| | Probe 1: Threshold value Min. | 104 | 1000 | n |
| | Probe 2: Threshold value Min. | 105 | 1000 | n |
| | Probe 3: Threshold value Min. | 106 | 1000 | n |
| Detector | | | | |
| Pattern matching
Contour
Contour 3D | Threshold value Min. | 1 | 1000 | n |
| | Threshold value Max. | 2 | 1000 | n |
| | Result offset
0: "Off"
1: "Image plane (in pixels)"
2: "Align (2D)"
3: "Robot (3D)" | 30 | 1 | n |
| | Result offset Image plane: Pos. X | 31 | 1000 | n |
| | Result offset Image plane: Pos. Y | 32 | 1000 | n |
| | Result offset Image plane: angle | 33 | 1000 | n |
| | Result offset
Align (2D), Robot (3D): Pos. X,
Pos. Y, Pos. Z, Angle X, Angle Y,
Angle Z | 34 | 1000 | 48 (6 * 8 bytes per value) |
| | Calculate Result offset* with transmitted position <ul style="list-style-type: none"> • Align (2D): Pos. X, Pos. Y, 0, 0, 0, Angle Z • Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z *A valid position for the detector must be available | 35 | 1000 | 48 (6 * 8 bytes per value) |
| Gray | Threshold value Min. | 1 | 1000 | n |
| | Threshold value Max. | 2 | 1000 | n |
| | Grayscale value Min. | 101 | 1000 | n |

| Detector | Function | Value | Multiplier | Length |
|--------------------------------------|--|-------|------------|--------|
| | Grayscale value Max. | 102 | 1000 | n |
| | Invert grayscale value | 103 | 1 | n |
| Contrast
Brightness | Threshold value Min. | 1 | 1000 | n |
| | Threshold value Max. | 2 | 1000 | n |
| Caliper | Threshold value Distance Min. | 101 | 1000 | n |
| | Threshold value Distance Max. | 102 | 1000 | n |
| | Invert distance threshold value
0: not inverted
1: inverted | 103 | 1 | 1 |
| | Distance mode
0: Minimum
1: Maximum
2: Mean
3: Median
4: Smallest opposite
5: Largest opposite | 104 | 1 | n |
| | Probe 1: Threshold value Min. | 105 | 1000 | n |
| | Probe 2: Threshold value Min. | 106 | 1000 | n |
| | Probe 1: Smoothing | 107 | 1000 | n |
| | Probe 2: Smoothing | 108 | 1000 | n |
| | Probe 1: Transition
0: Any
1: Dark to light
2: Light to dark | 109 | 1 | n |
| | Probe 2: Transition
0: Any
1: Dark to light
2: Light to dark | 110 | 1 | n |
| | Probe 1: Number of search
stripes | 111 | 1 | n |
| Probe 2: Number of search
stripes | 112 | 1 | n | |
| BLOB | Grayscale value Min. | 101 | 1000 | n |
| | Grayscale value Max. | 102 | 1000 | n |
| | Invert grayscale value
0: not inverted
1: inverted | 103 | 1 | 1 |

| Detector | Function | Value | Multiplier | Length |
|----------------------------|---|-------|---------------------------------------|----------------------|
| | Threshold value Number of BLOBs Min. | 120 | 1 | n |
| | Threshold value Number of BLOBs Max. | 121 | 1 | n |
| | Invert number threshold value
0: not inverted
1: inverted | 122 | 1 | 1 |
| | Number of set features (read only) | 123 | 1 | n |
| | Selection of a feature from the list | 124 | 1 | n |
| | Feature threshold value Min. | 125 | 1000 | n |
| | Feature threshold value Max. | 126 | 1000 | n |
| | Invert feature threshold value | 127 | 1 | 1 |
| Barcode
Datacode
OCR | Reference string | 101 | - | n (length of string) |
| Color Value
Color Area | Color space (read only) | 21 | 0 = RGB
1 = HSV
2 = LAB | 3 |
| | Channel selection (read only) | 22 | Bit field one digit per color channel | 4 |
| | Color channel 1: Threshold value Min. | 101 | 1000 | n |
| | Color channel 1: Threshold value Max. | 102 | 1000 | n |
| | Color channel 1: Invert threshold value | 103 | 1 | n |
| | Color channel 2: Threshold value Min. | 104 | 1000 | n |
| | Color channel 2: Threshold value Max. | 105 | 1000 | n |
| | Color channel 2: Invert threshold value | 106 | 1 | n |
| | Color channel 3: Threshold value Min. | 107 | 1000 | n |

| Detector | Function | Value | Multiplier | Length |
|-------------------|---|-------|---------------------------------------|----------------------|
| | Color channel 3: Threshold value Max. | 108 | 1000 | n |
| | Color channel 3: Invert threshold value | 109 | 1 | n |
| Color List | Color space (read only) | 21 | 0 = RGB
1 = HSV
2 = LAB | 3 |
| | Channel selection (read only) | 22 | Bit field one digit per color channel | 4 |
| | Color distance threshold value | 101 | 1000 | n |
| | Set color distance threshold value active | 102 | 1 | n |
| | Number of colors in list | 103 | 1 | n |
| | Selection of a color from the list | 104 | 1 | n |
| | Color value of the selected color (color channel 1, color channel 2, color channel 3, color channel 4 [constantly 0]) | 105 | 1000 | 32 |
| Busbar Wafer | Threshold value Min. | 1 | 1000 | n |
| | Threshold value Max. | 2 | 1000 | n |
| Result processing | Name of the active expression | 122 | - | n (length of string) |
| | Current expression | 124 | - | n (length of string) |

Read parameter (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Read parameter (GPA) Request string to sensor (ASCII) | | |
|--|-----------|--|
| Byte no. | Content | Meaning |
| 1 | G | Read parameter (Get Parameter) |
| 2 | P | |
| 3 | A | |
| 4 - 6 | X | Detector number
e.g. 001 |
| 7 - 9 | X | Command: Parameter number, see table Overview detector parameters |
| Example: | GPA001001 | |
| Read parameter (GPA) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | G | Read parameter (Get Parameter) |
| 2 | P | |
| 3 | A | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 8 | X | SI08 - Signed Integer 08
UI08 - Unsigned Integer 08
SI16 - Signed Integer 16
UI16 - Unsigned Integer 16
SI32 - Signed Integer 32
UI32 - Unsigned Integer 32
SI40 - Signed Integer 40
UI40 - Unsigned Integer 40
FLOT - Float
DOBL - Double
STRG - String
BOOL - Boolean
SP08 - Special Signed 8
UDEF - Undefined
IARR - Integer Array
ZERO - Default Zero Parameter |
| 9 - 13 | X | Length of value (n)
e.g. 00005 |

| 14 ... n | X | Value |
|---|--|-------|
| Example: | GPAPSTRG00005Test1 | |
| Additional information: | | |
| Accepted in run mode: | Yes | |
| Accepted in configuration mode: | No | |
| Accepted when Ready is low: | Yes | |
| Status of Ready signal during processing: | No change | |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) | |
| End of telegram: | Max. 4 bytes (optional) | |

Set search range (ROI) (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Set ROI (SRP/SRT) Request string to sensor (ASCII) | | |
|---|----------------------|---|
| Byte no. | Content | Meaning |
| 1 | S | Set search range (Set ROI) |
| 2 | R | |
| 3 | P
T | P = Permanent
T = Temporary |
| 4 - 11 | X | ROI Info Length in bytes, from byte 5 to end
39 Byte: circle
55 bytes: rectangle, ellipse, free shape |
| 12 - 14 | X | Detector no.
e.g. 001 |
| 15 - 16 | X | ROI Index
00: for yellow search range
01: for red teach range
02: Position control |
| 17 - 18 | X | ROI shape
01: Circle
02: Rectangle
03: Ellipse
04: Free shape |
| 19 - 26 | X | Center X (value in pixels * 1000), e.g. 160 pixels = 00160000 |
| 27 - 34 | X | Center Y (value in pixels * 1000), e.g. 120 pixels = 00120000 |
| 35 - 42 | X | Half width / X-radius (value in pixels * 1000), e.g. 80 pixels = 00080000 |
| 43 - 50 | X | Half height (not for circle) (value in pixels * 1000), e.g. 40 pixels = 00040000 |
| 51 - 58 | X | Angle (not for circle) (value in ° * 1000), e.g. 180° = 00180000 |

| | | |
|--|---|--------------------------------------|
| Example: | SRP000000550010002001600000012000000
0800000004000000180000
Length=55, detector=1, yellow search range, rectangle, center X=160,
center Y=120, half width= 80, half height=40, orientation=180 | |
| Set ROI (SRP/SRT) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | S | Set search range (Set ROI) |
| 2 | R | |
| 3 | P
T | Permanent
Temporary |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| Example: | SRPP | |
| Additional information: | | |
| Accepted in run mode: | Yes | |
| Accepted in configuration mode: | No | |
| Accepted when Ready is low: | Yes | |
| Status of Ready signal during processing: | Low | |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) | |
| End of telegram: | Max. 4 bytes (optional) | |
| Parameter: | The parameters are given in the coordinate system of the Alignment and not in the coordinate system of the image. | |

Read search range (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)


| Read search range (GRI) Request string to sensor (ASCII) | | |
|--|----------|---|
| Byte no. | Content | Meaning |
| 1 | G | Read search range (Get ROI) |
| 2 | R | |
| 3 | I | |
| 4 - 6 | X | Detector no.
e.g. 001 |
| 7 - 8 | X | ROI Index
00: for yellow search range
01: for red teach range
02: Position control |
| Example: | GRI00100 | |
| Read search range (GRI) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | G | Read search range (Get ROI) |
| 2 | R | |
| 3 | I | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 12 | X | ROI Info Length in bytes, from byte 5 to end
39 Byte: circle
55 bytes: rectangle, ellipse, free shape |
| 13 - 15 | X | Detector no.
e.g. 001 |
| 16 - 17 | X | ROI Index
00: for yellow search range
01: for red teach range
02: Position control |

| | | |
|---|--|--|
| 18 - 19 | X | ROI shape
01: Circle
02: Rectangle
03: Ellipse
04: Free shape |
| 20 - 27 | X | Center X (value in pixels * 1000) |
| 28 - 35 | X | Center Y (value in pixels * 1000) |
| 36 - 43 | X | Half width / X-radius (value in pixels * 1000) |
| 44 - 51 | X | Half height (not for circle) (value in pixels * 1000), e.g. 40 pixels = 00040000 |
| 52 - 59 | X | Angle (not for circle) (value in ° * 1000), e.g. 180° = 00180000 |
| Example: | GRIP000000550010002001600000012000000
0800000004000000090000
(Length= 55, detector 1, search range, rectangle, center X= 160, center Y= 120, half width= 80, half height= 40, angle= 90) | |
| Additional information: | | |
| Accepted in run mode: | Yes | |
| Accepted in configuration mode: | No | |
| Accepted when Ready is low: | Yes | |
| Status of Ready signal during processing: | Low | |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) | |
| End of telegram: | Max. 4 bytes (optional) | |

Read job list (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)


| Read job list (GJL) Request string to sensor (ASCII) | | |
|---|---|--|
| Byte no. | Content | Meaning |
| 1 | G | Read job list (Get Job List) |
| 2 | J | |
| 3 | L | |
| Example: | GJL | |
| Read job list (GJL) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | G | Read job list (Get Job List) |
| 2 | J | |
| 3 | L | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Response version |
| 8 - 10 | X | Number of jobs |
| 11 - 13 | X | Active job number |
|  | NOTE:
The following byte sequence is repeated for each job from 1 to "Number of jobs".
The byte numbers shift accordingly. | |
| 14 - 16 | X | Number of characters for the job name. This can be used to specify a unique name for job n. |
| 17 ... n | X | From this position, the name for job n follows in the specified length. |
| n+1 ... n + 3 | X | Number of subsequent bytes. A description for job n can be specified. |
| n + 4 ... m | X | From this position, the description for job n follows in the specified length. |
| m + 1 ... m + 3 | X | Number of subsequent bytes. This can be used to specify a unique name for the author of job n. |
| m + 4 ... k | X | From this position, the name for the author of job n follows in the specified length. |
| k + 1 ... k + 19 | X | Date of creation of Job n (19 bytes) |

| | | |
|---|--|--|
| k + 20 ... k + 39 | X | Date of last modification of job n (19 bytes) |
| Example: | GJLP001001001007testjob010DefaultJob
004Test2014112720141128 | |
| Additional information: | | |
| Accepted in run mode: | Yes | |
| Accepted in configuration mode: | No | |
| Accepted when Ready is low: | Yes | |
| Status of Ready signal during processing: | No change | |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) | |
| End of telegram: | Max. 4 bytes (optional) | |

Read detector list (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Read detector list (GDL) Request string to sensor (ASCII) | | |
|--|---|---|
| Byte no. | Content | Meaning |
| 1 | G | Get Detector List |
| 2 | D | |
| 3 | L | |
| Example: | GDL | |
| Read detector list (GDL) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | G | Get Detector List |
| 2 | D | |
| 3 | L | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Job number of the current job |
| 8 - 10 | X | Number of detectors in the current job |
| |  | NOTE:
The following byte sequence is repeated for each detector in the job. The byte numbers shift accordingly. |
| 11 - 13 | X | Number of subsequent bytes. This allows a unique name for the detector n to be specified. |
| 14 ... n | X | From this position, the name for detector n follows, in the given length. |

| | | |
|---|--|--|
| n + 1 ... n+5 | X | 001 - Pattern matching
004 - Contour
005 - Gray
006 - Contrast
007 - Brightness
010 - Wafer
011 - OCR
013 - Datacode
014 - Barcode
017 - Busbar
018 - Color Value
019 - Color Area
020 - Color List
021 - Caliper
022 - BLOB |
| Example: | GDLP001001012testdetector00005 | |
| Additional information: | | |
| Accepted in run mode: | Yes | |
| Accepted in configuration mode: | No | |
| Accepted when Ready is low: | Yes | |
| Status of Ready signal during processing: | No change | |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) | |
| End of telegram: | Max. 4 bytes (optional) | |

Teach detector (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Teach detector (TED) Request string to sensor (ASCII) | | |
|---|----------|--|
| Byte no. | Content | Meaning |
| 1 | T | Teach detector |
| 2 | E | |
| 3 | D | |
| 4 - 6 | X | 0 = Alignment
≥ 1 Detectors |
| 7 | X | 0: Temporary
1: Permanent |
| 8 | X | 0: No trigger, teach-in with next image acquisition
1: Trigger is executed for teach-in |
| Example: | TED00111 | |
| Teach detector (TED) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | T | Teach detector |
| 2 | E | |
| 3 | D | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| Example: | TEDP | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Set trigger delay (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Set trigger delay (STD) Request string to sensor (ASCII) | | |
|---|---------------|--|
| Byte no. | Content | Meaning |
| 1 | S | Set Trigger Delay |
| 2 | T | |
| 3 | D | |
| 4 | 1 | Request version |
| 5 | X | 0: Temporary
1: Permanent |
| 6 - 13 | X | Trigger delay
in msec (max. 3000 msec)
in encoder steps (max. 65535 steps) |
| Example: | STD1100001000 | |
| Set trigger delay (STD) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | S | Set Trigger Delay |
| 2 | T | |
| 3 | D | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| Example: | STDP000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Read trigger delay (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Read trigger delay (GTD) Request string to sensor (ASCII) | | |
|--|-----------------|--|
| Byte no. | Content | Meaning |
| 1 | G | Read Trigger Delay (Get Trigger Delay) |
| 2 | T | |
| 3 | D | |
| 4 | 1 | Request version |
| Example: | GTD1 | |
| Get trigger delay (GTD) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | G | Read Trigger Delay (Get Trigger Delay) |
| 2 | T | |
| 3 | D | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error code |
| 8 - 15 | X | Trigger delay
in msec (max. 3000 msec)
in encoder steps (max. 65535 steps) |
| Example: | GTDP00000001000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Save job permanently (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Save Job Permanently (SJP) Request string to sensor (ASCII) | | |
|--|--|--|
| Byte no. | Content | Meaning |
| 1 | S | Save Job Permanently (Store Job Permanently) |
| 2 | J | |
| 3 | P | |
| Example: | SJP | |
| Save Job Permanently (SJP) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | S | Save Job Permanently (Store Job Permanently) |
| 2 | J | |
| 3 | P | |
| 4 | P | P: (Pass) Success |
| | F | F: (Fail) Error |
| Example: | SJPP | |
| Additional information: | | |
| Accepted in run mode: | Yes | |
| Accepted in configuration mode: | No | |
| Accepted when Ready is low: | Yes | |
| Status of Ready signal during processing: | Low | |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) | |
| End of telegram: | Max. 4 bytes (optional) | |

11.4.4 Calibration

Calibration: Initialization (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Initialize (CCD) Request string to sensor (ASCII) | | |
|--|----------|--|
| Byte no. | Content | Meaning |
| 1 | C | Initialize (Calibration: Clear Data) |
| 2 | C | |
| 3 | D | |
| Example: | CCD | |
| Initialize (CCD) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | C | Initialize (Calibration: Clear Data) |
| 2 | C | |
| 3 | D | |
| 4 | P | P: (Pass) Success |
| | F | F: (Fail) Error |
| Example: | CCDP | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Calibration: Add world point (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Add world point (CAW) Request string to sensor (ASCII) | | |
|---|--|---|
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Add World Point |
| 2 | A | |
| 3 | W | |
| 4 | 1 | Request version |
| 5 | X | 1: Calibration plate (Robotics) Fiducials only
4: Point pair list (Robotics) World point and image point |
| 6 - 10 | 0 | Constant (5 bytes) |
| 11 - 18 | X | World X (in user unit * 1000) |
| 19 - 26 | X | World Y (in user unit * 1000) |
| 27 - 34 | 0 | Constant (8 bytes) |
| Example: | CAW100001001000000020000000000000 (World X = 100 mm; World Y = 200mm) | |
| Calibration: Add world point (CAW) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Add World Point |
| 2 | A | |
| 3 | W | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 12 | X | Current number of points |
| 13 - 20 | X | Image point X |
| 21 - 28 | X | Image point Y |
| Example: | CAWP000000010028800000566000 (Reference point 1; Image X = 288; Image Y = 566) | |
| Additional information: | | |
| Accepted in run mode: | | Yes |

| | |
|---|--|
| Accepted in configuration mode: | No |
| Accepted when Ready is low: | Yes |
| Status of Ready signal during processing: | No change |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | Max. 4 bytes (optional) |

Note: For the CAW request, the overall job result must be positive.

Calibration: Point pair list (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration by point pair list (CCL) Request string to sensor (ASCII) | | |
|--|---|--|
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Point pair list |
| 2 | C | |
| 3 | L | |
| 4 | X | 0: Temporary
1: Permanent |
| Example: | CCL1 | |
| Calibration: Point pair list (CCL) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Point pair list |
| 2 | C | |
| 3 | L | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 9 | X | Current highest point pair index |
| 10 - 17 | X | Deviation calibration, RMSE |
| 18 - 25 | X | Deviation calibration, mean |
| 26 - 33 | X | Deviation calibration, max. |
| 34 - 41 | X | Deviation calibration, min. |
| Example: | CCLP0001012345678123456781234567812345678 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Calibration: Validate point pair list (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Validate point pair list (CVL) Request string to sensor (ASCII) | | |
|--|---|--|
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Validate Point Pair List |
| 2 | V | |
| 3 | L | |
| Example: | CVL | |
| Calibration: Validate point pair list (CVL) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Validate Point Pair List |
| 2 | V | |
| 3 | L | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 9 | X | Current highest point pair index |
| 10 - 17 | X | Deviation calibration, RMSE |
| 18 - 25 | X | Deviation calibration, mean |
| 26 - 33 | X | Deviation calibration, max. |
| 34 - 41 | X | Deviation calibration, min. |
| Example: | CVLP0001012345678123456781234567812345678 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Calibration: Calibration plate (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Calibration Plate (CCP) Request string to sensor (ASCII) | | |
|---|----------|---|
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Calibration plate |
| 2 | C | |
| 3 | P | |
| 4 | 1 | Request version |
| 5 | X | 0: Temporary
1: Permanent |
| 6 | X | 0: No fiducials are used. Origin of Measuring coordinate system identical to origin of Calibration Plate Coordinate System.
1: No fiducials are used. Measuring coordinate system identical with Camera coordinate system.
2: Uses world system, fiducial job
3: Uses world system, fiducial command
CAW |
| 7 | X | 0: Calibration internal and external sensor parameters
1: Validation of calibration
2: Calibration internal sensor parameters
5: Calibration transformation Measuring coordinate system |
| Example: | CCP1110 | |
| Calibration: Calibration Plate (CCP) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Calibration plate |
| 2 | C | |
| 3 | P | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 12 | X | Number of currently detected calibration points |

| | | |
|---|--|------------------------------------|
| 13 - 20 | X | Deviation calibration, RMSE |
| 21 - 28 | X | Deviation calibration, mean |
| 29 - 36 | X | Deviation calibration, max. |
| 37 - 44 | X | Deviation calibration, min. |
| 45 - 52 | X | CPF_MF X (in user unit * 1000) |
| 53 - 60 | X | CPF_MF Y (in user unit * 1000) |
| 61 - 68 | 0 | CPF_MF Z (in user unit * 1000) |
| 69 - 76 | 0 | CPF_MF Angle X (in degrees * 1000) |
| 77 - 84 | 0 | CPF_MF Angle Y (in degrees * 1000) |
| 85 - 92 | X | CPF_MF Angle Z (in degrees * 1000) |
| 93 - 100 | X | Deviation fiducials, mean |
| 101 - 108 | X | Deviation fiducials, max. |
| 109 - 116 | X | Deviation fiducials, min. |
| Example: | CCPP00000012000010010000200200003003000040040
00050050000600600007007000080080000900900001001 | |
| Additional information: | | |
| Accepted in run mode: | Yes | |
| Accepted in configuration mode: | No | |
| Accepted when Ready is low: | Yes | |
| Status of Ready signal during processing: | No change | |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) | |
| End of telegram: | Max. 4 bytes (optional) | |

Calibration: Set fiducial (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Set fiducial (CSF) Request string to sensor (ASCII) | | |
|---|--|---------------------------------------|
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Set fiducial |
| 2 | S | |
| 3 | F | |
| 4 | 1 | Request version |
| 5 | X | 0: Temporary
1: Permanent |
| Example: | CSF11 | |
| Calibration: Set fiducial (CSF) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Set fiducial |
| 2 | S | |
| 3 | F | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 15 | X | X value (in user unit * 1000) |
| 16 - 23 | X | Y value (in user unit * 1000) |
| 24 - 31 | X | Z value (in user unit * 1000) |
| 32 - 39 | X | Angle X value (in degrees * 1000) |
| 40 - 47 | X | Angle Y value (in degrees * 1000) |
| 48 - 55 | X | Angle Z value (in degrees * 1000) |
| 56 - 63 | X | Deviation fiducials, mean |
| 64 - 71 | X | Deviation fiducials, max. |
| 72 - 79 | X | Deviation fiducials, min. |
| Example: | CSFP000000010010000200200003003000040040
00050050000600600001001000020200003003 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |

| | |
|---|--|
| Accepted when Ready is low: | Yes |
| Status of Ready signal during processing: | No change |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | Max. 4 bytes (optional) |

Calibration: Add image (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Add image (CAI) Request string to sensor (ASCII) | | |
|---|----------|---|
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Add Image |
| 2 | A | |
| 3 | I | |
| 4 | 1 | Request version |
| 5 | X | Mode
1: Multi-image calibration
2: Hand-Eye calibration (Robotics)
3: Base-Eye calibration (Robotics) |
| 6-8 | 0 | Append at the end of the list (5 bytes) |
| 9 | X | Define Measurement plane
0: Do not use image to define Measurement plane
1: Use image to define Measurement plane |
| 10-11 | X | "Robot: Order of rotation"
00: Use rotation order specified in job
01: Yaw-Pitch-Roll (e.g. Stäubli)
02: Roll-Pitch-Yaw (e.g. Kuka, Fanuc, Hanwha, ABB**, UR**)
** when using the corresponding conversion function |
| 12-19 | X | Pose_TCP Pos. X
(in user unit * 1000) |
| 20-27 | X | Pose_TCP Pos. Y
(in user unit * 1000) |
| 28-35 | X | Pose_TCP Pos. Z
(in user unit * 1000) |
| 36-43 | X | Pose_TCP Angle X
(in degrees * 1000) |
| 44-51 | X | Pose_TCP Angle Y
(in degrees * 1000) |
| 52-59 | X | Pose_TCP Angle Z
(in degrees * 1000) |

| Example: | CAI11 001 1 02
000040040000500500006006000070070000800800009009 | |
|---|--|---------------------------------------|
| Calibration: Add image (CAI) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Add Image |
| 2 | A | |
| 3 | I | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5-7 | X | Error codes (Page 94) |
| 8-10 | X | Current number of images in list |
| 11-15 | X | Total number of detected points |
| Example: | CAIP 000 001 00021 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |

Calibration: Multi-image (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Multi-image (CMP) Request string to sensor (ASCII) | | |
|---|----------|--|
| Byte no. | Content | Meaning |
| 1 | C | Calibration Calibrate Multi-Image Plate |
| 2 | M | |
| 3 | P | |
| 4 | 1 | Request version |
| 5 | X | 0: Temporary
1: Permanent |
| 6 | X | Origin of the world coordinate system:
0: World coordinate system identical with the Calibration Plate Coordinate System (center of the plate).
1: Origin of World coordinate system so that it is identical to origin of Image Coordinate System (upper left pixel).
2: Use World coordinate system of fiducials, as specified in the job file.
3: Use World coordinate system of fiducials as set in request CAW. |
| 7 | X | Mode
0: Calibration (internal and external parameters)
1: Validieren (vorhandene Kalibrierung verwenden; mindestens ein Kalibrierpunkt wird hinzugefügt. Über Rückprojektion kann zurückgeschlossen werden, ob der Punkt zur aktuellen Kalibrierung passt, oder verschoben ist)
2: Calibration (internal parameters only)
3: Calibration (external parameters only using new internal parameters)
4: Calibration (external parameters only)
5: Calibrate Measurement plane only (CPF_MF) |
| Example: | CMP1105 | |

| Calibration: Multi-image (CMP) Response string from sensor (ASCII) | | |
|---|--|---|
| Byte no. | Content | Meaning |
| 1 | C | Calibration Calibrate Multi-image |
| 2 | M | |
| 3 | P | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5-7 | X | Error codes (Page 94) |
| 8-10 | X | Field of view coverage (%) |
| 11-15 | X | Total number of detected points |
| 16-18 | X | Number of images used |
| 19-21 | X | Number of invalid images |
| 22 | X | Sufficient tilt between calibration plate poses
0: not sufficient
1: sufficient |
| 23-30 | X | Deviation calibration, RMSE [px] |
| 31-38 | X | Deviation calibration, max. [px] |
| 39-46 | X | Deviation fiducials, RMSE (in user unit * 1000) |
| 47-54 | X | Deviation fiducials, max. (in user unit * 1000) |
| Example: | CMPP 000 089 00312 011 002 0
00001001000020020000300300004004 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Calibration: Robotics multi-image (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Robotics multi-image (CRP) Request string to sensor (ASCII) | | |
|--|----------|---|
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Robotics multi-image (Calibrate Robotics Plate) |
| 2 | R | |
| 3 | P | |
| 4 | 1 | Request version |
| 5 | x | 0: Temporary
1: Permanent |
| 6 | X | Origin of the world coordinate system:
4: Set world frame to User Robot Frame |
| 7 | X | Mode
0: Calibration (internal and external parameters)
1: Validieren (vorhandene Kalibrierung verwenden; mindestens ein Kalibrierpunkt wird hinzugefügt. Über Rückprojektion kann zurückgeschlossen werden, ob der Punkt zur aktuellen Kalibrierung passt, oder verschoben ist)
2: Calibration (internal parameters only)
4: Calibration (external parameters only)
5: Calibrate Measurement plane only (CPF_MF)
6: Calibrate Hand-Eye/Base-Eye |
| Example: | CRP1140 | |
| Calibration: Robotics multi-image (CRP) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Robotics multi-image (Calibrate Robotics Plate) |
| 2 | R | |
| 3 | P | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5-7 | X | Error codes (Page 94) |
| 8-10 | X | Field of view coverage (%) |

| | | |
|---|---|--|
| 11-15 | X | Total number of detected points |
| 16-18 | X | Number of images used |
| 19-21 | X | Number of invalid images |
| 22-29 | X | Deviation calibration, RMSE [px] |
| 30-37 | X | Deviation calibration, max. [px] |
| 38-45 | X | Deviations calibration plate pose Translation RMSE (in user unit * 1000) |
| 46-53 | X | Deviations calibration plate pose Translation Max. (in user unit * 1000) |
| 54-61 | X | Deviations calibration plate pose Rotation RMSE (in degrees * 1000) |
| 62-69 | X | Deviations calibration plate pose Rotation Max. (in degrees * 1000) |
| Example: | CRPP 000 092 01349 012 004 0000100100002002
00003003000040040000500500006006 | |
| Additional information: | | |
| Accepted in run mode: | Yes | |
| Accepted in configuration mode: | No | |
| Accepted when Ready is low: | Yes | |
| Status of Ready signal during processing: | No change | |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) | |
| End of telegram: | Max. 4 bytes (optional) | |

Calibration: Copy Calibration (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Copy calibration (CCC) Request string to sensor (ASCII) | | |
|--|------------|---|
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Copy calibration |
| 2 | C | |
| 3 | C | |
| 4 | 1 | Request version |
| 5 | 1 | Constant |
| 6 - 8 | X | Destination
0 : Copy to all jobs
>0: Copy to specified job |
| 9 | X | 0: Always copy when the calibration is active.
1: Only copy if the calibration method is the same. |
| Example: | CCC110021 | |
| Calibration: Copy calibration (CCC) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Copy calibration |
| 2 | C | |
| 3 | C | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 10 | X | Job number of the job where the error occurred
00: Successful
>0 - Job number of the job where the error first occurred |
| Example: | CCCP000000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |

| | |
|---|--|
| Status of Ready signal during processing: | No change |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | Max. 4 bytes (optional) |

Calibration: Set parameter (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Set parameter (CSP) Request string to sensor (ASCII) | | |
|--|-------------------|--|
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Set Parameter |
| 2 | S | |
| 3 | P | |
| 4 | 1 | Request version |
| 5 | X | 0: Temporary
1: Permanent |
| 6 - 8 | X | Parameter number, see table Calibration parameters CSP and CGP |
| 9 - 16 | X | Length of value |
| 17 ... n | X | Value for selected parameter, see table Calibration parameters CSP and CGP |
| Example: | CSP11002000000019 | |
| Calibration: Set parameter (CSP) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Set Parameter |
| 2 | S | |
| 3 | P | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| Example: | CSPP000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Calibration parameters: see table [Calibration parameters for telegrams CSP and CGP](#)

Calibration: Read parameter (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Read parameter (CGP) Request string to sensor (ASCII) | | |
|---|---------------------|--|
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Read Parameter |
| 2 | G | |
| 3 | P | |
| 4 | 1 | Request version |
| 5 - 7 | X | Parameter number, see calibration parameters CSP and CGP |
| Example: | CGP1001 | |
| Calibration: Read parameter (CGP) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | C | Calibration: Read Parameter |
| 2 | G | |
| 3 | P | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 10 | X | Parameter number, see calibration parameters CSP and CGP |
| 11 - 18 | X | Length of the following data |
| 19 ... n | X | Parameter values, depending on the selected parameter |
| Example: | CGPP000001000000011 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Calibration parameters for telegrams CSP and CGP

| Parameter description | Parameter number | Value | length | Calibration status after CSP |
|---|------------------|---|-------------------------------|------------------------------|
| Status calibration | 001 | 0: Invalid
1: Valid | 1 byte | —* |
| Calibration method | 002 | 0: None
2: Point pair list (Robotics)
3: Calibration plate (Measurement)
4: Calibration plate (Robotics)
5: Hand-Eye calibration (Robotics)
6: Base-Eye calibration (Robotics) | 1 byte | invalid |
| Unit (user unit) | 004 | 0: Millimeter [mm]
1: Centimeter [cm]
2: Meter [m]
3: Inch ["]
4: Arbitrary unit [au] | 1 byte | no change |
| Internal parameters | 010 | Focal length (in mm *1000)
Kappa (*1000)
Pixel pitch X (in μm * 1000)
Pixel pitch Y (in μm * 1000)
Coordinate origin X (in pixels * 1000)
Coordinate origin Y (in pixels * 1000)
Image size X (number of pixels)
Image size Y (number of pixels) | 64
(8 * 8 bytes per value) | —* |
| Reference Camera- to Measuring coordinate system (CF_MF) | 011 | Translation X, Y, Z (in user unit * 1000)
Angle X, Y, Z (in degrees * 1000) | 48
(6 * 8 bytes per value) | —* |
| Reference Camera- to Calibration Plate Coordinate System (CF_CPF) | 012 | Translation X, Y, Z (in user unit * 1000)
Angle X, Y, Z (in degrees * 1000) | 48
(6 * 8 bytes per value) | —* |

| Parameter description | Parameter number | Value | length | Calibration status after CSP |
|--|------------------|--|----------------------------|--------------------------------|
| Reference Robot- to Camera coordinate system (RF_CF) | 013 | Translation X, Y, Z (in user unit * 1000)
Angle X, Y, Z (in degrees * 1000) | 48 (6 * 8 bytes per value) | —* |
| Reference Calibration plate- to Measuring coordinate system (CPF_MF) | 014 | Translation X, Y, Z (in user unit * 1000)
Angle X, Y, Z (in degrees * 1000) | 48 (6 * 8 bytes per value) | —* |
| Reference Robot- to Measuring coordinate system (RF_MF) | 015 | Translation X, Y, Z (in user unit * 1000)
Angle X, Y, Z (in degrees * 1000) | 48 (6 * 8 bytes per value) | —* |
| Reference TCP- to Camera coordinate system (TCP_CF) | 016 | Translation X, Y, Z (in user unit * 1000)
Angle X, Y, Z (in degrees * 1000) | 48 (6 * 8 bytes per value) | —* |
| Reference robot- to TCP coordinate system (RF_TCP) | 017 | Translation X, Y, Z (in user unit * 1000)
Angle X, Y, Z (in degrees * 1000) | 48 (6 * 8 bytes per value) | no change |
| Z-shift Measurement plane | 021 | Value (in user unit * 1000) | 8 Byte | no change |
| Focal length in [mm] | 022 | [mm * 1000] | 8 Byte | invalid (CSP for C-Mount only) |
| Calibration plate type | 023 | Character string with name of the description file | n | invalid |
| Fiducial 1 | 024 | Translation X, Y, Z (in user unit * 1000) | 24 (3 * 8 bytes per value) | invalid |
| Fiducial 2 | 025 | | | |
| Fiducial 3 | 026 | | | |
| Fiducial 4 | 027 | | | |

| Parameter description | Parameter number | Value | length | Calibration status after CSP |
|--|------------------|---|--|------------------------------|
| Number of existing calibration plate types | 037 | Request - Selection of type:
0: All
1: Measurement
2: Robotics
Response:
Number of plates | Request: 1
Response:
5 | —* |
| Available calibration plate types (file names) | 038 | Request - Selection of type:
0: All
1: Measurement
2: Robotics
Request - Index:
0: All file names
>0: Index selection
Response:
File names of Calibration plates | Request: 1 /
5
Response:
n (String) | —* |
| Robot: Order of rotation | 039 | "Robot: Order of rotation"
00: Use rotation order specified in job
01: Yaw-Pitch-Roll (e.g. Stäubli)
02: Roll-Pitch-Yaw (e.g. Kuka, Fanuc, Hanwha, ABB**, UR**)
** when using the corresponding conversion function | 1 | invalid |
| Average sensor resolution | 041 | Value (in user unit/pixel * 1000) | 8 bytes | —* |

* CSP not possible (parameter is read-only and cannot be set).

11.4.5 Visualization

Get image (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Get image (GIM) Request string to sensor (ASCII) | | |
|---|----------|--|
| Byte no. | Content | Meaning |
| 1 | G | Get Image |
| 2 | I | |
| 3 | M | |
| 4 | X | 0: Last image
1: Last bad image
2: Last good image |
| Example: | GIM1 | |
| Get image (GIM) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | G | Get Image |
| 2 | I | |
| 3 | M | |
| 4 | P | P: (Pass) Success |
| | F | F: (Fail) Error |
| 5 | X | Error codes (Page 94) |
| 6 | X | Image type
0: Grayscale
3: Bayer-Pattern_BG
When converting the color image from Bayer into RGB, the appropriate image type must be considered.
Pre-processing filters of the category "Arrangement" have an influence on the Bayer type.
Bayer Pattern begins with blue - green. |
| 7 | X | Image result
1: Good image
0: Failed image |

| | | |
|---|--|--|
| 8 - 11 | X | Number of rows
e.g. 0480 / 0200 |
| 12 - 15 | X | Number of columns
e.g. 0640 / 0320 |
| 16 - 19 | X | End of the message string if specified. Otherwise start image data from Byte no. 16. |
| 20 ... n | X | Binary image data (rows * columns) |
| Example: | GIMP0004800640... | |
| Additional information: | | |
| Accepted in run mode: | Yes | |
| Accepted in configuration mode: | No | |
| Accepted when Ready is low: | Yes | |
| Status of Ready signal during processing: | Low | |
| Supported interfaces: | Telegrams: Availability and supported interfaces (Page 91) | |
| End of telegram: | Max. 4 bytes (optional) | |

11.4.6 Service (available only on port 1998 and in ASCII format)

Update visualization data (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Update visualization data (UVR) Request string to sensor (ASCII) | | |
|--|----------|--|
| Byte no. | Content | Meaning |
| 1 | U | Update visualization data (Update Visualization Results) |
| 2 | V | |
| 3 | R | |
| 4 | 1 | Request version |
| 5 | X | Image:
0: No image is created
1: Grayscale / RGB image without filter, BMP format
2: Grayscale image / Bayer pattern without filter, BMP format
3: Grayscale / RGB image with filter, BMP format
4: Grayscale image / Bayer pattern with filter, BMP format
5: Grayscale / RGB image without filter, JPEG format (low compression)
6: Grayscale / RGB image with filter, JPEG format (low compression)
7: Grayscale / RGB image without filter, JPEG format (compression high)
8: Grayscale / RGB image with filter, JPEG format (compression high) |
| 6 | X | Result XML:
0: Result file is not created
1: Result file is created |
| 7 | X | Statistic XML:
0: Statistics file is not created
1: Statistics file is created |

| | | |
|--|-------------|---|
| 8 | X | Image type:
0: Last image (Any)
1: Last fail image (Fail)
2: Last pass image (Pass)
3: Next image (Any)
4: Next fail image (Fail)
5: Next pass image (Pass) |
| 9 - 11 | X | Directory number (constant)
001: visu001 |
| Example: | UVR11110001 | |
| Update visualization data (UVR) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | U | Update visualization data (Update Visualization Results) |
| 2 | V | |
| 3 | R | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 | X | Data available:
0: New data available when ready.txt is written
1: No new data available. |
| 9 - 11 | X | Directory number (constant)
001: visu001 |
| Example: | UVRP0000001 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Status of Ready signal during processing: | | No change |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

The created files are available for download in the directory /tmp/[Directory number]:

- image.bmp
- overlay.xml

With the file "overlay.xml", all relevant information for creating the overlay can be obtained. The file is created in XML format. The most important elements are described in the table below

| Name | | Value | Description |
|----------|-------------|--|---|
| detector | type | pattern_matching
contour
contrast
brightness
gray
caliper
blob
ocr
datacode
barcode | Detector Type |
| | number | Integer | Position in detector list |
| | name | String | Name of the detector defined in the configuration |
| roi | purpose | Search
teach
position_control
result | Type of overlay element. The different types have different colors. |
| | shape | rectangle
rectangle_mask
ellipse | Shape of the overlay element |
| center | x | Float | Center position in X (pixels) |
| | y | Float | Center position in Y (pixels) |
| size | half_width | Float | Half width of overlay element |
| | half_height | Float | Half height of overlay element |
| angle | angle | Float | Angle of overlay element (degrees) |
| number | value | Float | Number of element types in this detector |
| line | x1 | Float | Start point X line 1 (pixels) |
| | y1 | Float | Start point Y line 1 (pixels) |
| | x2 | Float | Start point X line 2 (pixels) |
| | y2 | Float | Start point Y line 2 (pixels) |

Depending on the detector type (detector → type), there are different elements that can be displayed. The following table indicates which element can be displayed on which detector.

| Detector | Search | teach | position_control | result |
|------------------|--------|-------|------------------|--------|
| Pattern matching | Yes | Yes | Yes | 1 |
| Contour | Yes | Yes | Yes | 200 |
| Contrast | Yes | No | No | 0 |
| Brightness | Yes | No | No | 0 |
| Gray | Yes | No | No | 0 |
| Caliper | Yes | No | No | 0 |
| BLOB | Yes | No | No | 1000 |
| OCR | Yes | No | No | 1 |
| Datacode | Yes | No | No | 5 |
| Barcode | Yes | No | No | 5 |

Read sensor identity (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Read sensor identity (GSI) Request string to sensor (ASCII) | | |
|---|---------------------------------------|--|
| Byte no. | Content | Meaning |
| 1 | G | Read sensor identity (Get Sensor Identity) |
| 2 | S | |
| 3 | I | |
| 4 | 1 | Request version |
| Example: | GSI1 | |
| Read sensor identity (GSI) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | G | Read sensor identity (Get Sensor Identity) |
| 2 | S | |
| 3 | I | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 10 | X | Length of the following data |
| 11 ... n | X | Version of the firmware as well as information about the hardware. Areas are clearly separated by a semicolon. |
| Example: | GSIP0000262.0.0.3; V20-RO-P3-R-M-M2-L | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

Update firmware (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Update firmware (UFW) Request string to sensor (ASCII) | | |
|---|----------|--|
| Byte no. | Content | Meaning |
| 1 | U | Update firmware |
| 2 | F | |
| 3 | W | |
| 4 | 1 | Request version |
| Example: | UFW1 | |
| Update firmware (UFW) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | U | Update firmware |
| 2 | F | |
| 3 | W | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| Example: | UFWP000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

After the command is sent, the /tmp/ on the VISOR® vision sensor will be checked for a valid firmware file. The name must correspond to the typical name allocation (e.g. as after the download from the SensoPart homepage). The end is reached as soon as the camera signals ready (pin 4 GN) again. Alternatively, the telegram "GSI1" can be used to check whether a valid response is being sent.



NOTE:

The voltage supply must be ensured during the firmware update. An update may take up to 10 minutes.

Read job set (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Read job set (SJS) Request string to sensor (ASCII) | | |
|--|-------------------|--|
| Byte no. | Content | Meaning |
| 1 | S | Read job set (Set Jobset) |
| 2 | J | |
| 3 | S | |
| 4 | 1 | Request version |
| 5 - 7 | X | Length of subsequent file name. Maximum length 250 characters. |
| 8 ... n | X | Optional file name. If no file name is specified, the default name "Jobset.job" is used. |
| Example: | SJS1010jobset.job | |
| Read job set (SJS) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | S | Read job set (Set Jobset) |
| 2 | J | |
| 3 | S | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| 8 - 10 | X | Active job number in the loaded job set |
| Example: | SJSP000001 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | No |
| Status of Ready signal during processing: | | Low |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

The job set with the specified name will be searched for in the /tmp/ directory on the VISOR® vision sensor. If the file exists, this job set is activated. The file is then removed.

Save job set (ASCII)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Save job set (GJS) Request string to sensor (ASCII) | | |
|---|-------------------|--|
| Byte no. | Content | Meaning |
| 1 | G | Save job set from VISOR® |
| 2 | J | |
| 3 | S | |
| 4 | 1 | Request version |
| 5 - 7 | X | Length of subsequent file name. Maximum length 250 characters. |
| 8 ... n | X | Optional file name. If no file name is specified, the default name "Jobset.job" is used. |
| Example: | GJS1010jobset.job | |
| Save job set (GJS) Response string from sensor (ASCII) | | |
| Byte no. | Content | Meaning |
| 1 | G | Save job set from VISOR® |
| 2 | J | |
| 3 | S | |
| 4 | P
F | P: (Pass) Success
F: (Fail) Error |
| 5 - 7 | X | Error codes (Page 94) |
| Example: | GJSP000 | |
| Additional information: | | |
| Accepted in run mode: | | Yes |
| Accepted in configuration mode: | | No |
| Accepted when Ready is low: | | Yes |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) |
| End of telegram: | | Max. 4 bytes (optional) |

The read job set file is now available for download in the /tmp/ directory under the specified name.

11.4.7 Data output ASCII

Output data (ASCII), dynamically composed according to user settings in the software under: SensoConfig / Output / Data output.

Basic string structure:

```
<START> (((<OPTIONAL FIELDS> <SEPARATOR> <PAYLOAD>))) <CHKSUM>
<TRAILER>
```

Output data (ASCII):

| <OPTIONAL FIELDS> | | | | |
|--------------------------------|---|----------------------------|--|----------------------|
| Parameter | Description | Length ASCII [Byte] | Data type | Available for |
| Selected fields | With this checkbox all selected fields are displayed. The checkbox "Selected fields" itself is not displayed. | 16 | The output sequence is from left to right and from top to bottom, i.e. one byte is set per active checkbox, starting with the LSB. | All types |
| Telegram length | Number of characters including the characters for the telegram length itself. | 1 ... 10 | E.g. output string with 10 characters; telegram length 10 + 2 characters (one byte per decimal place) = 12 | All types |
| Status byte | Returns the Trigger mode. | 3 | PPF = Trigger
PPF = Free run | All types |

| <OPTIONAL FIELDS> | | | | |
|--------------------------------|--|------------------------------------|---|----------------------|
| Parameter | Description | Length
ASCII
[Byte] | Data type | Available for |
| Detector results | Output of overall result for each detector. | 4 ... 261 | Byte 1 = AND conjunction of all detectors
Byte 2 = Overall Alignment result
Byte 3 = Overall result of current job
Followed by the number of detectors; one byte per decimal place
Followed by one byte for each detector; P = Detector pass
F = Detector fail | All types |
| Digital outputs | Returns the logic gate result for each digital output. | 2 ... 7 | Byte 1 Number of active outputs (logic gate result assigned)
Followed by bytes 2 – 7; one byte per output
P = Detector pass
F = Detector fail 0 = Inactive output (gap between two active outputs) | All types |

| <OPTIONAL FIELDS> | | | | |
|--------------------------------|--|------------------------------------|--|----------------------|
| Parameter | Description | Length
ASCII
[Byte] | Data type | Available for |
| log. Outputs | Returns the logic gate result for each logic output. | 1 ... 259 | Starting from byte 1
Number of active outputs (logic gate result assigned); 1 byte per decimal place
Following bytes:
One byte per logic output
P = Detector pass
F = Detector fail
0 = Inactive output (gap between two active outputs) | All types |
| Execution time | Returns the execution time for the last evaluation. | 1 ... 3 | Signed integer | All types |
| Active job | Returns the job for the last evaluation. | 1 ... 3 | Unsigned int U8 | All types |

<PAYLOAD>**Overview of detector-specific payload - Values****GENERAL**











| <PAYLOAD> General | | | | |
|--------------------------------|--|------------------------------------|------------------|----------------------|
| Value | Description | Length
ASCII
[Byte] | Data type | Available for |
| "All evaluations" counter | Total number of checks | 1 ... 11 | Signed integer | GENERAL |
| Pass parts counter | Number of inspections with result "OK" | 1 ... 11 | Signed integer | GENERAL |

| <PAYLOAD> General | | | | |
|--------------------------------|--|----------------------------|------------------|----------------------|
| Value | Description | Length ASCII [Byte] | Data type | Available for |
| Fail parts counter | Number of inspections with result "Error" | 1 ... 11 | Signed integer | GENERAL |
| Timeout | Indicates that the maximum cycle time has been exceeded. | 1 | BOOL | GENERAL |
| Recording | Indicates the number of image acquisition repetitions for the last evaluation
Only in combination with repeat mode. | 1 ... 3 | INT | GENERAL |
| String | This field can be used to enter a constant string into the data output. | 1 ... 50 | STRING | GENERAL |









Base values

| <PAYLOAD> Base values | | | | |
|------------------------------------|--|----------------------------|------------------|----------------------|
| Value | Description | Length ASCII [Byte] | Data type | Available for |
| Score | [%] | 1 ... 6 | Signed integer | All detectors |
| Overall result | Boolean detector result | 1 | BOOL | All detectors |
| Execution time | Execution time of individual detector in [msec]. | 1 ... 11 | Signed integer | All detectors |


Position







| <PAYLOAD> Position / location | | | | |
|-------------------------------|---|---------------------|----------------|---|
| Value | Description | Length ASCII [Byte] | Data type | Available for |
| Pos. X | X coordinate for the found position, 1/1000 [user unit] | 1 ... 11 | Signed integer |  |
| Pos. Y | Y coordinate for the found position, 1/1000 [user unit] | 1 ... 11 | Signed integer |  |
| Pos. Z | Z coordinate of the found position, 1/1000 [user unit] | | Signed integer | 
With Result offset:
 |
| Delta Pos. X | X position delta between the taught object and the found object, 1/1000 [user unit] | 1 ... 11 | Signed integer |  |
| Delta Pos. Y | Y position delta between the taught object and the found object, 1/1000 [user unit] | 1 ... 11 | Signed integer |  |
| Delta Pos. Z | Z position delta between the taught object and the found object, 1/1000 [user unit] | 1 ... 11 | Signed integer | 
With Result offset:
 |
| Angle X | Orientation of the found object, relative to the X-axis, 1/1000 [°] | 1 ... 11 | Signed integer | 
With Result offset:
 |

| <PAYLOAD> Position / location | | | | |
|--|--|----------------------------|------------------|---|
| Value | Description | Length ASCII [Byte] | Data type | Available for |
| Angle Y | Orientation of the found object, relative to the Y-axis, 1/1000 [°] | 1 ... 11 | Signed integer | 
With Result offset:
 |
| Angle Z | Orientation of the found object, relative to the Z-axis, 1/1000 [°] | 1 ... 11 | Signed integer | 
A |
| Angle (45) | Orientation of bounding box for found code [°], Value range: -45° to 45° | 1 ... 6 | Signed integer |  |
| Angle (180) | Orientation of object width (long axis) [°], Value range: -90° to 90°
0° = East, counterclockwise | 1 ... 7 | Signed integer |  |
| Angle (360) | Orientation of object width (long axis) [°], Value range: -180° to 180°
0° = East, counterclockwise | 1 ... 7 | Signed integer |  |
| Delta Angle X | Angle between taught-in and found object, referred to the X-axis, 1/1000 [°] | 1 ... 7 | Signed integer | 
With Result offset:
 |




| <PAYLOAD> Position / location | | | | |
|--|--|---|------------------|---|
| Value | Description | Length ASCII [Byte] | Data type | Available for |
| Delta Angle Y | Angle between taught-in and found object, referred to the Y-axis, 1/1000 [°] | 1 ... 7 | Signed integer | 
With Result offset:
 |
| Delta Angle Z | Angle between taught-in and found object, referred to the Z-axis, 1/1000 [°] | 1 ... 7 | Signed integer |  |
| Pose 3D (X, Y, Z, Angle X, Angle Y, Angle Z) | Coordinates of the found object, 1/1000 [user unit]
Angle: 1/1000 degrees | 1...7 bytes per value; separated by specified separator | Signed integer | 
With Result offset:
 |
| Delta Pose 3D (X, Y, Z, Angle X, Angle Y, Angle Z) | Delta coordinates of the found object, 1/1000 [user unit]
Angle: 1/1000 degrees | 1...7 bytes per value; separated by specified separator | Signed integer | 
With Result offset:
 |
| Position control | | 1 | BOOL |  |







Measurement

| <PAYLOAD> Measurement | | | | |
|------------------------------------|---|----------------------------|------------------|---|
| Value | Description | Length ASCII [Byte] | Data type | Available for |
| height | Height of geometric element [user unit]*,
Height ≥ 0, height ≤ width | 1 ... 11 | Signed integer |  |



| <PAYLOAD> Measurement | | | | |
|------------------------------------|--|----------------------------|------------------|---|
| Value | Description | Length ASCII [Byte] | Data type | Available for |
| Width | Width of geometric element [user unit]*, Width ≥ 0, width ≥ height | 1 ... 11 | Signed integer |   |
| Radius | Radius of fitted circle [user unit] | 1 ... 11 | Signed integer |  |
| Area | Area of BLOB without holes, 1/1000 [pixels] | 1 ... 11 | Signed integer |  |
| Area (incl. holes) | Area of BLOB including holes, 1/1000 [pixels] | 1 ... 11 | Signed integer |  |
| Distance | Calculated distance [user unit] | 1 ... 11 | Signed integer |  |

Identification



| <PAYLOAD> Identification | | | | |
|---------------------------------------|-----------------------------|----------------------------|------------------|---|
| Value | Description | Length ASCII [Byte] | Data type | Available for |
| String length | Length of read code [bytes] | 1 ... 3 | Signed integer |    |

| <PAYLOAD> Identification | | | | |
|---------------------------------------|--|------------------------------------|------------------|--|
| Value | Description | Length
ASCII
[Byte] | Data type | Available for |
| String | Content of the read code.
Depending on the code, the string length may vary. If a fixed string length is desired, the minimum string length (detector-specific payload) and the maximum string length (detector settings) must be set to the same value (e.g. 127). | 0 ... 255 | STRING |   A |
| String comparison | Content check for the read information.
The content of the read information is checked on the basis of regular expressions (see detector Data-code, Reference string tab) | 1 | BOOL |   A |
| Truncated | Code complete or truncated
F: Code complete
P: Code truncated | 1 | BOOL |   A |





Identification - quality







| <PAYLOAD> Identification - Quality | | | | |
|---|---|--|--|---|
| Value | Description | Length
ASCII
[Byte] | Data type | Available for |
| Quality - overall | Output of all Q parameters. Depending on the selected code type and standard. | 1 byte per value; separated by specified separator
For 2D code parameter Q9 (mean light): 1...3 | Unsigned Char;
for 2D Code Q9 (Meanlight)
Unsigned Short |  |
| Quality - individual | Output of individual quality values: Selection Q1-Q24 depending on the selected code type and standard.
Numbers: 1-4
Letters: A-F | 1
For 2D code parameter Q9 (mean light): 1...3 | Unsigned Char;
for 2D Code Q9 (Meanlight)
Unsigned Short |  |
| Min. Quality | Used to check whether the minimum required quality is being met | 1 ... 7 | Unsigned int | A |

Color



| <PAYLOAD> Color | | | | |
|--|--|---------------------|----------------|---|
| Value | Description | Length ASCII [Byte] | Data type | Available for |
| Color value: <ul style="list-style-type: none"> • Red, green, blue • Hue, saturation, lightness • Luminance, a, b | Value for color parameter | 0 ... 7 | Signed integer |  |
| Color distance | Distance of the current color versus the taught-in color | 0 – 7 | Signed integer |  |






Counting / number








| <PAYLOAD> Counting / number | | | | |
|-----------------------------|--|---------------------|----------------|---|
| Value | Description | Length ASCII [Byte] | Data type | Available for |
| Number of objects | Number of objects found [units] | 1 ... 5 | Signed integer |  |
| Number of valid objects | Number of valid objects found [units] | 1 ... 5 | Signed integer |  |
| Number of search stripes | Number of parallel search stripes into which the width of the search range is divided. [units] | 1 ... 5 | Signed integer |  (Edge detector only)
 |






| <PAYLOAD> Counting / number | | | | |
|--|---|----------------------------|------------------|---|
| Value | Description | Length ASCII [Byte] | Data type | Available for |
| Number of valid search stripes | Number of search stripes used to generate results [units] | 1 ... 3 | Signed integer |  (Edge detector only)
 |
| Result vector | Vector containing the result (1/0) of the instances found | | |    |
| Too many BLOBs | | 1 | BOOL |  |

Extended

| <PAYLOAD> Extended | | | | |
|---------------------------------|---|----------------------------|------------------|--|
| Value | Description | Length ASCII [Byte] | Data type | Available for |
| Scaling | Current scaling factor to the taught-in reference. 1/1000 (factor). Value range of 0.5 to 2 | 3 ... 4 | Unsigned int |  (Contour matching only)
 |
| Eccentricity | Numerical eccentricity
Value range of 0.0 to 1.0 | N | Signed integer | |

| <PAYLOAD> Extended | | | | |
|---------------------------------|--|------------------------------------|------------------|---|
| Value | Description | Length
ASCII
[Byte] | Data type | Available for |
| Security | Output of the security values of the individual characters. The reliability value specifies how reliably the reader was able to interpret a character. Value range of 0 to 100 [%] | N | Unsigned int | A |
| Reference string met | The output string matches the reference string. | 1 | BOOL | A |
| contrast | Code contrast Value range of 0 to 100 [%] | N | Unsigned int |  |
| Correction | Number of modules corrected by error corrections [units] | N | Unsigned int |  |
| Contour length | Number of pixels of outer contour, 1/1000 [pixels] | N | Signed integer |  |
| Compactness | BLOB compactness (circle =1; other > 1). The more the shape of the BLOB deviates from a circle, the greater the compactness value will be. | N | Signed integer |  |
| Center of gravity X | X coordinate of centroid, 1/1000 | N | Signed integer |  |

| <PAYLOAD> Extended | | | | |
|---------------------------------|---|----------------------------|------------------|---|
| Value | Description | Length ASCII [Byte] | Data type | Available for |
| Center of gravity Y | Y coordinate of centroid, 1/1000 | N | Signed integer |  |
| Gray scale value, average | Average gray scale value of all the pixels that belong to the BLOB. | N | Signed integer |  |
| Min. signal threshold | Lower threshold for the binarization of the objects. 0...255 | 1 ... 3 | Unsigned int |  |
| Max. signal threshold | Upper threshold for the binarization of the objects. 0...255 | 1 ... 3 | Unsigned int |  |
| Inverted signal threshold | Specifies whether the range Min <-> Max is inverted.
P: inverted
F: not inverted | 1 | Unsigned Char |  |
| Deviation, inside | Returns the largest deviation between the BLOB contour and the contour of the geometric element (deviation inside the fitted circle).
[User unit * 1000] | 1 ... 7 | Signed integer |  |
| Deviation, outside | Returns the largest deviation between the BLOB contour and the contour of the geometric element (deviation outside the fitted circle).
[user unit] | 1 ... 7 | Signed integer |  |

| <PAYLOAD> Extended | | | | |
|---------------------------------|---|----------------------------|------------------|---|
| Value | Description | Length ASCII [Byte] | Data type | Available for |
| Deviation, mean | Returns the mean of the absolute "inside" and "outside" deviation values between the BLOB contour and the contour of the geometric element. | 1 ... 7 | Signed integer |  |
| Axial ratio | Ratio of the long to the short axis (a / b) | 1 ... 7 | Signed integer |  |
| Face up / down, area | Face up / down position, based on: area, position indicated by sign, 1/1000 | N | Signed integer |  |
| Result index | List index | N | Signed integer |  |
| Search stripe distance | Calculated distance [user unit] / 1000 per pair of search stripes | 1 ... 11 | Signed integer |  |

| <CHKSUM> | | | | |
|-----------------------|--|----------------------------|------------------|----------------------|
| Parameter | Description | Length ASCII [Byte] | Data type | Available for |
| Check sum | XOR check sum of all bytes in the telegram. Is transmitted as the last byte. | 1 | Unsigned int | All types |

| <TRAILER> | | | | |
|------------------------|--|------------------------------------|------------------|----------------------|
| Parameter | Description | Length
ASCII
[Byte] | Data type | Available for |
| Start | User-defined, up to a max. of 8 characters | 0 ... 8 | Unsigned int | All types |


***NOTE:**

If no calibration has been performed, all values refer to pixels.

11.5 Description Telegrams BINARY

11.5.1 General

Reset statistics (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Reset Statistics (RST) Request string to sensor (BINARY) | | | |
|--|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x04 | Reset statistics |
| Reset Statistics (RST) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x04 | Reset statistics |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

11.5.2 Control

Trigger (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Trigger (TRG) Request string to sensor (BINARY) | | | |
|---|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x01 | Trigger, (simple trigger without index, via port 2006) |
| Trigger (TRG) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x01 | Trigger, (response to trigger command without index, via port 2006. If defined: Result data without index via port 2005) |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | Yes |
| Accepted when Ready is low: | | | No |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Extended trigger (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Extended Trigger (TRX) Request string to sensor (BINARY) | | | |
|---|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length 6 bytes + length of subsequent data (n) |
| 5 | Unsigned Char | 0x13 | Extended trigger (trigger with index, for correlation of trigger to corresponding result data, via port 2006) |
| 6 | Unsigned Char | 0xXX | Length of following data (n) |
| 7 ... n | Unsigned Char | 0xXX | Data |
| Extended Trigger (TRX) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length |
| 5 | Unsigned Char | 0x13 | Extended Trigger command, (response to trigger with index and result data, via port 2006, for correlation of trigger to corresponding result, Result data without index, via port 2005 also) |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Length of following data (n) |
| 9 ... n | Unsigned Char | 0xXX | Data of sending command |
| n+1 | Unsigned Char | 0xXX | Operating mode
0 = Config mode
1 = Run mode |
| n + 2 ... n + 5 | Unsigned int | 0xXX | Length of result data |
| n + 6 ... m | Unsigned Char | 0xXX | Result data |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |

| | |
|---|---|
| Accepted in configuration mode: | Yes |
| Accepted when Ready is low: | No |
| Status of Ready signal during processing: | Low |
| Supported interfaces: | Telegram: Availability and supported interfaces (Page 91) |

Trigger Robotics (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Trigger Robotics (TRR) Request string to sensor (BINARY) | | | |
|--|----------------|---------|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length 31 (0x1F) + Length of trigger identifier in Bytes |
| 5 | Unsigned Char | 0x37 | Trigger Robotics |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | Length of trigger identifier in bytes |
| 8-n | Unsigned Char | 0xXX | Trigger Identifier |
| n+1...n+4 | Unsigned int | 0xXX | Pose_TCP Pos. X
(in user unit * 1000) |
| n+5...n+8 | Unsigned int | 0xXX | Pose_TCP Pos. Y
(in user unit * 1000) |
| n+9...n+12 | Unsigned int | 0xXX | Pose_TCP Pos. Z
(in user unit * 1000) |
| n+13...n+16 | Unsigned int | 0xXX | Pose_TCP Angle X
(in degrees * 1000) |
| n+17...n+20 | Unsigned int | 0xXX | Pose_TCP Angle Y
(in degrees * 1000) |
| n+20...n+24 | Unsigned int | 0xXX | Pose_TCP Angle Z
(in degrees * 1000) |
| Trigger Robotics (TRR) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length 8 (0x08) + Length of trigger identifier in Bytes |
| 5 | Unsigned Char | 0x37 | Trigger Robotics, (Response to command Trigger without index, via port 2006. If defined: Result data without index via port 2005) |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |

| | | | |
|---------------------------------|---------------|------|---|
| 8 | Unsigned Char | 0xXX | Length of trigger identifier |
| 9-n | Unsigned Char | 0xXX | Trigger Identifier |
| n+1 | Unsigned Char | 0xXX | Operation Mode
0x00 = Config
0x01 = Run |
| n+2...n+5 | Unsigned int | 0xXX | Length of the result data in bytes |
| n+6...m | Unsigned int | 0xXX | Result data |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | Yes |
| Accepted when Ready is low: | | | No |

Note: For "Calibration plate (Robotics)" and "Point pair list (Robotics)" only the X and Y position are taken into account. The other values (position Z and rotations) must be 0.

Set Trigger ID (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Set Trigger ID (STI) Request string to sensor (BINARY) | | | |
|--|--|---------|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length
7 Bytes + length of Trigger ID |
| 5 | Unsigned Char | 0x2E | Set trigger ID |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | Length of the following data (max 99) |
| 8-n | Unsigned Char | 0xXX | Trigger ID |
| Example: | 0x00 0x00 0x00 0x0D 0x2E 0x01 0x06 0x30 0x31 0x32 0x33 0x34 0x35 | | |
| Set Trigger ID (STI) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x2E | Set trigger ID |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Example: | 0x00 0x00 0x00 0x07 0x2E 0x00 0x00 | | |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | Yes |
| Accepted when Ready is low: | | | Yes |

Job change (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Job change (CJB) Request string to sensor (BINARY) | | | |
|--|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x06 | Telegram length |
| 5 | Unsigned Char | 0x02 | Change job |
| 6 | Unsigned Char | 0xXX | Job no. XX = 1 - n |
| Job change (CJB) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x09 | Telegram length |
| 5 | Unsigned Char | 0x02 | Change job |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Trigger mode
0x00: Trigger
0x01: Free run |
| 9 | Unsigned Char | 0xXX | Job no. XX = 1 - n |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |



NOTE:

If an error occurs during the job change, it is possible to change to Job 1.

Job Change Permanent (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Job Change Permanent (CJP) Request string to sensor (BINARY) | | | |
|--|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x06 | Telegram length |
| 5 | Unsigned Char | 0x22 | Job change permanent |
| 6 | Unsigned Char | 0xXX | Job no. XX = 1 - n |
| Job Change Permanent (CJP) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x09 | Telegram length |
| 5 | Unsigned Char | 0x22 | Job change permanent |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Trigger Mode
0x00: Trigger
0x01: Free run |
| 9 | Unsigned Char | 0xXX | Job no. XX = 1 - n |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |



NOTE:

If an error occurs during the job change, it is possible to change to Job 1.

Job change by job name (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Job change by job name (CJN) Request string to sensor (BINARY) | | | |
|--|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length 7 bytes + length job name (n) |
| 5 | Unsigned Char | 0x2C | Job change by job name |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | Job name length (n) |
| 8 - n | Unsigned Char | 0xXX | Job name |
| Job change by job name (CJN) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x08 | Telegram length |
| 5 | Unsigned Char | 0x2C | Job change by job name |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Trigger mode
0x00: Trigger
0x01: Free run |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

11.5.3 Job settings

Auto working distance (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Auto working distance (AFC) Request string to sensor (BINARY) | | | |
|---|---------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length 11 Bytes (0x0B) + selected options 8 Bytes (0x08) |
| 5 | Unsigned Char | 0x32 | Auto working distance |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary
0x01: Permanent |
| 8 | Unsigned Char | 0xXX | Step size of search (0x01 - 0x05) |
| 9 | Unsigned Char | 0xXX | Selection of distance value
0x00: Highest score
0x01: Min. Working distance
0x02: Max. working distance
0x03: Average working distance
0x04: Median working distance
0x05: Highest score - output of all working distances found |
| 10 | Unsigned Char | 0xXX | Unit
0x00: 1/1000 millimeters (μm)
0x01: Motor steps |
| 11 | Unsigned Char | 0xXX | Selection of search range
0x00: Entire range
0x01: Selected range |
| 12...15 | Unsigned int | X | Start of search range (only if search range selection == 0x01) |
| 16...19 | Unsigned int | X | End of search range (only if selection Search range == 0x01) |
| Auto working distance (AFC) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |

| | | | |
|---|----------------|------|--|
| 1 - 4 | Unsigned int | 0xXX | Telegram length 11 Bytes (0x0B) + working distances + score values |
| 5 | Unsigned Char | 0x32 | Auto working distance |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | Unsigned int | X | Number of output working distances |
| 12 - n | Unsigned int | X | Distance value in 1/1000 mm or motor steps (4 bytes per output working distance) |
| n-m | Unsigned int | X | Score value to distance value multiplied by 1000 (4 bytes per output working distance) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Set working distance (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Set working distance (SFC) Request string to sensor (BINARY) | | | |
|--|----------------|---------|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0D | Telegram length |
| 5 | Unsigned Char | 0x31 | Set working distance |
| 6 | Unsigned Char | 0xX1 | Request version |
| 7 | Unsigned Char | 0xXX | 0: Temporary
1: Permanent |
| 8 | Unsigned Char | 0xXX | Movement
0: Absolute
1: Relative
2: Absolute with reinitialization |
| 9 | Unsigned Char | 0xXX | Unit
0: 1/1000 millimeters
4: Steps |
| 10 - 13 | Signed integer | 0xXX | Working distance |
| Set working distance (SFC) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0B | Telegram length |
| 5 | Unsigned Char | 0x31 | Set working distance |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | INT | 0xXX | Current working distance |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Read working distance (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Read working distance (GFC) Request string to sensor (BINARY) | | | |
|--|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x30 | Read working distance |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | Unit
0x00: 1/1000 millimeter
0x04: Steps |
| Read working distance (GFC) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0B | Telegram length |
| 5 | Unsigned Char | 0x30 | Read working distance |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | INT | 0xXX | Current working distance |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Auto shutter speed (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Auto shutter speed (ASH) Request string to sensor (BINARY) | | | |
|--|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x07 | Auto shutter speed |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary
0x01: Permanent |
| Auto shutter speed (ASH) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0F | Telegram length |
| 5 | Unsigned Char | 0x07 | Auto shutter speed |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | INT | 0xXX | Auto shutter speed value |
| 12 - 15 | INT | 0xXX | Score |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Set shutter speed value (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Set shutter speed (SSP/SST) Request string to sensor (BINARY) | | | |
|---|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x09 | Telegram length |
| 5 | Unsigned Char | 0xXX | 0x0E Set shutter speed temporarily
0x0F Set shutter speed permanently |
| 6 - 9 | Unsigned int | 0xXX | Shutter speed value in 1/1000 ms |
| Set shutter speed (SSP/SST) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0xXX | 0x0E Set shutter speed temporarily
0x0F Set shutter speed permanently |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Read shutter speed value (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Read Shutter Speed Value (GSH) Request string to sensor (BINARY) | | | |
|--|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x17 | Read shutter speed value |
| Read Shutter Speed Value (GSH) Response string from sensor (BINARY) | | | |
| 1 - 4 | Unsigned int | 0x0B | Telegram length |
| 5 | Unsigned Char | 0x17 | Read shutter speed value |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | Unsigned int | 0xXX | Shutter speed value |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode:: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Set gain value (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Set gain value (SGA) Request string to sensor (BINARY) | | | |
|--|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0A | Telegram length |
| 5 | Unsigned Char | 0x1B | Set gain value |
| 6 | Unsigned Char | 0xXX | 0: Temporary
1: Permanent |
| 7 - 10 | Unsigned int | 0xXX | Gain value * 1000 |
| Set gain value (SGA) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0B | Telegram length |
| 5 | Unsigned Char | 0x1B | Set gain value |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | Unsigned int | 0xXX | Current gain value (value *1000) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Read gain value (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Read gain value (GGA) Request string to sensor (BINARY) | | | |
|---|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x1C | Read gain value |
| Read gain value (GGA) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0B | Telegram length |
| 5 | Unsigned Char | 0x1C | Read gain value |
| 6 | Unsigned Short | 0xFF | Error codes (Page 94) |
| 7 | Short | 0xFF | |
| 8 - 11 | Unsigned int | 0xFF | Current gain value * 1000 |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Set parameter (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Set parameters (SPP/SPT) Request string to sensor (BINARY) | | | |
|--|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length = 9 bytes + length of the selected parameter |
| 5 | Unsigned Char | 0xXX | 0x05: Set parameter permanently
0x06: Set parameter temporarily |
| 6 | Unsigned Char | 0xXX | Detector no., XX = 1- n |
| 7 | Unsigned Char | 0xXX | Command Set Reference string / value, see table Overview detector Parameter |
| 8 - 9 | Unsigned Short | 0xXX | Length of new reference string / value (n), see table Overview of detector Parameter |
| 10 ... n | Unsigned Char | 0xXX | Reference string / value |
| Set parameters (SPP/SPT) Response string from sensor (BINARY) | | | |
| (may be delayed up to 4-5 seconds) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x08 | Telegram length + length of the selected parameter in bytes |
| 5 | Unsigned Char | 0xXX | 0x05: Set parameter permanently
0x06: Set parameter temporarily |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |

| | | | |
|---|---------------|--|--|
| 8 | Unsigned Char | 0xXX | Parameter type
0x00: I8
0x01: U8
0x02: I16
0x03: U16
0x04: I32
0x05: U32
0x06: I40
0x07: U40
0x08: Float
0x09: Double
0x0A: String
0x0B: Boolean
0x0C: Special signed8
0x0D: Undefined |
| Additional information: | | | |
| Accepted in run mode: | | Yes | |
| Accepted in configuration mode: | | No | |
| Accepted when Ready is low: | | Yes | |
| Status of Ready signal during processing: | | Low | |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) | |

Overview Detector Parameters (set / read)

| Detector | Function | Value | Multiplier | Length |
|--------------------------------------|---|-------|------------|----------------------------|
| Alignment | | | | |
| Pattern matching
Contour matching | Threshold value Min. | 0x01 | 1000 | 4 |
| | Threshold value Max. | 0x02 | 1000 | 4 |
| | Result offset
0: "Off"
1: "Image plane (in pixels)"
2: "Align (2D)"
3: "Robot (3D)" | 0x1E | 1 | 1 |
| | Result offset Image plane: Pos. X | 0x1F | 1000 | 4 |
| | Result offset Image plane: Pos. Y | 0x20 | 1000 | 4 |
| | Result offset Image plane: angle | 0x21 | 1000 | 4 |
| | Result offset
Align (2D), Robot (3D): Pos. X,
Pos. Y, Pos. Z, Angle X, Angle Y,
Angle Z | 0x22 | 1000 | 24 (6 * 4 bytes per value) |
| | Calculate Result offset* with transmitted position <ul style="list-style-type: none"> • Align (2D): Pos. X, Pos. Y, 0, 0, 0, Angle Z • Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z *A valid position for the detector must be available | 0x23 | 1000 | 24 (6 * 4 bytes per value) |
| Edge detector | Probe 1: Transition
0: Any
1: Dark to light
2: Light to dark | 0x65 | 1 | 1 |
| | Probe 2: Transition
0: Any
1: Dark to light
2: Light to dark | 0x66 | 1 | 1 |

| Detector | Function | Value | Multiplier | Length |
|---|---|-------|------------|----------------------------------|
| | Probe 3: Transition
0: Any
1: Dark to light
2: Light to dark | 0x67 | 1 | 1 |
| | Probe 1: Threshold value Min. | 0x68 | 1000 | 4 |
| | Probe 2: Threshold value Min. | 0x69 | 1000 | 4 |
| | Probe 3: Threshold value Min. | 0x6A | 1000 | 4 |
| Detector | | | | |
| Pattern matching
Contour
Contour 3D | Threshold value Min. | 0x01 | 1000 | 4 |
| | Threshold value Max. | 0x02 | 1000 | 4 |
| | Result offset
0: "Off"
1: "Image plane (in pixels)"
2: "Align (2D)"
3: "Robot (3D)" | 0x1E | 1 | 1 |
| | Result offset Image plane: Pos.
X | 0x1F | 1000 | 4 |
| | Result offset Image plane: Pos.
Y | 0x20 | 1000 | 4 |
| | Result offset Image plane: angle | 0x21 | 1000 | 4 |
| | Result offset
Align (2D), Robot (3D): Pos. X,
Pos. Y, Pos. Z, Angle X, Angle Y,
Angle Z | 0x22 | 1000 | 24 (6 * 4
bytes per
value) |
| | Calculate Result offset* with
transmitted position <ul style="list-style-type: none"> • Align (2D): Pos. X, Pos. Y,
0, 0, 0, Angle Z • Robot (3D): Pos. X, Pos.
Y, Pos. Z, Angle X, Angle
Y, Angle Z *A valid position for the detector
must be available | 0x23 | 1000 | 24 (6 * 4
bytes per
value) |
| Gray | Threshold value Min. | 0x01 | 1000 | 4 |
| | Threshold value Max. | 0x02 | 1000 | 4 |
| | Grayscale value Min. | 0x65 | 1000 | 4 |

| Detector | Function | Value | Multiplier | Length |
|-----------------------------------|--|-------|------------|--------|
| | Grayscale value Max. | 0x66 | 1000 | 4 |
| | Invert grayscale value | 0x67 | 1 | 4 |
| Contrast
Brightness | Threshold value Min. | 0x01 | 1000 | 4 |
| | Threshold value Max. | 0x02 | 1000 | 4 |
| Caliper | Threshold value Distance Min. | 0x65 | 1000 | 4 |
| | Threshold value Distance Max. | 0x66 | 1000 | 4 |
| | Invert distance threshold value | 0x67 | 1 | 1 |
| | Distance mode
0: Minimum
1: Maximum
2: Mean
3: Median
4: Smallest opposite
5: Largest opposite | 0x68 | 1 | 1 |
| | Probe 1: Threshold value Min. | 0x69 | 1000 | 4 |
| | Probe 2: Threshold value Min. | 0x6A | 1000 | 4 |
| | Probe 1: Smoothing | 0x6B | 1000 | 4 |
| | Probe 2: Smoothing | 0x6C | 1000 | 4 |
| | Probe 1: Transition
0: Any
1: Dark to light
2: Light to dark | 0x6D | 1 | 1 |
| | Probe 2: Transition
0: Any
1: Dark to light
2: Light to dark | 0x6E | 1 | 1 |
| | Probe 1: Number of search stripes | 0x6F | 1 | 1 |
| Probe 2: Number of search stripes | 0x70 | 1 | 4 | |
| BLOB | Grayscale value Min. | 0x65 | 1000 | 4 |
| | Grayscale value Max. | 0x66 | 1000 | 4 |
| | Invert grayscale value
0: not inverted
1: inverted | 0x67 | 1 | 1 |
| | Threshold value Number of BLOBs Min. | 0x78 | 1 | 1 |

| Detector | Function | Value | Multiplier | Length |
|----------------------------|---|-------|--|----------------------|
| | Threshold value Number of BLOBs Max. | 0x79 | 1 | 1 |
| | Invert number threshold value
0: not inverted
1: inverted | 0x7A | 1 | 1 |
| | Number of set features (read only) | 0x7B | 1 | 1 |
| | Selection of a feature from the list | 0x7C | 1 | 1 |
| | Feature threshold value Min. | 0x7D | 1000 | 4 |
| | Feature threshold value Max. | 0x7E | 1000 | 4 |
| | Invert feature threshold value | 0x7F | 1 | 1 |
| Barcode
Datacode
OCR | Reference string | 0x65 | - | n (length of string) |
| | Reference string | 0x65 | - | n (length of string) |
| | Reference string | 0x65 | - | n (length of string) |
| Color Value
Color Value | Color space (read only) | 0x15 | 0x00 = RGB
0x01 = HSV
0x02 = LAB | 1 |
| | Channel selection (read only) | 0x16 | Bit field one digit per color channel | 1 |
| | Color channel 1: Threshold value Min. | 0x65 | 1000 | 4 |
| | Color channel 1: Threshold value Max. | 0x66 | 1000 | 4 |
| | Color channel 1: Invert threshold value | 0x67 | 1 | 1 |
| | Color channel 2: Threshold value Min. | 0x68 | 1000 | 4 |
| | Color channel 2: Threshold value Max. | 0x69 | 1000 | 4 |
| | Color channel 2: Invert threshold value | 0x6A | 1 | 1 |

| Detector | Function | Value | Multiplier | Length |
|-------------------|---|-------|---------------------------------------|----------------------|
| | Color channel 3: Threshold value Min. | 0x6B | 1000 | 4 |
| | Color channel 3: Threshold value Max. | 0x6C | 1000 | 4 |
| | Color channel 3: Invert threshold value | 0x6D | 1 | 1 |
| Color List | Color space (read only) | 0x15 | 0 = RGB
1 = HSV
2 = LAB | 3 |
| | Channel selection (read only) | 0x16 | Bit field one digit per color channel | 4 |
| | Color distance threshold value | 0x65 | 1000 | N |
| | Set color distance threshold value active | 0x66 | 1 | N |
| | Number of colors in list | 0x67 | 1 | N |
| | Selection of a color from the list | 0x68 | 1 | N |
| | Color value of the selected color (color channel 1, color channel 2, color channel 3, color channel 4 [constantly 0]) | 0x69 | 1000 | 32 |
| Busbar Wafer | Threshold value Min. | 0x01 | 1000 | N |
| | Threshold value Max. | 0x02 | 1000 | N |
| Result processing | Name of the active expression | 0x7A | - | n (length of string) |
| | Current expression | 0x7V | - | n (length of string) |

Read parameter (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Read parameter (GPA) Request string to sensor (BINARY) | | | |
|---|----------------|---------|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x0A | Get parameter |
| 6 | Unsigned Char | 0xXX | Detector no., XX = 1- n |
| 7 | Unsigned Char | 0xXX | Command Set Reference string / value, see table Overview detector Parameter |
| Read parameter (GPA) Response string from sensor (BINARY) | | | |
| (may be delayed up to 4-5 seconds) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length = 10 Bytes + length of the selected parameter in Byte |
| 5 | Unsigned Char | 0x0A | Get parameter |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Parameter type string |
| 9 - 10 | Unsigned Short | 0xXX | Length of read parameter (n) |
| 11 ... n + n | Unsigned Char | 0xXX | Reference string / value |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Set search range (ROI) (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Set ROI (SRP/SRT) Request string to sensor (BINARY) | | | |
|---|---------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length in bytes
24 bytes: circle
32 bytes: rectangle, ellipse, free form |
| 5 | Unsigned Char | 0xXX | 0x10: Set parameter temporarily
0x11: Set parameter permanently |
| 6 - 9 | Unsigned int | 0xXX | 19 bytes: circle
27 bytes: rectangle, ellipse, free form |
| 10 | Unsigned Char | 0xXX | Detector no. |
| 11 | Unsigned Char | 0xXX | Search range (ROI) Type
0x00: Search area (yellow)
0x01: Teach area (red)
0x02: Position control (blue) |
| 12 | Unsigned Char | 0xXX | Search range (ROI) Shape
0x01: Circle
0x02: Rectangle
0x03: Ellipse
0x04: Free shape |
| 13 - 16 | Unsigned int | 0xXX | ROI parameter: Center X (value in [px] * 1000) |
| 17 - 20 | Unsigned int | 0xXX | ROI parameter: Center Y (value in [px] * 1000) |
| 21 - 24 | Unsigned int | 0xXX | ROI parameter: half width or radius X (value in [px] * 1000) |
| Only for ellipse / rectangle / free form: | | | |
| 25 - 28 | Unsigned int | 0xXX | ROI parameter: half height or radius Y (value in pixels * 1000) |
| 29 - 32 | Unsigned int | 0xXX | ROI parameter: Angle in ° degree (value in ° [degrees] * 1000) |
| Set ROI (SRP/SRT) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 | Unsigned int | 0x07 | Telegram length |

| | | | |
|---|----------------|------|---|
| 5 | Unsigned Char | 0xXX | 0x10: Set parameter permanently
0x11: Set parameter temporarily |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |
| Parameter: | | | The parameters are given in the coordinate system of the Alignment and not in the coordinate system of the image. |

Read search range (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)


| Read search range (GRI) Request string to sensor (BINARY) | | | |
|--|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x12 | Get ROI |
| 6 | Unsigned Char | 0xXX | Detector no. |
| 7 | Unsigned Char | 0xXX | Search range (ROI) Type
0x00: Search area (yellow)
0x01: Teach area (red)
0x02: Position control (blue) |
| Read search range (GRI) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length |
| 5 | Unsigned Char | 0x12 | Get ROI |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | Unsigned int | 0xXX | Search range (ROI) Info Length in bytes from Byte 8 |
| 12 | Unsigned Char | 0xXX | Detector no. |
| 13 | Unsigned Char | 0xXX | Search range (ROI) Type
0x00: Search area (yellow)
0x01: Teach area (red)
0x02: Position control (blue) |
| 14 | Unsigned Char | 0xXX | Search range (ROI) Shape
0x01: Circle
0x02: Rectangle
0x03: Ellipse
0x04: Free form |
| 15 - 18 | Unsigned int | 0xXX | ROI parameter: Center X (value in pixels * 1000) |
| 19 - 22 | Unsigned int | 0xXX | ROI parameter: Center Y (value in pixels * 1000) |

| | | | |
|---|--------------|------|---|
| 23 -26 | Unsigned int | 0xXX | ROI parameter: Half width / radius X (value in pixels [px] * 1000) |
| Only for ellipse / rectangle / free form: | | | |
| 27 - 30 | Unsigned int | 0xXX | ROI parameter: Half height / radius Y (value in pixels [px] * 1000) |
| 31 - 34 | Unsigned int | 0xXX | ROI parameter: Angle in ° (value in ° * 1000) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegram: Availability and supported interfaces (Page 91) |

Read job list (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)


| Read job list (GJL) Request string to sensor (BINARY) | | | |
|---|---|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x14 | Read job list |
| Read job list (GJL) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length |
| 5 | Unsigned Char | 0x14 | Read job list |
| 6 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0x01 | Constant |
| 9 | Unsigned Char | 0xXX | Number of jobs |
| 10 | Unsigned Char | 0xXX | Active job number |
|  | NOTE:
The following byte sequence is repeated for each job from 1 to "Number of jobs".
The byte numbers shift accordingly. | | |
| 11 | Unsigned Char | 0xXX | Number of subsequent bytes. This can be used to specify a unique name for job n. |
| 11 ... n | Char | 0xXX | From this position, the name for job n follows in the specified length. |
| n + 1 ... n + 3 | Unsigned Char | 0xXX | Number of subsequent bytes. A description for job n can be specified. |
| n + 4 ... m | Char | 0xXX | From this position, the description for Job 1 follows in the specified length. |
| m + 1 ... m + 3 | Unsigned Char | 0xXX | Number of subsequent bytes. This can be used to specify a unique name for the author of job n. |
| m + 4 ... k | Char | 0xXX | From this position, the name for the author of job n follows in the specified length. |

| | | | |
|---|--------------|------|---|
| k + 1 ... k + 7 | Unsigned int | 0xXX | Date of creation of Job n (7 bytes) |
| k + 8 ... k + 14 | Unsigned int | 0xXX | Date of last modification of job n (7 bytes) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegram: Availability and supported interfaces (Page 91) |

Read detector list (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Read detector list (GDL) Request string to sensor (BINARY) | | | |
|---|---|---|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x15 | Read detector list |
| Read detector list (GDL) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length |
| 5 | Unsigned Char | 0x18 | Read detector list |
| 6 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Job number of current job |
| 9 | Unsigned Char | 0xXX | Number of detectors in the current job |
| |  | NOTE:
The following byte sequence is repeated for each detector in the job. The byte numbers shift accordingly. | |
| 10 | Unsigned Char | 0xXX | Number of subsequent bytes. This allows a unique name for the detector n to be specified. |
| 11 ... n | Unsigned Char | 0xXX | From this position, the name for detector n follows, in the given length. |

| | | | |
|---|---------------|--|--|
| n + 1 ... n + 2 | Unsigned Char | 0xXX | Detector
0x01: Pattern matching
0x04: Contour
0x05: Gray
0x06: Contrast
0x07: Brightness
0x0A: Wafer
0x0B: OCR
0x0D: Datacode
0x0E: Barcode
0x11: Busbar
0x12: Color Value
0x13: Color Area
0x14: Color List
0x15: Caliper
0x16: BLOB |
| Additional information: | | | |
| Accepted in run mode: | | Yes | |
| Accepted in configuration mode: | | No | |
| Accepted when Ready is low: | | Yes | |
| Status of Ready signal during processing: | | No change | |
| Supported interfaces: | | Telegrams: Availability and supported interfaces (Page 91) | |

Teach detector (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Teach detector (TED) Request string to sensor (BINARY) | | | |
|--|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x08 | Telegram length |
| 5 | Unsigned Char | 0x18 | Teach detector |
| 6 | Unsigned Char | 0xXX | 0x00: Alignment
≥ 0x01: Detector selection |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary
0x01: Permanent |
| 8 | Unsigned Char | 0xXX | 0x00: No trigger, teach-in with next image acquisition
0x01: Trigger is executed for teach-in |
| Teach detector (TED) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x00 | Telegram length |
| 5 | Unsigned Char | 0x18 | Teach detector |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Set trigger delay (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Set trigger delay (STD) Request string to sensor (BINARY) | | | |
|---|----------------|---------|--|
| Byte no.. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x08 | Telegram length |
| 5 | Unsigned Char | 0x27 | Set trigger delay |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary
0x01: Permanent |
| 8 - 11 | Unsigned int | 0xXX | Trigger delay
in msec (max. 3000 msec)
in encoder steps (max. 65535 steps) |
| Set trigger delay (STD) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x27 | Set trigger delay |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Get trigger delay (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Get trigger delay (GTD) Request string to sensor (BINARY) | | | |
|---|----------------|---------|--|
| Byte no.. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x06 | Telegram length |
| 5 | Unsigned Char | 0x28 | Get trigger delay |
| 6 | Unsigned Char | 0xX1 | Request version |
| Get trigger delay (GTD) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0B | Telegram length |
| 5 | Unsigned Char | 0x28 | Get trigger delay |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | Unsigned int | 0xXX | Trigger delay
in msec (max. 3000 msec)
in encoder steps (max. 65535 steps) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Save job permanently (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Save Job Permanently (SJP) Request string to sensor (BINARY) | | | |
|--|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x0D | Saving of all telegrams that were previously executed temporarily |
| Save Job Permanently (SJP) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x0B | Telegram length |
| 5 | Unsigned Char | 0x0D | Save job permanently |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

11.5.4 Calibration

Calibration: Initialization (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Initialization (CCD) Request string to sensor (BINARY) | | | |
|---|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x1F | Initialize (Calibration: Clear Data) |
| Calibration: Initialization (CCD) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x1F | Initialize (Calibration: Clear Data) |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Calibration: Add world point (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Add world point (CAW) Request string to sensor (BINARY) | | | |
|--|----------------|---------|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x15 | Telegram length |
| 5 | Unsigned Char | 0x26 | Calibration: Add world point |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x01: Fiducials only Calibration plate (Robotics)
0x04: World point and pixel Point pair list (Robotics) |
| 9 - 10 | Unsigned Short | 0x00 | Constant (2 bytes) |
| 11 - 14 | Unsigned int | 0xXX | World X (in mm *1000) |
| 15 - 18 | Unsigned int | 0xXX | World Y (in mm *1000) |
| 19 - 22 | Unsigned Char | 0x00 | Constant (4 bytes) |
| Calibration: Add world point (CAW) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x11 | Telegram length |
| 5 | Unsigned Char | 0x26 | Calibration: Add world point |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 9 | Unsigned Short | 0xXX | Current number of points |
| 10 - 13 | Unsigned int | 0xXX | Image point X |
| 14 - 17 | Unsigned int | 0xXX | Image point Y |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |

| | |
|-----------------------|---|
| Supported interfaces: | Telegram: Availability and supported interfaces (Page 91) |
|-----------------------|---|

Note: For the CAW request, the overall job result must be positive.

Calibration: Point pair list (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Point pair list (CCL) Request string to sensor (BINARY) | | | |
|--|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x06 | Telegram length |
| 5 | Unsigned Char | 0x1E | Calibration: Point pair list |
| 6 | Unsigned Char | 0xXX | 0x00: Temporary
0x01: Permanent |
| Calibration: Point pair list (CCL) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x19 | Telegram length |
| 5 | Unsigned Char | 0x1E | Calibration: Point pair list |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 9 | Unsigned Short | 0xXX | Current highest point pair index |
| 10 - 13 | Unsigned int | 0xXX | Deviation calibration, RMSE |
| 14 - 17 | Unsigned int | 0xXX | Deviation calibration, mean |
| 18 - 21 | Unsigned int | 0xXX | Deviation calibration, max. |
| 22 - 25 | Unsigned int | 0xXX | Deviation calibration, min. |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Calibration: Validate point pair list (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Validate point pair list (CVL) Request string to sensor (BINARY) | | | |
|---|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x05 | Telegram length |
| 5 | Unsigned Char | 0x20 | Calibration: Validate point pair list |
| Calibration: Validate point pair list (CVL) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x19 | Telegram length |
| 5 | Unsigned Char | 0x20 | Calibration: Validate point pair list |
| 6 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 9 | Unsigned Short | 0xXX | Current highest point pair index |
| 10 - 13 | Unsigned int | 0xXX | Deviation calibration, RMSE |
| 14 - 17 | Unsigned int | 0xXX | Deviation calibration, mean |
| 18 - 21 | Unsigned int | 0xXX | Deviation calibration, max. |
| 22 - 25 | Unsigned int | 0xXX | Deviation calibration, min. |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Calibration: Calibration plate (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Calibration Plate (CCP) Request string to sensor (BINARY) | | | |
|--|----------------|---------|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x09 | Telegram length |
| 5 | Unsigned Char | 0x24 | Calibration: Calibration plate |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary
0x01: Permanent |
| 8 | Unsigned Char | 0xXX | 0x00: No fiducials are used. Origin of Measuring coordinate system identical to origin of Calibration Plate Coordinate System.
0x01: No fiducials are used. Measuring coordinate system identical with Camera coordinate system.
0x02: Uses world system, fiducial Job
0x03: Uses world system, fiducial Command CAW |
| 9 | Unsigned Char | 0xXX | 0x00: Calibration internal and external sensor parameters
0x01: Validation of calibration
0x02: Calibration internal sensor parameters
0x05: Calibration Transformation Measuring coordinate system |
| Calibration: Calibration Plate (CCP) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x3D | Telegram length |
| 5 | Unsigned Char | 0x24 | Calibration: Calibration plate |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 9 | Unsigned Short | 0xXX | Number of currently detected calibration points |
| 10 - 13 | Unsigned int | 0xXX | Deviation calibration, RMSE |
| 14 - 17 | Unsigned int | 0xXX | Deviation calibration, mean |
| 18 - 21 | Unsigned int | 0xXX | Deviation calibration, max. |

| | | | |
|---|--------------|------|--|
| 22 - 25 | Unsigned int | 0xXX | Deviation calibration, min. |
| 26 - 29 | Unsigned int | 0xXX | CPF_MF X (in user unit * 1000) |
| 30 - 33 | Unsigned int | 0xXX | CPF_MF Y (in user unit * 1000) |
| 34 - 37 | Unsigned int | 0x00 | CPF_MF Z (in user unit * 1000) |
| 38 - 41 | Unsigned int | 0x00 | CPF_MF Angle X (in degrees * 1000) |
| 42 - 45 | Unsigned int | 0x00 | CPF_MF Angle Y (in degrees * 1000) |
| 46 - 49 | Unsigned int | 0xXX | CPF_MF Angle Z (in degrees * 1000) |
| 50 - 53 | Unsigned int | 0xXX | Deviation fiducials, mean |
| 54 - 57 | Unsigned int | 0xXX | Deviation fiducials, max. |
| 58 - 61 | Unsigned int | 0xXX | Deviation fiducials, min. |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Calibration: Set fiducial (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Set fiducial (CSF) Request string to sensor (BINARY) | | | |
|---|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x2B | Calibration: Set fiducial |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary
0x01: Permanent |
| Calibration: Set fiducial (CSF) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x2B | Telegram length |
| 5 | Unsigned Char | 0x2B | Calibration: Set fiducial |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 - 11 | Unsigned int | 0xXX | X value |
| 12 - 15 | Unsigned int | 0xXX | Y value |
| 16 - 19 | Unsigned int | 0xXX | Z value |
| 20 - 23 | Unsigned int | 0xXX | Angle X value |
| 24 - 27 | Unsigned int | 0xXX | Angle Y value |
| 28 - 31 | Unsigned int | 0xXX | Angle Z value |
| 32 - 35 | Unsigned int | 0xXX | Deviation fiducials, mean |
| 36 - 39 | Unsigned int | 0xXX | Deviation fiducials, max. |
| 40 - 43 | Unsigned int | 0xXX | Deviation fiducials, min. |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Calibration: Add image (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Add image (CAI) Request string to sensor (BINARY) | | | |
|---|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x22 | Telegram length
34 (0x22) Bytes |
| 5 | Unsigned Char | 0x34 | Calibration: Add image |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | Mode
0x01: Multi-image calibration
0x02: Hand-Eye calibration (Robotics)
0x03: Base-Eye calibration (Robotics) |
| 8 | Unsigned Short | 0x00 | Constant |
| 9 | Unsigned Char | 0xXX | Define Measurement plane
0x00: Do not use image to define Measurement plane
0x01: Use image to define Measurement plane |
| 10 | Unsigned Char | 0xXX | "Robot: Order of rotation"
0x00: Use order of rotation specified in job
0x01: Yaw-Pitch-Roll (e.g. Stäubli)
0x02: Roll-Pitch-Yaw (e.g. Kuka, Fanuc, Hanwha, ABB**, UR**)
** when using the corresponding conversion function |
| 11-14 | Unsigned Char | | Pose_TCP Pos. X
(in user unit * 1000) |
| 15-18 | Unsigned Char | | Pose_TCP Pos. Y
(in user unit * 1000) |
| 19-22 | Unsigned Char | | Pose_TCP Pos. Z
(in user unit * 1000) |
| 23-26 | Unsigned Char | | Pose_TCP Angle X
(in degrees * 1000) |
| 27-30 | Unsigned Char | | Pose_TCP Angle Y
(in degrees * 1000) |

| | | | |
|--|----------------|---------|---|
| 31-34 | Unsigned Char | | Pose_TCP Angle Z
(in degrees * 1000) |
| Calibration: Add image (CAI) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1-4 | Unsigned int | 0x0A | Telegram length |
| 5 | Unsigned Char | 0x34 | Calibration: Add image |
| 6-7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Short | 0xXX | Current number of images in list |
| 9-10 | Unsigned Char | 0xXX | Total number of detected points |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | Yes |
| Accepted when Ready is low: | | | No |

Calibration: Multi-image (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Multi-image (CMP) Request string to sensor (BINARY) | | | |
|---|---------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x09 | Telegram length 9 (0x09) Bytes |
| 5 | Unsigned Char | 0x35 | Calibration: Multi-image |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary
0x01: Permanent |
| 8 | Unsigned Char | 0xXX | Origin of the world coordinate system:
0x00: World coordinate system identical with the Calibration Plate Coordinate System (center of the plate).
0x01: Origin of World coordinate system so that it is identical to origin of Image Coordinate System (upper left pixel).
0x02: (only for Calibration plate (Robotics)) Use World coordinate system of fiducials, as specified in the job file.
0x03: (only for Calibration plate (Robotics)) Use World coordinate system of fiducials as set in request CAW. |
| 9 | Unsigned Char | 0xXX | Mode
0x00: Calibration (internal and external parameters)
0x01: Validieren (vorhandene Kalibrierung verwenden; mindestens ein Kalibrierpunkt wird hinzugefügt. Über Rückprojektion kann zurückgeschlossen werden, ob der Punkt zur aktuellen Kalibrierung passt, oder verschoben ist)
0x02: Calibration (internal parameters only)
0x03: Calibration (external parameters only using new internal parameters)
0x04: Calibration (external parameters only)
0x05: Calibrate Measurement plane only (CPF_MF) |

| Calibration: Multi-image (CMP) Response string from sensor (BINARY) | | | |
|---|----------------|---------|---|
| Byte no. | Data type | Content | Meaning |
| 1-4 | Unsigned int | 0x1D | Telegram length 29 (0x1D) Bytes |
| 5 | Unsigned Char | 0x35 | Calibration: Multi-image |
| 6-7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Field of view coverage (%)
0x00: no coverage
0x64: Coverage 100% |
| 9-10 | Unsigned Short | 0xXX | Total number of detected points |
| 11 | Unsigned Char | 0xXX | Number of images used |
| 12 | Unsigned Char | 0xXX | Number of invalid images |
| 13 | Unsigned Char | 0xXX | Sufficient tilt between calibration plate poses
0x00: not sufficient
0x01: sufficient |
| 14-17 | Unsigned int | 0xXX | Deviation calibration plate RMSE [px] |
| 18-21 | Unsigned int | 0xXX | Deviation calibration plate Max. [px] |
| 22-25 | Unsigned int | 0xXX | Deviation fiducials, RMSE (in user unit * 1000) |
| 26-29 | Unsigned int | 0xXX | Deviation fiducials, max. [px] |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |

Calibration: Robotics multi-image (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Robot multi-picture (CRP) Request string to sensor (BINARY) | | | |
|--|---------------|---------|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x09 | Telegram length (bytes)
9 Byte |
| 5 | Unsigned Char | 0x36 | Calibration: Calibration plate robotics |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary
0x01: Permanent |
| 8 | Unsigned Char | 0xXX | Origin of the world coordinate system:
0x04: Set origin of coordinate system equal to Robot Coordinate System |
| 9 | Unsigned Char | X | Mode
0x00: Calibration (internal and external parameters)
0x01: Validieren (vorhandene Kalibrierung verwenden; mindestens ein Kalibrierpunkt wird hinzugefügt. Über Rückprojektion kann zurückgeschlossen werden, ob der Punkt zur aktuellen Kalibrierung passt, oder verschoben ist)
0x02: Calibration (internal parameters only)
0x03: Calibration (external parameters only using new internal parameters)
0x04: Calibration (external parameters only)
0x05: Calibrate Measurement plane only (CPF_MF)
0x06: Hand-Eye calibration (Robotics) / Base-Eye calibration (Robotics) |
| Calibration: Robot multi-picture (CRP) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1-4 | Unsigned int | 0x2C | Telegram length 44 (0x2C) Bytes |
| 5 | Unsigned Char | 0x36 | Calibration: Calibration plate robotics |

| | | | |
|---------------------------------|----------------|------|--|
| 6-7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Field of view coverage
0x00: not sufficient
0x01: sufficient |
| 9-10 | Unsigned Short | 0xXX | Total number of detected points |
| 11 | Unsigned Char | 0xXX | Number of images used |
| 12 | Unsigned Char | 0xXX | Number of invalid images |
| 13-16 | Unsigned int | 0xXX | Deviation calibration plate RMSE [px] |
| 17-20 | Unsigned int | 0xXX | Deviation calibration plate Max. [px] |
| 21-24 | Unsigned int | 0xXX | Deviations calibration plate pose Translation RMSE (in user unit * 1000) |
| 25-28 | Unsigned int | 0xXX | Deviations calibration plate pose Translation Max. (in user unit * 1000) |
| 29-32 | Unsigned int | 0xXX | Deviations calibration plate pose Rotation RMSE (in degrees * 1000) |
| 33-36 | Unsigned int | 0xXX | Deviations calibration plate pose Rotation Max. (in degrees * 1000) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |

Calibration: Copy Calibration (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Copy calibration (CCC) Request string to sensor (BINARY) | | | |
|---|----------------|---------|---|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x09 | Telegram length |
| 5 | Unsigned Char | 0x25 | Calibration: Copy Calibration |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0x01 | Constant |
| 8 | Unsigned Char | 0xXX | Destination
0 : Copy to all jobs
>0: Copy to specified job |
| 9 | Unsigned Char | 0xXX | 0: Always copy when the calibration is active.
1: Only copy if the calibration method is the same. |
| Calibration: Copy calibration (CCC) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x08 | Telegram length |
| 5 | Unsigned Char | 0x25 | Calibration: Copy Calibration |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | 00: Successful
>0 : Job number at which the error occurs. |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Calibration: Set parameter (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Set parameter (CSP) Request string to sensor (BINARY) | | | |
|--|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length in Byte, 16 Bytes (0x13) + length of selected parameter |
| 5 | Unsigned Char | 0x29 | Calibration: Set parameter |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | 0x00: Temporary
0x01: Permanent |
| 8 | Unsigned Char | 0xXX | Parameter number, see table Calibration parameters for telegrams CSP and CGP |
| 9 - 12 | Unsigned int | 0xXX | Length of the following data |
| 13 ... n | Unsigned Char | 0xXX | Parameter value, see table Calibration parameters for telegrams CSP and CGP |
| Calibration: Set parameter (CSP) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x29 | Calibration: Set parameter |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Calibration parameters: see table [Calibration parameters for telegrams CSP and CGP](#)

Calibration: Read parameter (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Calibration: Read parameter (CGP) Request string to sensor (BINARY) | | | |
|---|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x07 | Telegram length |
| 5 | Unsigned Char | 0x2A | Calibration: Read parameter |
| 6 | Unsigned Char | 0x01 | Request version |
| 7 | Unsigned Char | 0xXX | Parameter number (Page 241) |
| Calibration: Read parameter (CGP) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length in bytes, 12 bytes (0x0C) + length of selected parameter |
| 5 | Unsigned Char | 0x2A | Calibration: Read parameter |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Parameter number (Page 241) |
| 9 - 12 | Unsigned int | 0xXX | Length of the following data |
| 13 ... n | Unsigned Char | 0xXX | Parameter value (Page 241) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | No change |
| Supported interfaces: | | | Telegrams: Availability and supported interfaces (Page 91) |

Calibration parameters for telegrams CSP and CGP

| Parameter description | Parameter number | Parameter value | Length | Calibration status after CSP |
|---|------------------|---|---------------------------------|------------------------------|
| Status calibration | 0x01 | 0x00: Invalid
0x01: Valid | 1 byte | —* |
| Selection of calibration method | 0x02 | 0x00: None
0x02: Point pair list (Robotics)
0x03: Calibration plate (Measurement)
0x04: Calibration plate (Robotics)
0x05: Hand-Eye calibration (Robotics)
0x06: Base-Eye calibration (Robotics) | 1 byte | invalid |
| User unit | 0x04 | 0x00: Millimeter [mm]
0x01: Centimeter [cm]
0x02: Meter [m]
0x03: Inch ["]
0x04: Arbitrary unit [au] | 1 byte | no change |
| Internal parameters | 0x0A | Focal length (in mm *1000)
Kappa (*1000)
Pixel pitch X (in μm * 1000)
Pixel pitch Y (in μm * 1000)
Coordinate origin X (in pixels * 1000)
Coordinate origin Y (in pixels * 1000)
Image size X (number of pixels)
Image size Y (number of pixels) | 0x20
(8 * 4 bytes per value) | —* |
| Reference Camera- to Measuring coordinate system (CF_MF) | 0x0B | Translation X, Y, Z (in user unit * 1000)
Angle X, Y, Z (in degrees * 1000) | 0x18
(6 * 4 bytes per value) | —* |
| Reference Camera- to Calibration Plate Coordinate System (CF_CPF) | 0x0C | Translation X, Y, Z (in user unit * 1000)
Angle X, Y, Z (in degrees * 1000) | 0x18
(6 * 4 bytes per value) | —* |

| Parameter description | Parameter number | Parameter value | Length | Calibration status after CSP |
|--|------------------|--|---------------------------------|--------------------------------|
| Reference Robot- to Camera coordinate system (RF_CF) | 0x0D | Translation X, Y, Z (in user unit * 1000)
Angle X, Y, Z (in degrees * 1000) | 0x18
(6 * 4 bytes per value) | —* |
| Reference Calibration plate- to Measuring coordinate system (CPF_MF) | 0x0E | Translation X, Y, Z (in user unit * 1000)
Angle X, Y, Z (in degrees * 1000) | 0x18
(6 * 4 bytes per value) | —* |
| Reference Robot- to Measuring coordinate system (RF_MF) | 0x0F | Translation X, Y, Z (in user unit * 1000)
Angle X, Y, Z (in degrees * 1000) | 0x18
(6 * 4 bytes per value) | —* |
| Reference TCP- to Camera coordinate system(TCP_CF) | 0x10 | Translation X, Y, Z (in user unit * 1000)
Angle X, Y, Z (in degrees * 1000) | 0x18
(6 * 4 bytes per value) | —* |
| Reference robot- to TCP coordinate system (RF_TCP) | 0x11 | Translation X, Y, Z (in user unit * 1000)
Angle X, Y, Z (in degrees * 1000) | 0x18
(6 * 4 bytes per value) | no change |
| Z-shift Measurement plane | 0x15 | (in user unit * 1000) | 4 bytes | no change |
| Focal length in [mm] | 0x16 | [mm * 1000] | 4 bytes | invalid (CSP for C-Mount only) |
| Calibration plate type | 0x17 | Character string with name of the description file | n | invalid |
| Fiducial 1 | 0x18 | Translation X, Y, Z (in user unit * 1000) | 0x0C
(3* 4 bytes per value) | invalid |
| Fiducial 2 | 0x19 | | | |
| Fiducial 3 | 0x1A | | | |
| Fiducial 4 | 0x1B | | | |

| Parameter description | Parameter number | Parameter value | Length | Calibration status after CSP |
|--|------------------|--|---------------------------------------|------------------------------|
| Number of existing calibration plate types | 0x25 | Request - Selection of type:
0x00: All
0x01: Measurement
0x02: Robotics
Response:
Number of plates | Request: 1
Response:
2 | —* |
| Available calibration plate types (file names) | 0x26 | Request - Selection of type:
0x00: All
0x01: Measurement
0x02: Robotics
Request - Index:
0: All file names
>0: Index selection
Response:
File names of Calibration plates | Request: 1
Response:
5 (String) | —* |
| Robot: Order of rotation | 0x27 | "Robot: Order of rotation"
0x00: Use order of rotation specified in job
0x01: Yaw-Pitch-Roll (e.g. Stäubli)
0x02: Roll-Pitch-Yaw (e.g. Kuka, Fanuc, Hanwha, ABB**, UR**)
** when using the corresponding conversion function | 1 byte | invalid |
| Average sensor resolution | 0x29 | Value (in user unit/pixel * 1000) | 4 bytes | —* |

* CSP not possible (parameter read-only, cannot be set).

11.5.5 Visualization

Get image (BINARY)

[Telegrams: Availability and supported interfaces \(Page 91\)](#)

[Overview telegrams \(Page 87\)](#)

| Get image (GIM) Request string to sensor (BINARY) | | | |
|---|----------------|---------|--|
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0x06 | Telegram length |
| 5 | Unsigned Char | 0x03 | Get image |
| 6 | Unsigned Char | 0xXX | 0x00: Last image
0x01: Last failed image
0x02: Last good image |
| Get image (GIM) Response string from sensor (BINARY) | | | |
| Byte no. | Data type | Content | Meaning |
| 1 - 4 | Unsigned int | 0xXX | Telegram length in bytes, 13 bytes (0x0D) + number of bytes depending on the image format
e.g. 00 04 B0 0D (Dez. 307213) |
| 5 | Unsigned Char | 0x03 | Get image |
| 6 - 7 | Unsigned Short | 0xXX | Error codes (Page 94) |
| 8 | Unsigned Char | 0xXX | Image type
0: Grayscale
3: Bayer Pattern_BG
When converting the color image from Bayer into RGB, the appropriate image type must be considered. |
| 9 | Unsigned Char | 0xXX | Image result
00: Failed image
01: Good image |
| 10 - 11 | Unsigned Short | 0xXX | Number of rows
e.g. 01 E0 = 480 |

| | | | |
|---|----------------|------|---|
| 12 - 13 | Unsigned Short | 0xXX | Number of columns
e.g. 02 80 = 640 |
| 14 ... n | Unsigned Char | 0xXX | Binary image data (rows * columns) |
| Additional information: | | | |
| Accepted in run mode: | | | Yes |
| Accepted in configuration mode: | | | No |
| Accepted when Ready is low: | | | Yes |
| Status of Ready signal during processing: | | | Low |
| Supported interfaces: | | | Telegram: Availability and supported interfaces (Page 91) |

11.5.6 Data output BINARY

Output data (BINARY), dynamically composed according to user settings in the software under: SensoConfig / Output / Telegram.

Basic string structure:

<START> (((<OPTIONAL FIELDS> <PAYLOAD>))) <CHKSUM> <TRAILER>



NOTE:

The length and data types of the payload are standard values. The factor and bit depth can be set via "Telegram" / "Payload".

Output data (BINARY):

| <OPTIONAL FIELDS> | | | | |
|-------------------|---|----------------------|--|---------------|
| Parameter | Description | Length BINARY [Byte] | Data type | Available for |
| Selected fields | With this checkbox all selected fields are displayed. The checkbox "Selected fields" itself is not displayed. | 2 | The output sequence is from left to right and from top to bottom, i.e. one bit is set per active checkbox, starting with the lowest-value one. | All types |
| Telegram length | Number of characters including the characters for the telegram length itself. | 2 | Unsigned Short | All types |
| Status byte | Returns the Trigger mode. | 2 | 0x06 0x00 = Trigger;
0x05 0x00 = Free run | All types |

| <OPTIONAL FIELDS> | | | | |
|--------------------------------|---|-------------------------------------|--|----------------------|
| Parameter | Description | Length
BINARY
[Byte] | Data type | Available for |
| Detector results | Output of overall result for each detector.
Byte 1
Bit 1 (LSB) = Global job result (1 = Pass, 0 = Fail)
Bit 2 = Boolean result
Alignment only,
Alignment inactive = True | 3 ... 35 | | All types |
| Digital outputs | Returns the logic gate result for each digital output. | N | Bytes 1 and 2: Number of active Outputs
Bytes 3 – n: Outputs, bit-coded | All types |
| log. Outputs | Returns the logic gate result for each logic output. | N | Byte 1 and byte 2: Number of active log. Outputs
Byte 3 – n All active logic outputs, | All types |
| Execution time | Returns the execution time for the last evaluation. | 4 | Signed integer | All types |
| Active job | Returns the job for the last evaluation. | 1 | Unsigned int U8 | All types |






<PAYLOAD>
Overview of detector-specific payload - Values
GENERAL












| <PAYLOAD> General | | | | |
|--------------------------------|--|-------------------------------------|------------------|----------------------|
| Value | Description | Length
BINARY
[Byte] | Data type | Available for |
| "All evaluations" counter | Total number of checks | 4 | Signed integer | GENERAL |
| Pass parts counter | Number of inspections with result "OK" | 4 | Signed integer | GENERAL |
| Fail parts counter | Number of inspections with result "Error" | 4 | Signed integer | GENERAL |
| Timeout | Indicates that the maximum cycle time has been exceeded. | 1 | BOOL | GENERAL |
| Recording | Indicates the number of image acquisition repetitions for the last evaluation
Only in combination with repeat mode. | 4 | INT | GENERAL |
| String length | This field can be used to enter a constant string into the data output. | 0 ... 5 | STRING | GENERAL |












Base values

| <PAYLOAD> Base values | | | | |
|-----------------------|--|----------------------|----------------|---------------|
| Value | Description | Length BINARY [Byte] | Data type | Available for |
| Score | [%] | 4 | Signed integer | All detectors |
| Overall result | Boolean detector result | 1 | BOOL | All detectors |
| Execution time | Execution time of individual detector in [msec]. | 4 | Signed integer | All detectors |







Position

| <PAYLOAD> Position / location | | | | |
|-------------------------------|---|----------------------|----------------|--|
| Value | Description | Length BINARY [Byte] | Data type | Available for |
| Pos. X | X coordinate for the found position, 1/1000 [user unit] | 4 | Signed integer |  |
| Pos. Y | Y coordinate for the found position, 1/1000 [user unit] | 4 | Signed integer |  |
| Pos. Z | Z coordinate of the found position, 1/1000 [user unit] | | Signed integer | 
With Result offset:
 |
| Delta Pos. X | X position delta between the taught object and the found object, 1/1000 [user unit] | 4 | Signed integer |  |


| <PAYLOAD> Position / location | | | | |
|--|--|-----------------------------|------------------|---|
| Value | Description | Length BINARY [Byte] | Data type | Available for |
| Delta Pos. Y | Y position delta between the taught object and the found object, 1/1000 [user unit] | 4 | Signed integer |  |
| Delta Pos. Z | Z position delta between the taught object and the found object, 1/1000 [user unit] | 4 | Signed integer | 
With Result offset:
 |
| Angle X | Orientation of the found object, relative to the X-axis, 1/1000 [°] | 4 | Signed integer | 
With Result offset:
 |
| Angle Y | Orientation of the found object, relative to the Y-axis, 1/1000 [°] | 4 | Signed integer | 
With Result offset:
 |
| Angle Z | Orientation of the found object, relative to the Z-axis, 1/1000 [°] | 4 | Signed integer | 
 |
| Angle (45) | Orientation of bounding box for found code [°], value range: -45° to 45° | 4 | Signed integer |  |
| Angle (180) | Orientation of object width (long axis) [°], Value range: -90° ... +90°
0° = East, counterclockwise | 4 | Signed integer |  |










| <PAYLOAD> Position / location | | | | |
|--|---|-----------------------------|------------------|---|
| Value | Description | Length BINARY [Byte] | Data type | Available for |
| Angle (360) | Orientation of object width (long axis) [°], Value range -180° ... +180°. 0° = East, counterclockwise | 4 | Signed integer |  |
| Delta Angle X | Angle between taught object and found object, 1/1000 [°] | 4 | Signed integer | 
With Result offset:
 |
| Delta Angle Y | Angle between taught object and found object, 1/1000 [°] | 4 | Signed integer | 
With Result offset:
 |
| Delta Angle Z | Angle between taught object and found object, 1/1000 [°] | 4 | Signed integer |  |
| Pose 3D (X, Y, Z, Angle X, Angle Y, Angle Z) | Coordinates of the found object, 1/1000 [user unit] Angle: 1/1000 degrees | 4 bytes per value each | Signed integer | 
With Result offset:
 |
| Delta Pose 3D (X, Y, Z, Angle X, Angle Y, Angle Z) | Delta coordinates of the found object, 1/1000 [user unit] Angle: 1/1000 degrees | 4 bytes per value each | Signed integer | 
With Result offset:
 |
| Position control | | 1 | BOOL |  |

Measurement



| <PAYLOAD> Measurement | | | | |
|------------------------------------|---|-----------------------------|------------------|---|
| Value | Description | Length BINARY [Byte] | Data type | Available for |
| Height | Height of geometric element [user unit],
Height ≥ 0 , height \leq width | 4 | Signed integer |  |
| Width | Width of geometric element [user unit]
Width ≥ 0 , width \geq height | 4 | Signed integer |  |
| Radius | Radius of fitted circle [user unit] | 4 | Signed integer |  |
| Area | Area of BLOB without holes, 1/1000 [pixels] | 4 | Signed integer |  |
| Area (incl. holes) | Area of BLOB including holes, 1/1000 [pixels] | 4 | Signed integer |  |
| Distance | Calculated distance [user unit] | 4 | Signed integer |  |

Identification



| <PAYLOAD> Identification | | | | |
|---------------------------------------|-----------------------------|-----------------------------|------------------|---|
| Value | Description | Length BINARY [Byte] | Data type | Available for |
| String length | Length of read code [bytes] | 4 | Signed integer |  |

| <PAYLOAD> Identification | | | | |
|---------------------------------------|--|-------------------------------------|------------------|--|
| Value | Description | Length
BINARY
[Byte] | Data type | Available for |
| String length | Content of the read code.
Depending on the code, the string length may vary. If a fixed string length is desired, the minimum string length (detector-specific payload) and the maximum string length (detector settings) must be set to the same value (e.g. 127). | N | STRING |    A |
| String comparison | Content check for the read information.
The content of the read information is checked on the basis of regular expressions (see detector Data-code, Reference string tab) | 1 | BOOL |    A |
| Truncated | Code complete or truncated
0: Code complete
1: Code truncated | 1 | BOOL |    A |





Identification - quality







| <PAYLOAD> Identification - Quality | | | | |
|---|---|--|--|---|
| Value | Description | Length
BINARY
[Byte] | Data type | Available for |
| Quality - overall | Output of all Q parameters. Depending on the selected code type and standard. | 1 byte per value; separated by specified separator
For 2D code parameter Q9 (mean light): 1...3 | Unsigned Char;
for 2D Code Q9 (Meanlight)
Unsigned Short |  |
| Quality - individual | Output of individual quality values: Selection Q1-Q24 depending on the selected code type and standard.
Numbers: 1-4
Letters: A-F | 1 | Unsigned Char;
for 2D Code Q9 (Meanlight)
Unsigned Short |  |
| Min. Quality | Used to check whether the minimum required quality is being met | 4 | Unsigned int | A |

Color



| <PAYLOAD> Color | | | | |
|--|--|----------------------|----------------|---|
| Value | Description | Length BINARY [Byte] | Data type | Available for |
| Color value: <ul style="list-style-type: none"> • Red, green, blue • Hue, saturation, lightness • Luminance, a, b | Value for color parameter | 4 | Signed integer |  |
| Color distance | Distance of the current color versus the taught-in color | 4 | Signed integer |  |






Counting / number








| <PAYLOAD> Counting / number | | | | |
|-----------------------------|--|----------------------|----------------|---|
| Value | Description | Length BINARY [Byte] | Data type | Available for |
| Number of objects | Number of objects found [units] | 4 | Signed integer |  |
| Number of valid objects | Number of valid objects found [units] | 4 | Signed integer |  |
| Number of search stripes | Number of parallel search stripes into which the width of the search range is divided. [units] | 4 | Signed integer |  (Edge detector only)
 |






| <PAYLOAD> Counting / number | | | | |
|--|---|-----------------------------|------------------|---|
| Value | Description | Length BINARY [Byte] | Data type | Available for |
| Number of valid search stripes | Used to check whether the number of search stripes found falls within a specific range. [Good/Bad or units] | 4 | Signed integer |  (Edge detector only)
 |
| Result vector | Vector containing the result (1/0) of the instances found | N | BOOL |    |
| Too many BLOBs | | 1 | BOOL |  |

Extended

| <PAYLOAD> Extended | | | | |
|---------------------------------|---|-----------------------------|------------------|--|
| Value | Description | Length BINARY [Byte] | Data type | Available for |
| Scaling | Outputs the scaling range, 1/1000. Within the scaling range, scaled-up or scaled-down objects will be detected. Value range of 0.5 to 2 | 4 | Signed integer |  (Contour matching only)
 |
| Eccentricity | Numerical eccentricity
Value range of 0.0 to 1.0 | 4 | Signed integer | |

| <PAYLOAD> Extended | | | | |
|---------------------------------|--|-------------------------------------|------------------|---|
| Value | Description | Length
BINARY
[Byte] | Data type | Available for |
| Security | Output of the security values of the individual characters.
The reliability value specifies how reliably the reader was able to interpret a character.
Value range of 0 to 100 [%] | 4 | Signed integer | A |
| Reference string met | The output string matches the reference string. | 1 | BOOL | A |
| contrast | Code contrast
Value range of 0 to 100 [%] | 4 | Signed integer |  |
| Correction | Number of modules corrected by error corrections [units] | 4 | Signed integer |  |
| Contour length | Number of pixels of outer contour,
1/1000 [pixels] | 4 | Signed integer |  |
| Compactness | BLOB compactness (circle =1; other > 1).
The more the shape of the BLOB deviates from a circle, the greater the compactness value will be. | 4 | Signed integer |  |
| Center of gravity X | X coordinate of centroid,
1/1000 | 4 | Signed integer |  |

| <PAYLOAD> Extended | | | | |
|---------------------------------|---|-------------------------------------|------------------|---|
| Value | Description | Length
BINARY
[Byte] | Data type | Available for |
| Center of gravity Y | Y coordinate of centroid, 1/1000 | 4 | Signed integer |  |
| Gray scale value, average | Average gray scale value of all the pixels that belong to the BLOB. | 4 | Signed integer |  |
| Min. threshold | Lower threshold for the binarization of the objects. 0...255 | 4 | Signed integer |  |
| Max. threshold | Upper threshold for the binarization of the objects. 0...255 | 4 | Signed integer |  |
| Inverted threshold | Specifies whether the range Min <-> Max is inverted.
P: inverted
F: not inverted | 1 | Unsigned Char |  |
| Deviation, inside | Returns the largest deviation between the BLOB contour and the contour of the geometric element (deviation inside the fitted circle).
[user unit] | 4 | Signed integer |  |
| Deviation, outside | Returns the largest deviation between the BLOB contour and the contour of the geometric element (deviation outside the fitted circle).
[user unit] | 4 | Signed integer |  |

| <PAYLOAD> Extended | | | | |
|---------------------------------|---|-----------------------------|------------------|---|
| Value | Description | Length BINARY [Byte] | Data type | Available for |
| Deviation, mean | Returns the mean of the absolute "inside" and "outside" deviation values between the BLOB contour and the contour of the geometric element. | 4 | Signed integer |  |
| Axial ratio | Ratio of the long to the short axis (a / b) | 4 | Signed integer |  |
| Face up / down, area | Face up / down position, based on: area, position indicated by sign, 1/1000 | 4 | Signed integer |  |
| Result index | List index | 4 | Signed integer |  |
| Search stripe distance | Calculated distance [user unit] / 1000 per pair of search stripes | 4 | Signed integer |  |

| <CHKSUM> | | | | |
|-----------------------|--|-----------------------------|------------------|----------------------|
| Parameter | Description | Length BINARY [Byte] | Data type | Available for |
| Check sum | XOR check sum of all bytes in the telegram. Is transmitted as the last byte. | 1 | Unsigned int | All types |

| <TRAILER> | | | | |
|------------------------|--|-------------------------------------|------------------|----------------------|
| Parameter | Description | Length
BINARY
[Byte] | Data type | Available for |
| Start | Characters appended at the end of the string | 0 ... 8 | Unsigned int | All types |


NOTE:

If no calibration has been performed, all values refer to pixels.

All detector-specific data with decimal places is transmitted as integers (multiplied by 1000) and must accordingly be divided by 1000 after the data is received. The values are transferred in the format "Big-endian".

Example: "Score" value (BINARY protocol)

In SensoConfig/SensoView "Score" = 35 is displayed.

Via Ethernet, the following four bytes, for example, are received: 000,000,139,115

Formula for conversion: $(\text{Byte4} * 256 + \text{Byte3}) * 65536 + \text{Byte2} * 256 + \text{Byte1} = \text{Value}$

Because big-endian (from the sensor) is sent, the following applies:

000 = HiWordByte, 000 = HiLowByte, 139 = HiByte, 115 = LoByte

$(0 * 256 + 0) * 65536 + (139 * 256) + 115 = 35699 / 1000 = 35.699 (= \text{real score value})$

Angle data or other negative values are represented in two's complement.

We look ahead
Yesterday, today and in the future



Germany
SensoPart Industriesensorik GmbH
79288 Gottenheim
Tel.: +49 7665 94769-0
info@sensopart.de

France
SensoPart France SARL
F-77420 Champs sur Marne
Tel.: +33 1 64 73 00 61
info@sensopart.fr

Great Britain
SensoPart UK Ltd.
Melton Mowbray, Leicestershire, LE13 0PB
Tel.: +44 1664 561539
uk@sensopart.com

USA
SensoPart Inc.
Perrysburg, OH 43551
Tel.: +1 866 282 7610
usa@sensopart.com

China
SensoPart (Shanghai)
Shanghai, 201803
Tel.: +86 216 901 7660
china@sensopart.cn