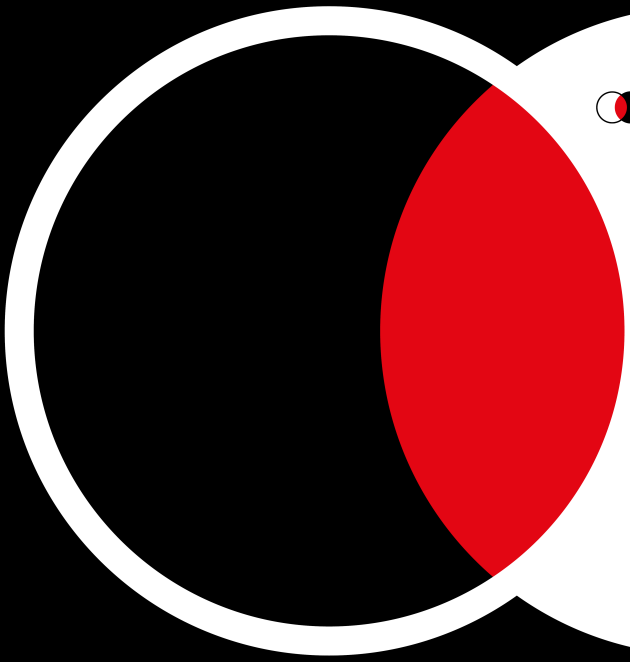


SENSOPART

Kommunikationshandbuch
VISOR®

V 00

First publication XX/2018

SensoPart Industriesensorik GmbH
Nägelseestr. 16
79288 Gottenheim

c UL us LISTED
IND. CONT. EQ. 77HA
For use in class 2 circuits
For use in NFPA 79 Applications only

CE

Table of contents

1 Information on this document

1.1 What the symbols mean

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 comparison

 Contrast

 Brightness

 Grayscale

 Caliper

 BLOB

 Barcode

 Data code

 OCR

 Color value

 Color list

 Color area

 Busbar

 Wafer

Alignment

Alignment
Includes the position detectors: Contour comparison, Pattern matching, and Edge detection

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® Communication Manual
- VISOR® Operating Manual

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 SensoFind is running. To update the list, press the "Find" button for e.g. 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, procedure / Troubleshooting - Direct connection

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

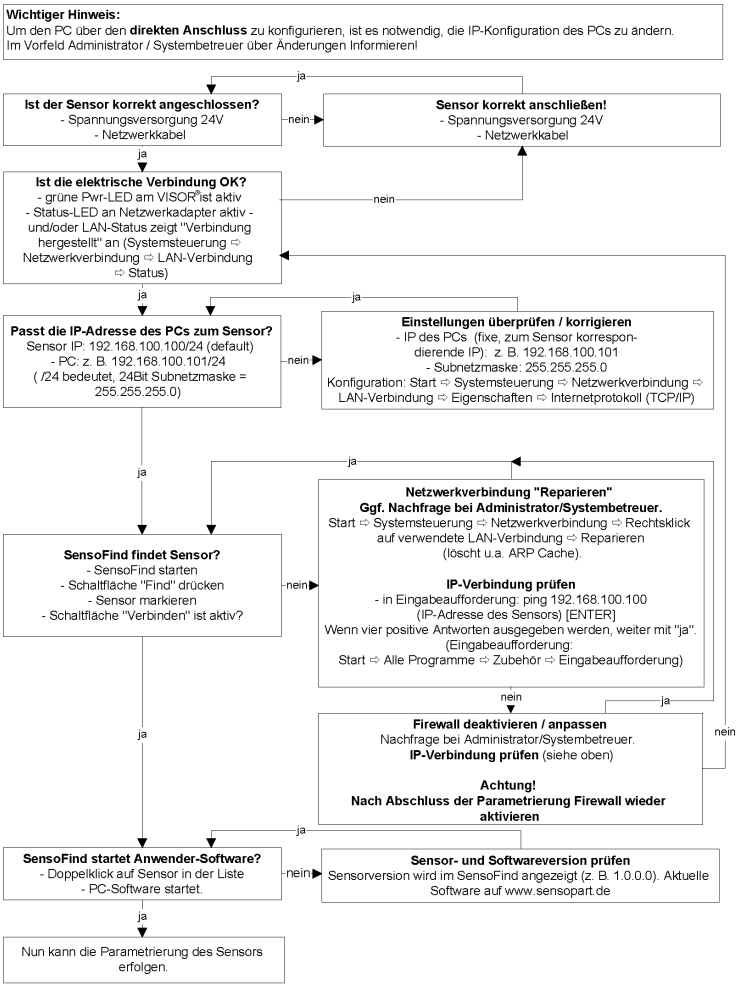


Fig. 1: Direct connection sensor / PC, procedure / troubleshooting

2.3 Network connection, procedure / Troubleshooting - Network connection

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

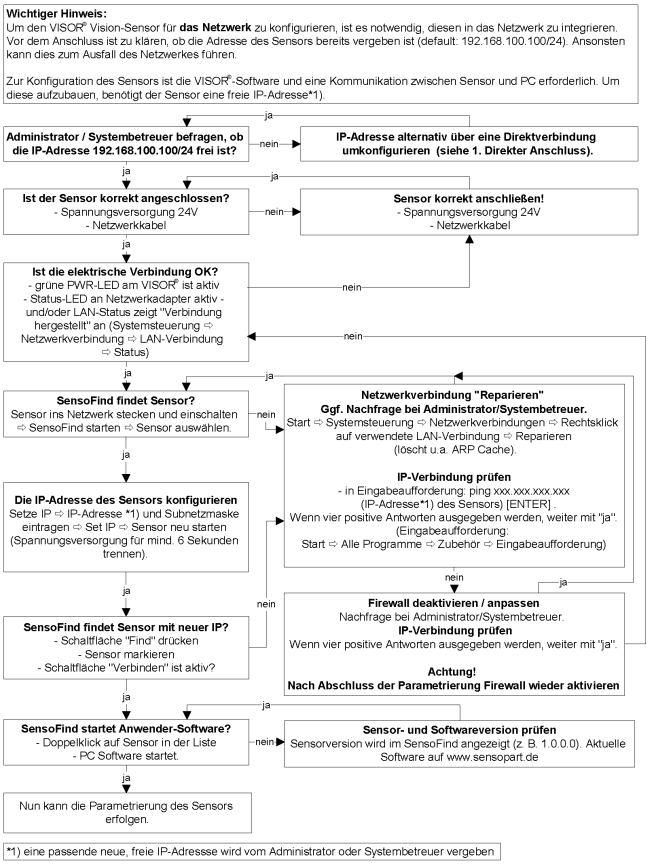


Fig. 2: Connection via network sensor / PC, proceeding / troubleshooting

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® vision sensor:

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

he following ports are used for communications between the PLC (PLC or control PC) and VISOR®.

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:
  - ???
- Service:
  - Port 1998
- 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!

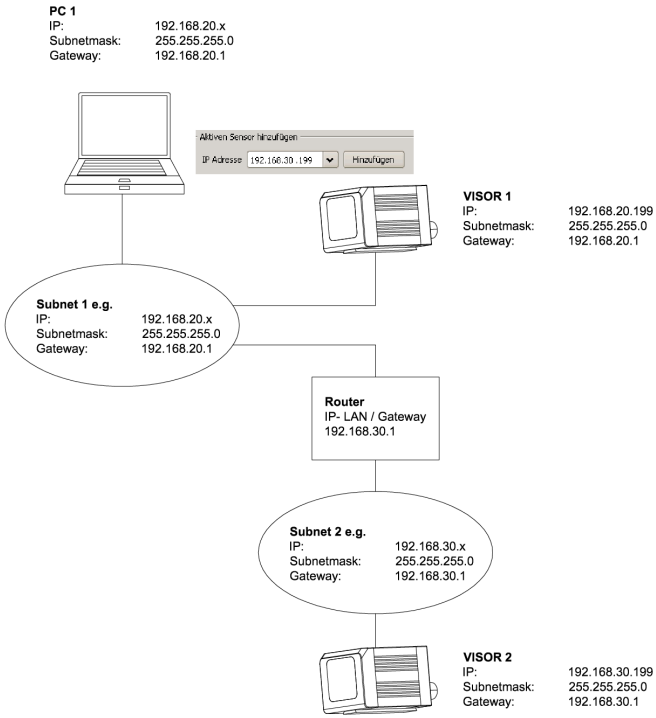


Fig. 3: Access to VISOR® through network, same or other subnet

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) enter and add the IP WAN of Router 2 (company network 2) under "Add active sensor" in SensoFind (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).

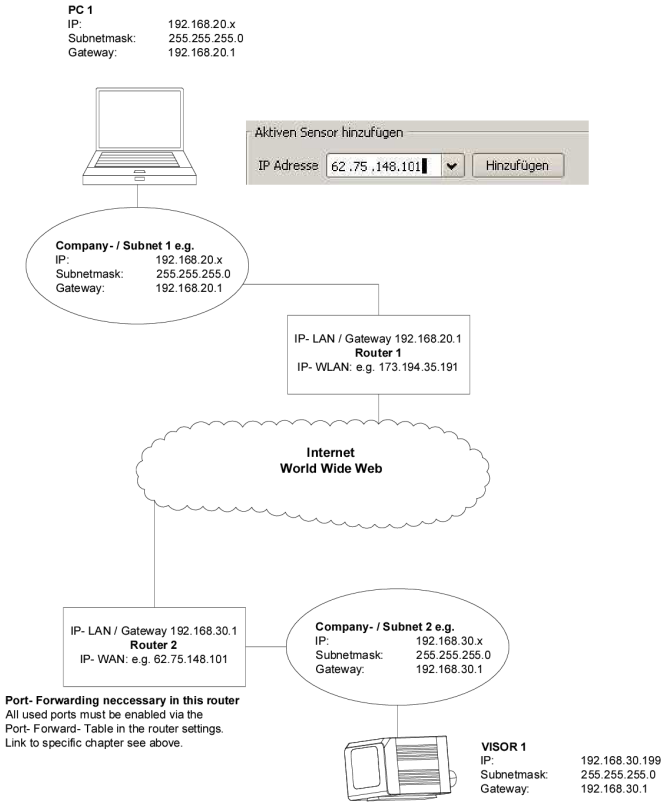


Fig. 4: Access to VISOR® through the Internet / World Wide Web

2.7 VISOR® electrical connection on the network

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

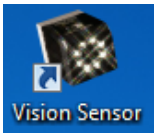
FIGURE

Fig. 5: VISOR® electrical connection on the network

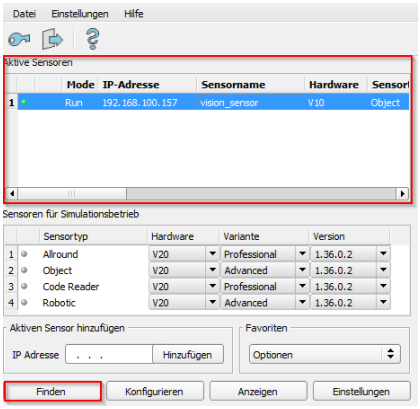
3 VISOR® configuration

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

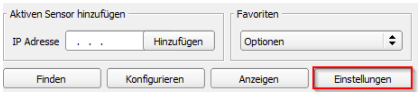
Settings in SensoFind



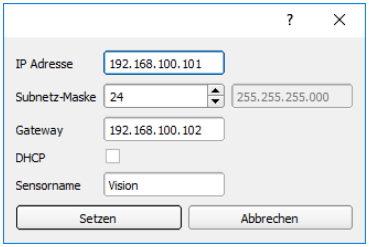
1. Start the VISOR® software . SensoFind are opened.



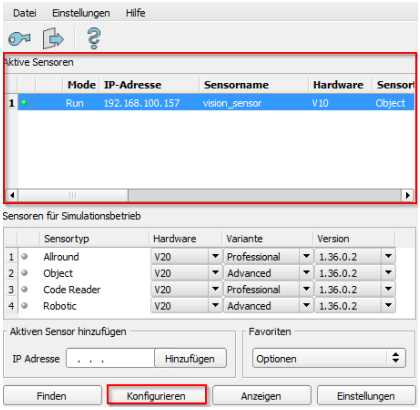
2. Click on the "Find" button. The vision sensor will be listed in the "Active sensors" window.



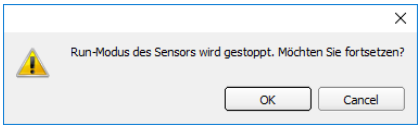
3. Click on the "Settings" button. The dialog box for configuring the IP address and the sensor name will appear.



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.

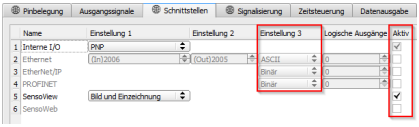


6. Open SensoConfig by selecting the sensor you want and then clicking on the "Configure" button.



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

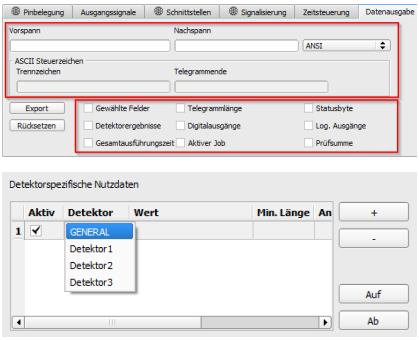
Select an interface in SensoConfig



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

NOTE:
For PROFINET, the binary format must be selected for the data output.

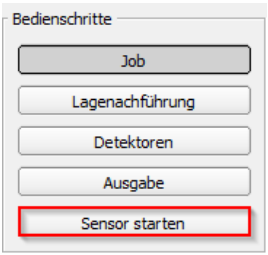
Defining telegrams / data output in SensoConfig



1. Use the "Output" setup step to open the "Data output" tab.
2. Set the control characters you want for the data output.
3. Select the drop-down menus you want.
4. 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
5. Select the detector you want in the "Detector" column.
6. 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



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

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: Commands (requests) from PC / PLC to VISOR®

With acknowledge / 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: Under Job/Image acquisition/Trigger mode = Set Trigger. Otherwise, the settings from Ethernet example 1 will remain the same in the VISOR®.

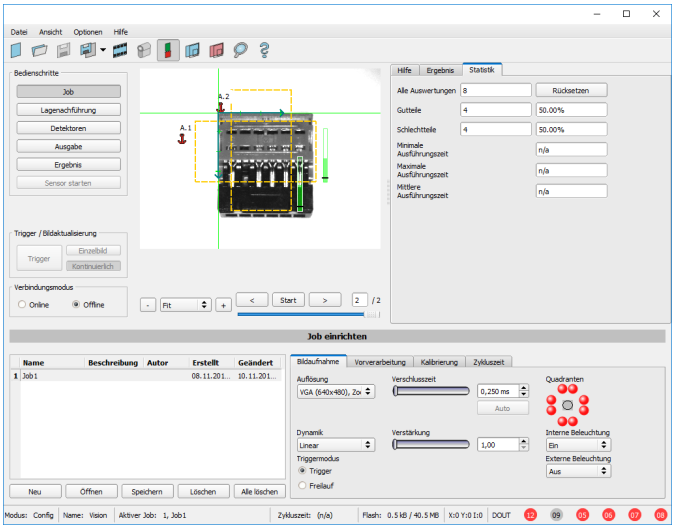


Fig. 6: Data output, Ethernet, Trigger

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 Serial communication ASCII.

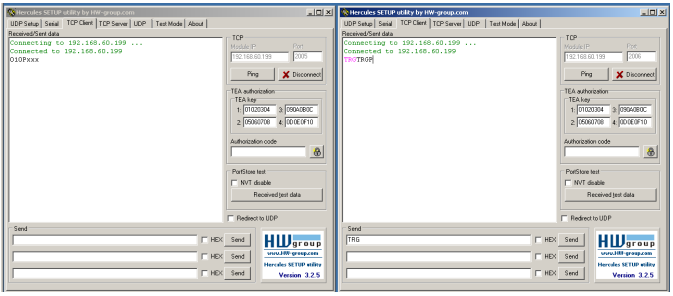


Fig. 7: Data output, Ethernet Tool / 4

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

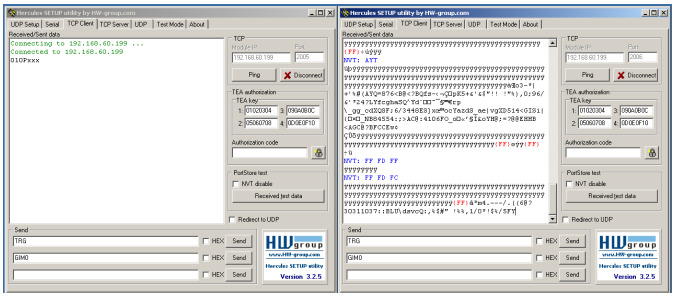


Fig. 8: Data output, Ethernet Tool / 5

In this example, the GIM0 (GetImage0) command was sent to the VISOR®. It responds with the binary image data, as shown in the right part. This means that the user data defined under data output was output again via Port 2005. However, the response to the request "GIM0" was output via Port 2006. This rule applies to all user or response data.

NOTE:
For the command GIMx, the image data recorder must be switched on!

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

With acknowledge / data output from VISOR®

Step 1

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

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

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

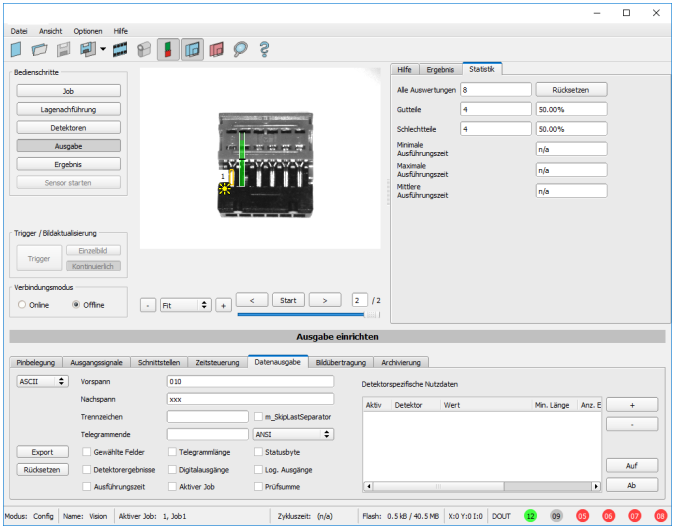


Fig. 9: Data output, Ethernet, Job switch Job 1

Under Job 2, Detector 1 was defined with the data output:

- Trailer: "020"
- Detector 1: "Overall result"
- Trailer: Defined "yyy".

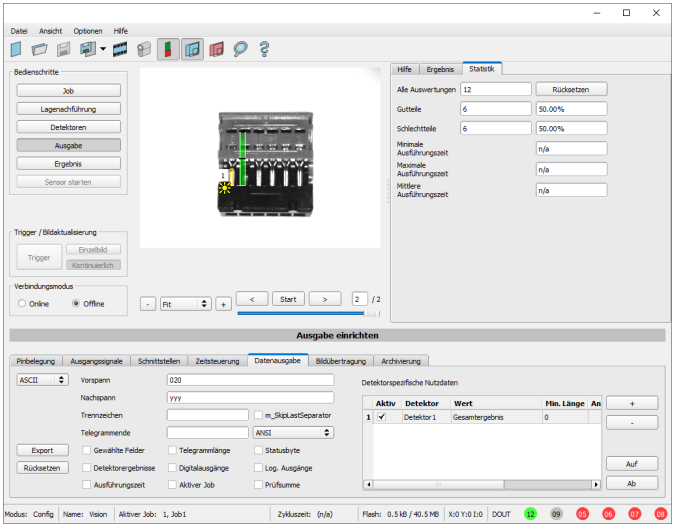


Fig. 10: Data output, Ethernet, Job switch, Job 2

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 Serial communication ASCII.

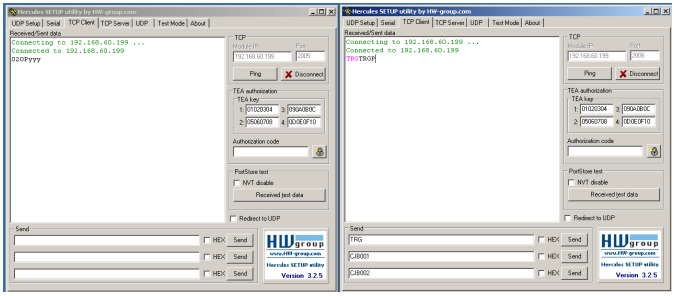


Fig. 11: Data output, Ethernet, Job switch, Tool / 1

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 pane (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".

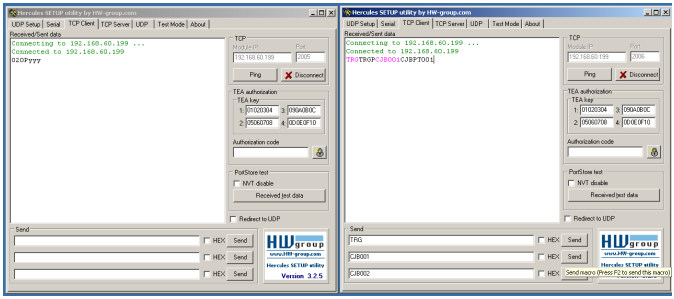


Fig. 12: Data output, Ethernet, Job switch, Tool / 2

Now in the right window (Port2006), the command CJB001 (ChangeJoB 001, 001 = Job no. 1, see "Send" below, second line) was issued. 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)

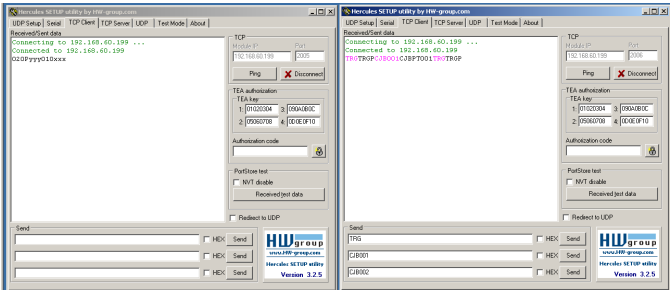


Fig. 13: Data output, Ethernet, Job switch, Tool / 3

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 pane (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"!

Function of both Ethernet ports for in- and output:

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

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

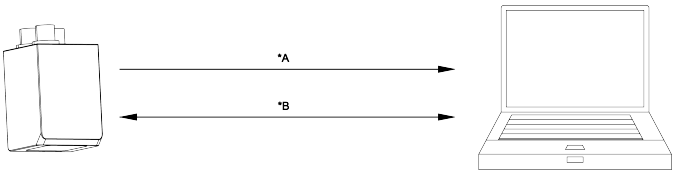


Fig. 14: Ethernet ports

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

Step 1:

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

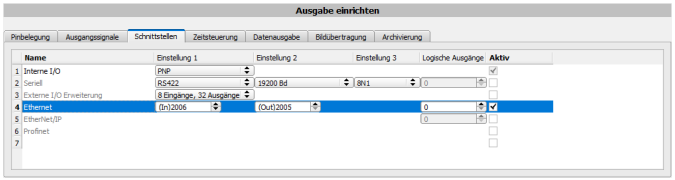


Fig. 15: Data output, Ethernet

In the example, the Ethernet interface is activated in the parameter field in the tab "Interfaces" by marking the checkbox "Active". 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 "Data output" tab configures the user data to be output via Ethernet Port 2005.

In this example, it is the:

- Trailer "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.

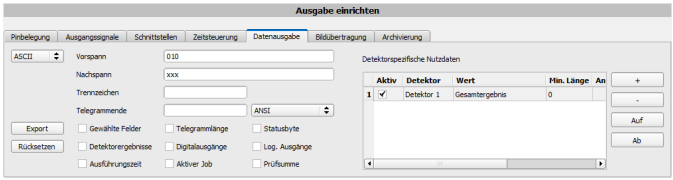


Fig. 16: Data output, Configure output data

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.

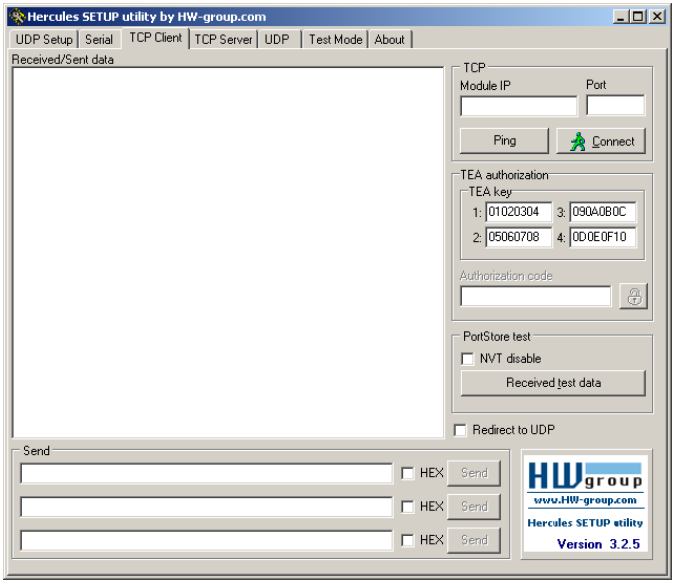


Fig. 17: Data output, Ethernet Tool / 1

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

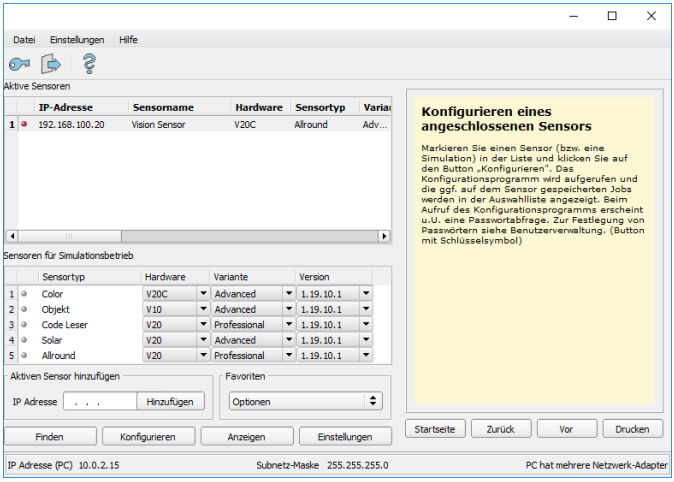


Fig. 18: SensoFind, IP address ...

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.

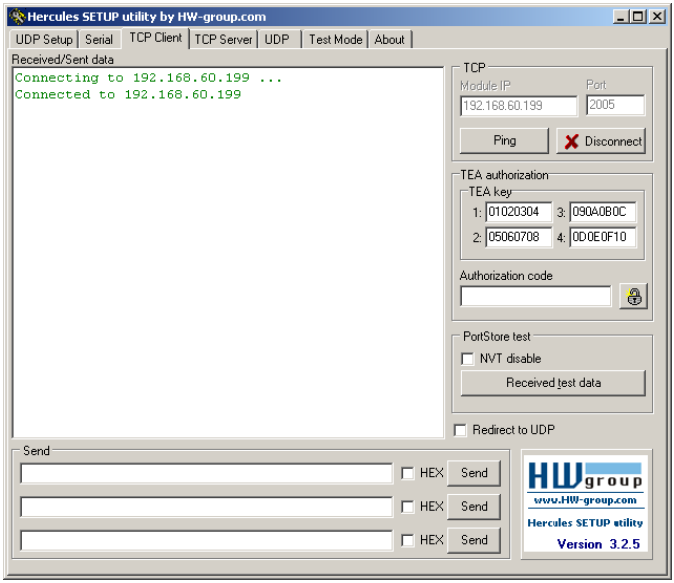


Fig. 19: Data output, Ethernet Tool / 2

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). Here in the example, Trigger mode = continuously set, i.e. evaluation is done continuously and data are sent. These are only visible in the main window of Hercules.

Fig. 20: Data output, Ethernet, Start sensor

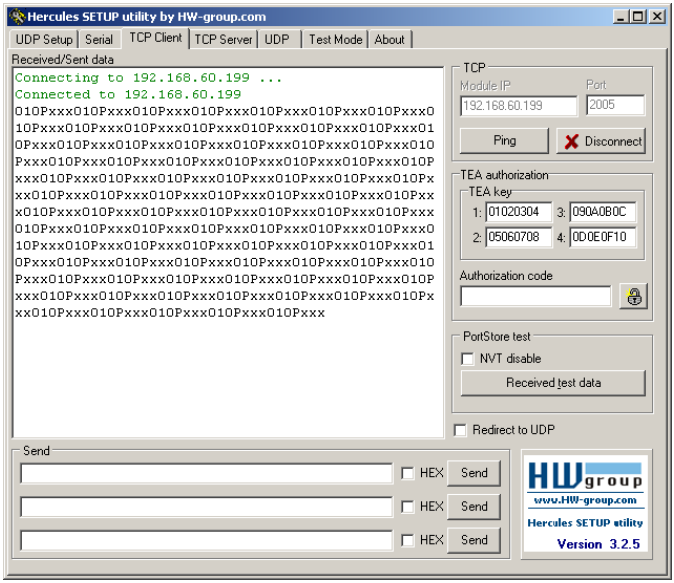


Fig. 21: Data output, Ethernet, Tool / 3

The data visible here are set under "Data output":

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

4.4 Example Beckhoff CX 1020

The connection to a Beckhoff CX 1020 and the corresponding configuration is described in the following document: 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 following document: "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 "Import 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 "Back up 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 through the service port so that applications can be visualized.

Additional information: Update visualization data (ASCII) (Page 160)

6 VISOR® telegrams for PROFINET and EtherNet/IP

6.1 Module 1: "Control" (from controller to VISOR®)

Name in controller "CTRL (3 bytes)"

Byte Offset	Bit-Adr.	Name	Datentyp	Bedeutung
0	0	Reset Error	1 Bit	Reset Error löscht den 4 Bit Error Code im Status Modul. Steigende Flanke (False → True) löscht den Error code.
	1	HW-Trigger Disable	1 Bit	Dieses Bit dient zum Disable des Hardware Triggers. Gültig für Trigger- und Freilaufmodus. • False (0): Hardware Trigger aktiviert. • True (1): Hardware Trigger deaktiviert. Falls der Digital Eingang "Trigger enable" genutzt wird, müssen zum Akzeptieren eines Triggers beide Bedingungen (Digitaler Eingang „Hardware Trigger“ und „HW- Trigger Disable- Bit“) auf “Enable” gesetzt sein.
	2	Trigger	1 Bit	Steigende Flanke (False → True): Trigger, wird sofort ausgeführt. Falls der Trigger nicht ausgeführt werden konnte, bleibt das Trigger acknowlede Bit False und Error Status Modul weist den Error Code "1: Failure Trigger Request" auf. S. auch Timing Diagramm "Fall: Trigger nicht möglich"
	3	Change Job	1 Bit	Steigende Flanke (False → True): Umschalten auf Job mit Nummer "Job number" aus Control Modul. Beim Ausführen dieses Requests kann es zu Verzögerungen kommen. Nach einem erfolgreichen Jobwechsel, zeigt das Byte "Job number" im Status Modul den gleichen Wert wie im Control Modul. Falls der Jobwechsel nicht durchgeführt werden konnte (auf Grund eines Fehlers, z.B. falsche Jobnummer) weist das Error Status Modul den Error Code "2: Failure Change Job" auf (und Ready bleibt False!). S. auch Timing Diagramm: "Fall: Jobchange nicht möglich"
	4	Switch-to-Run	1 Bit	Steigende Flanke (False → True): "Switch-to-Run" wird ausgeführt. Erfolg oder Fehler des Switch-to-Run Requests wird angezeigt im Error Status Modul (Error Code "3: Failure Switch-to-Run Request") und Bit "Operation Mode". S. auch Timing Diagramm: "Fall: Switch-to-Run nicht möglich"
	5 - 7	Reserve		
1		Reserve		
2		Job number		Job number auf die umgeschaltet werden soll, bei der steigenden Flanke des Change Job Bit. Binärer Wert 1 - 255 für "Job number Change". 0 steht für: „Keine Umschaltung”, auch wenn das Change Job Bit wechselt.

Timing diagrams for VISOR® communication (Page 35)

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

Name in PLC "STAT (6 bytes)"

Byte Offset	Bit-Adr.	Name	Datentyp	Bedeutung
0	0	Ready	1 Bit	Sensor ist bereit für die nächste Auswertung. Ready = 1.  **ACHTUNG** Das Ready Bit ist ausschließlich zur Anzeige der Bereitschaft des Sensors für die nächste Auswertung reserviert. Es ist nicht geeignet zur Anzeige, dass eine Auswertung abgeschlossen ist, bzw. die Ergebnisse einer Auswertung vorliegen! Siehe hierzu Image ID.
	1	Reserve	1 Bit	
	2	Trigger acknowledge	1 Bit	Acknowledge (Bestätigung) für erfolgreichen Trigger Request (via Trigger Bit im Control Modul). Acknowledge wird gelöscht als Response auf das Löschen des Trigger Bit. Falls der Trigger nicht ausgeführt werden konnte, bleibt das Trigger acknowledge Bit false.
	3	Change Job acknowledge	1 Bit	Acknowledge (Bestätigung) für den erfolgten Change Job Request (via Change Job Bit im Modul Control) – unabhängig von dessen Erfolg. Acknowledge wird gelöscht sobald das Change Job Request Bit gelöscht wurde. Erfolg oder Fehler des Change Job Request wird angezeigt im Error Status Modul (Fehlercode "2: Failure Change Job") und im Byte "Job number" im Status Modul. Bei Verzögerungen der Ausführung des Change Job kann dieses acknowledge Bit ebenfalls verzögert gesetzt werden.
	4	Switch-to-Run acknowledge	1 Bit	Acknowledge (Bestätigung) für den erfolgten Switch-to-Run Request (via Switch-to-Run Request Bit im Modul Control). Acknowledge wird gelöscht sobald das Request Bit gelöscht wurde. Erfolg oder Fehler von Switch-to-Run Request wird angezeigt im Error Status Modul (Fehlercode "3: Failure Switch-to-Run Request") und Bit "Operation Mode". Acknowledge wird gesetzt nachdem SensoConfig geschlossen, und der Job aus dem Flash geladen wurde, oder wenn ein Fehler auftrat.
	5 - 7	Reserve		
1		Reserve	1 Byte	

Byte Offset	Bit-Adr.	Name	Datentyp	Bedeutung
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	Dieses Byte wird gefüllt mit den Ergebnissen der digitalen Schaltausgänge. Die Bitposition ist fix. Der Wert des Ausgangs wird definiert im Tab: Ausgabe/Ausgangssignale, Spalte: „Logischer Ausdruck“ in SensoConfig. Falls nicht ausgewählt als Ergebnis-Ausgabe-Pin, oder falls kein gültiger logischer Ausdruck zugewiesen ist, ist der Wert = 0.
	7	Reserve	1 Bit	
3		Job number	U8	Nummer des aktuellen Jobs: 1 - 255.
4		Image ID	U8	Image ID (0 - 255) wird nach jeder Job-Ausführung um Eins erhöht, unabhängig von der Trigger Quelle.
5	0 - 3	Error Status Modul	4 Bit	4 Bit Error Code. Zeigt Fehler bei Requests via Control Modul oder Sensor Systemfehler an. Error wird gelöscht durch "Reset Error" oder bei Überschreiben durch nächsten Fehler. 0: Kein Fehler 1: Fehler Trigger Request (Sensor nicht Ready) 2: Fehler Change job 3: Fehler Switch-to-Run 5: Fehler Schnittstelle in Job nicht aktiv 15: System Fehler
	4	Trigger mode	1 Bit	1 = Freilauf 0 = Trigger
	5	Reserve	1 Bit	
	6	Operation mode		1 = Run 0 = Config
	7	Reserve	1 Bit	

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

Name in PLC "DATA (2 + 8 / 16 / … bytes)"

Byte Offset	Bit-Adr.	Name	Datentyp	Bedeutung
0		Image ID	U8	Image ID (0 - 255) wird nach jeder Job-Ausführung um Eins erhöht, unabhängig von der Trigger Quelle.
1	0	Result Data overrun	1 Bit	Ergebnis Daten wurden abgeschnitten. 1: Data overrun = abgeschnitten 0: No overrun
	1 - 7	Reserve	7 Bit	
2		Result Data	Byte array	Daten wie definiert in SensoConfig unter "Ausgabe/Datenausgabe/ Detektorspezifische Nutzdaten". Bei Nutzung von PROFINET muss im Reiter Schnittstellen “Binär” aktiviert sein.

6.4 Module 4: "Request" (from controller to VISOR®)

Name in PLC "REQU (4 + 8 / 16 / … bytes)"

Byte Offset	Bit-Adr.	Name	Datentyp	Bedeutung
0	1	Key	1 Byte	Key Request (Request counter)
1	1	Reserve	1 Byte	
2	1	Reserve	1 Byte	
3	1	Reserve	1 Byte	
4		Request Data	Byte array	Siehe Anhang Kapitel "Übersicht Telegramme".

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

Name in PLC "RESP (4 + 8 / 16 / … bytes)"

Byte Offset	Bit-Adr.	Name	Datentyp	Bedeutung
0		Key	U8	Key Response wird nach Ausführung von Request gespiegelt.
1	0	Result Data overrun	1 Bit	Response Daten wurden abgeschnitten.
	1 - 7	Reserve	7 Bit	
2		Reserve	1 Byte	
3		Reserve	1 Byte	
4		Request Data	Byte array	Siehe Anhang Kapitel "Übersicht Telegramme".

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

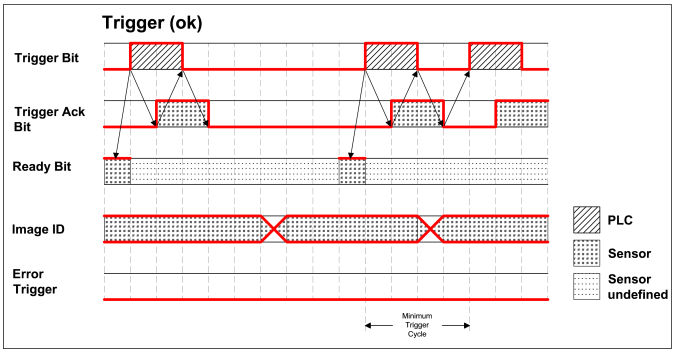


Fig. 22: Timing Trigger ok

Case: Trigger not possible (not ready)

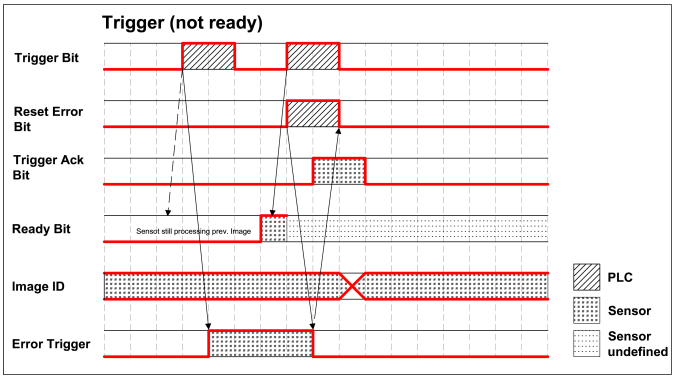


Fig. 23: Timing Trigger not ready

Case: Job change ok

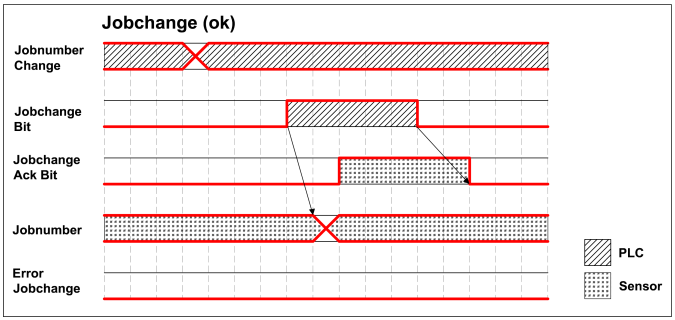


Fig. 24: Timing Job change ok

Case: Job change delayed

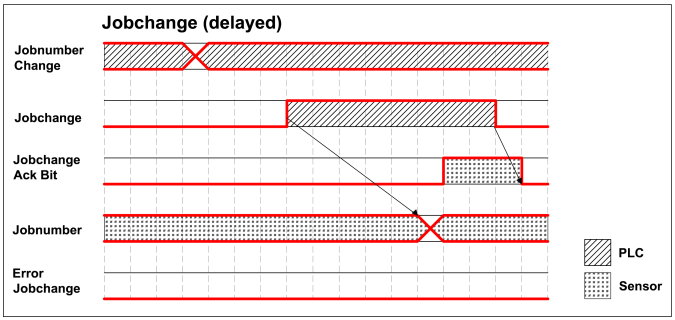


Fig. 25: Timing Job change delayed

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

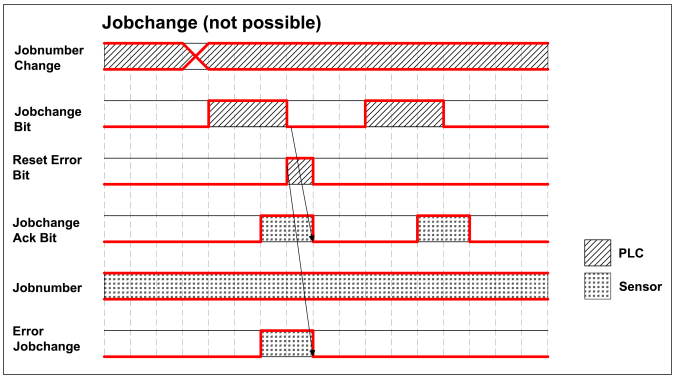


Fig. 26: Timing Job change not possible

Case: Switch to run ok

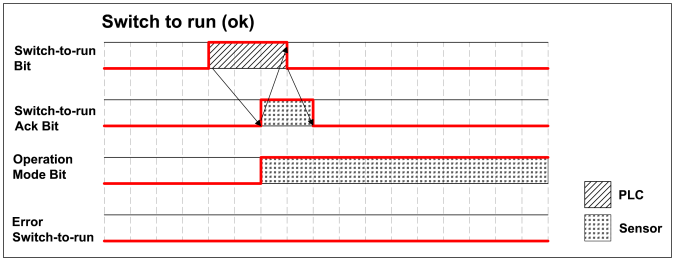


Fig. 27: Timing Switch to run ok

Case: Switch to run not possible

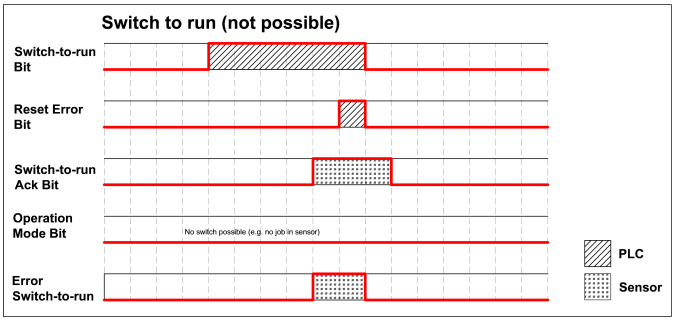


Fig. 28: Switch to run not possible

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.

   PLEASE NOTE:

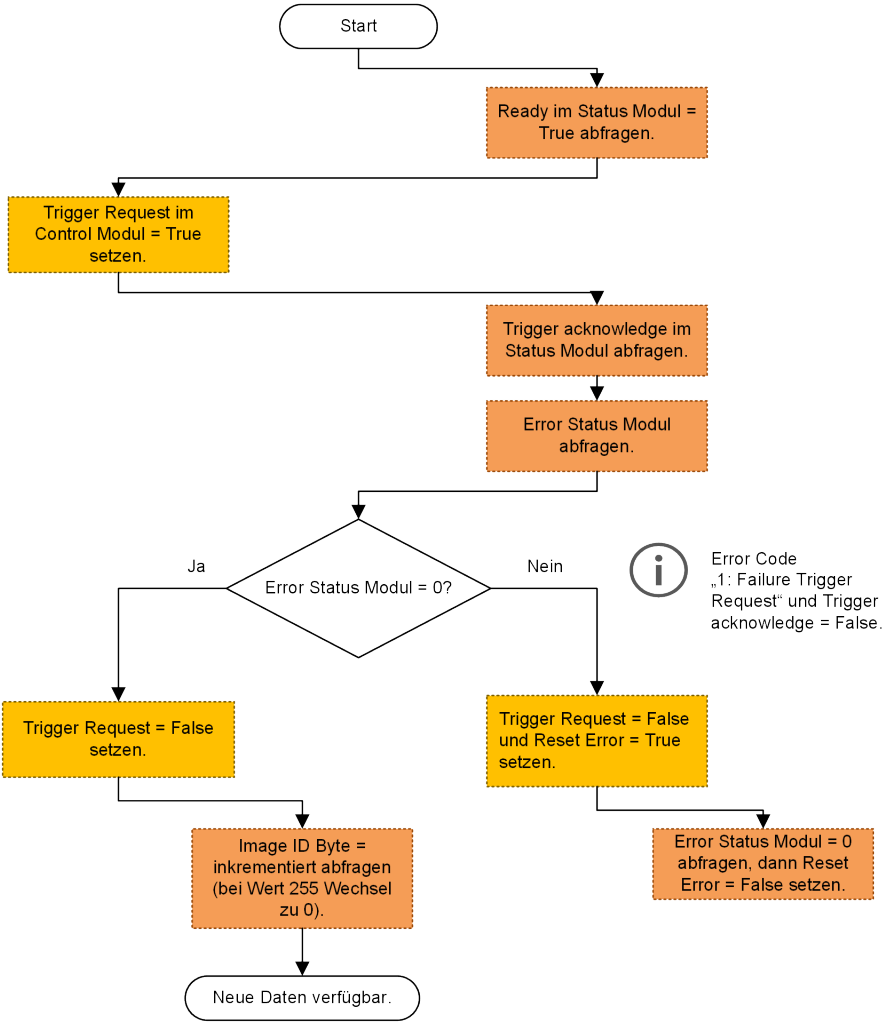
   The complete execution of an action cannot be reliably determined based on a low/high transition at READY, since it is possible for there to be long cycle times between the PLC and VISOR® (e.g., 32 ms) that result in READY never having a low state.

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

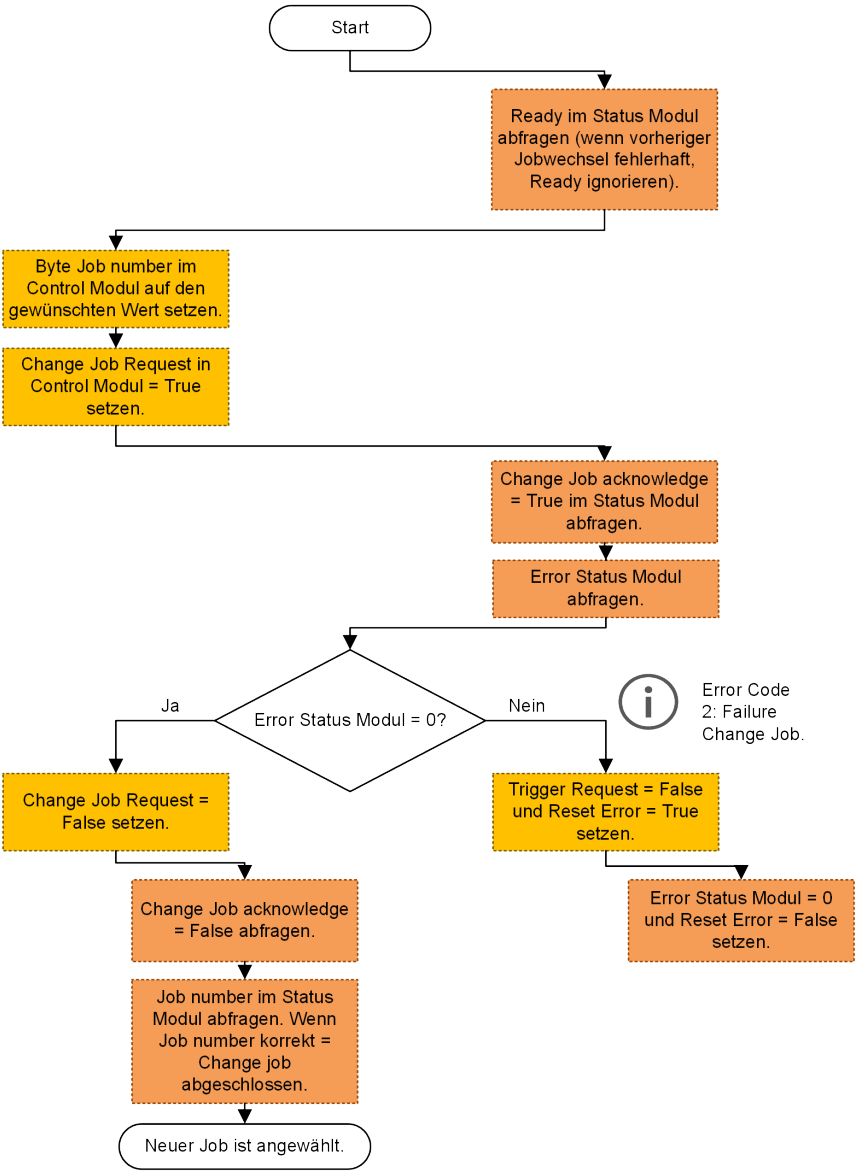
Accepting / discarding of requests of the control module

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)

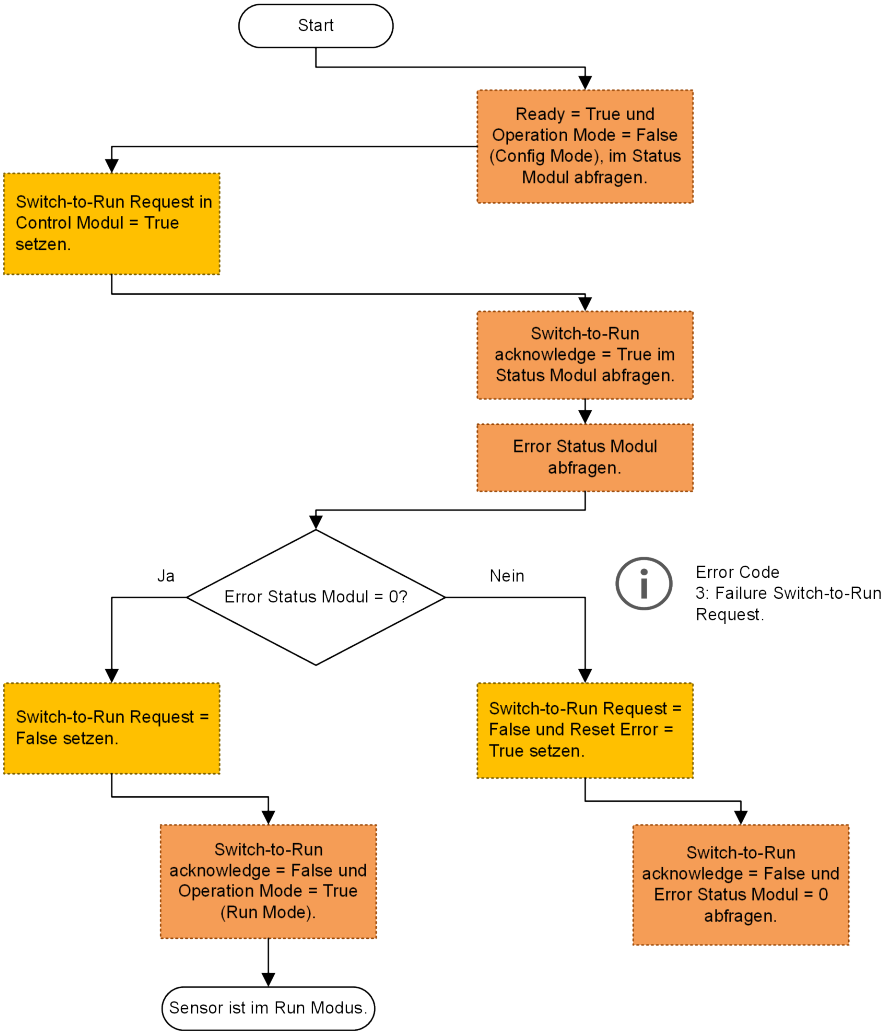
Trigger Request Sequence



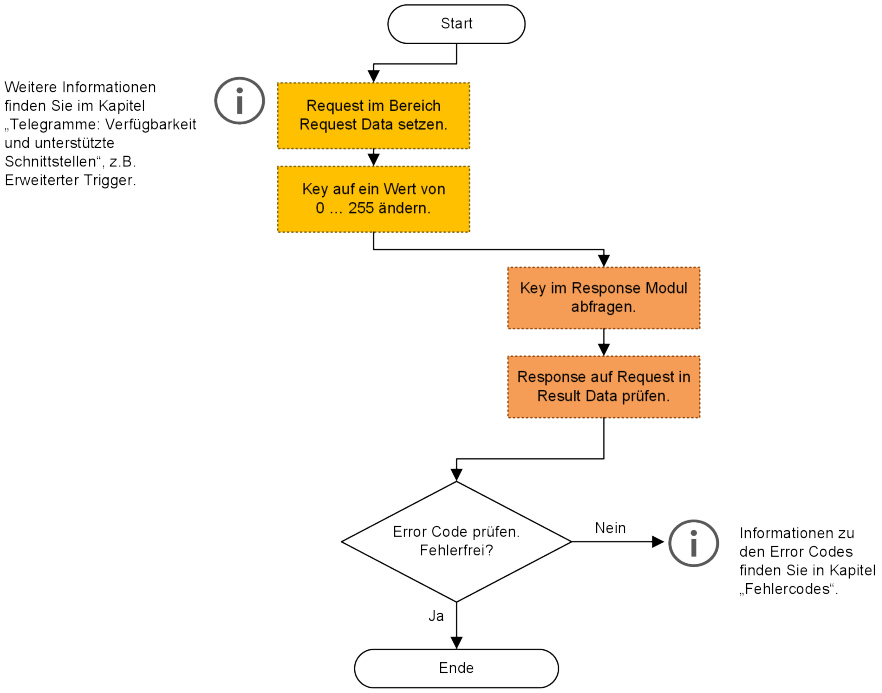
Change job request sequence



Switch to Run sequence


Start
Ready = True und Operation Mode = False (Config Mode), im Status Modul abfragen.
Switch-to-Run Request in Control Modul = True setzen.
Switch-to-Run acknowledge = True im Status Modul abfragen.
Error Status Modul abfragen.
Error Status Modul = 0?
Ja
Nein
Error Code 3: Failure Switch-to-Run Request.
Switch-to-Run Request = False setzen.
Switch-to-Run Request = False und Reset Error = True setzen.
Switch-to-Run acknowledge = False und Operation Mode = True (Run Mode).
Switch-to-Run acknowledge = False und Error Status Modul = 0 abfragen.
Sensor ist im Run Modus.

Sequence for requests via request/response module



Telegrams: Availability and supported interfaces (Page 84)

Error codes (Page 87)

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

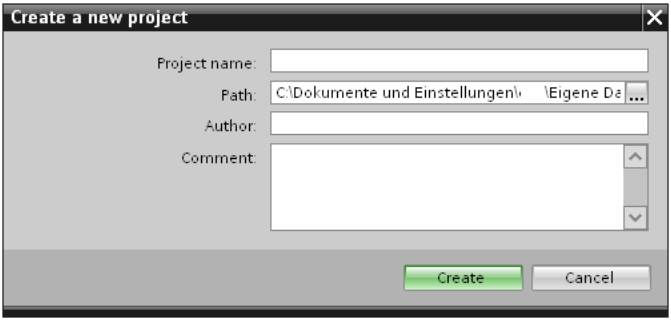


Fig. 29: PROFINET Create new project

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 GSD 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.

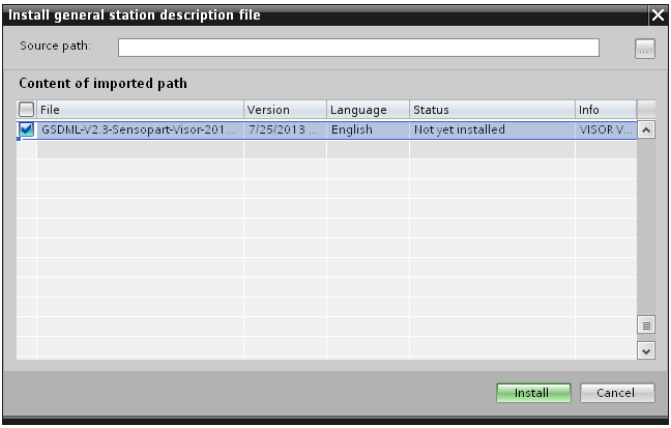


Fig. 30: Selecting and installing the GSD file

9.1.3 Adding the VISOR® to the project

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

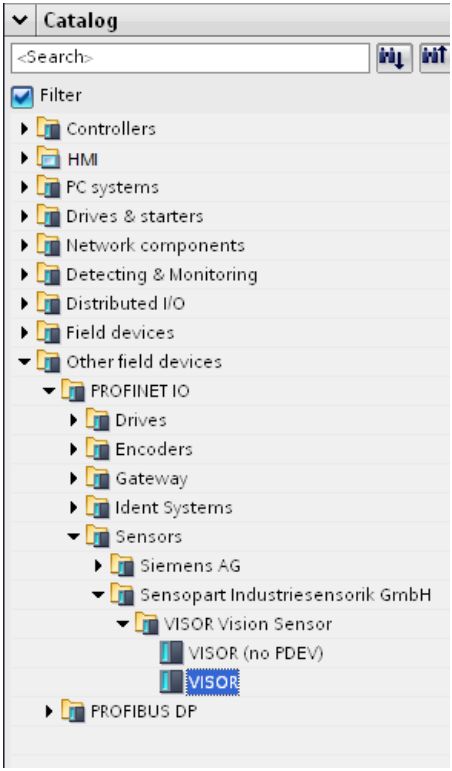


Fig. 31: Adding the VISOR® to the project

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).

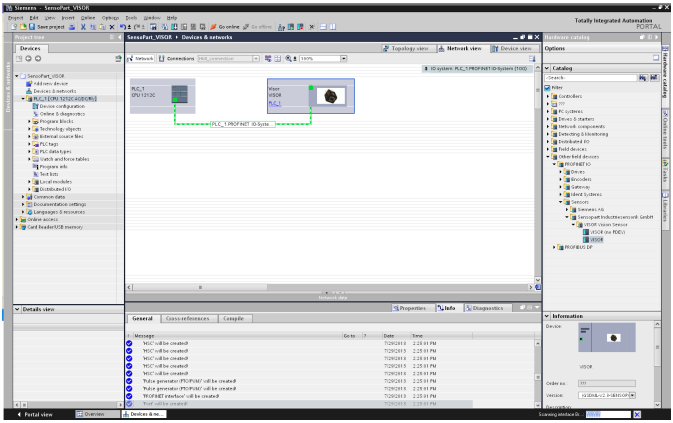


Fig. 32: Connecting the VISOR® to the PLC

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 controller) (Page 33)), + 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.

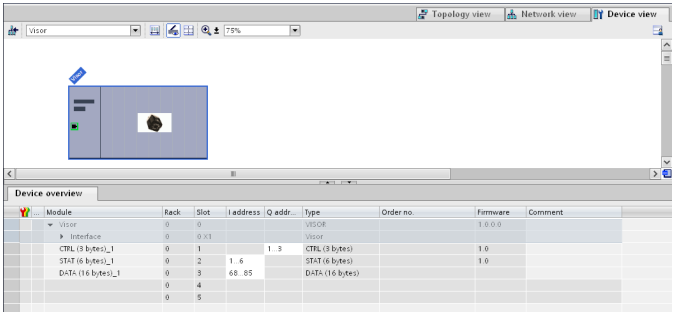


Fig. 33: Enter I/O data

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.

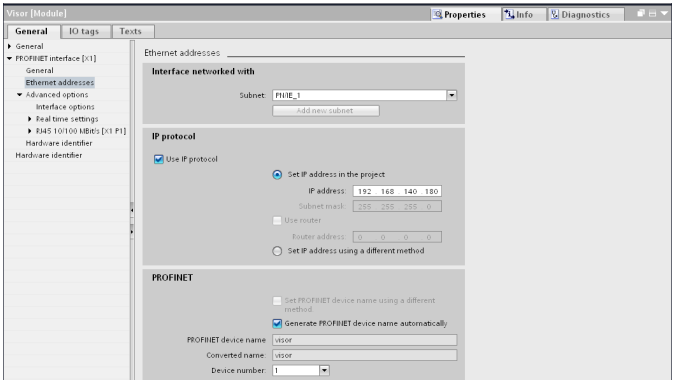


Fig. 34: Configuring the VISOR® IP address in the project

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 (please refer to section: Settings in SensoFind (Page 14)).

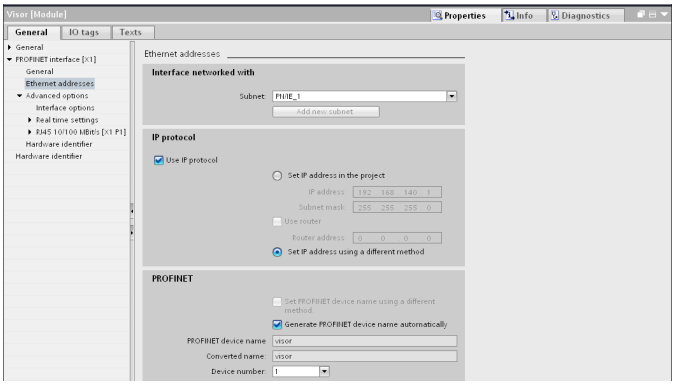


Fig. 35: Configure the IP address for the VISOR® in SensoFind; the corresponding settings can be found in the PLC/TIA interface

Set the name in the TIA portal

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 activated, any name can be edited.

Information: In the field "Converted name", a different name than entered is displayed, which is then also used. As in the PROFINET, not all characters can be used, a conversion may be necessary, which is made here automatically (names must be DNS-compatible, see also Chapter Settings in SensoFind (Page 14)).

If the VISOR® vision sensor'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 VISOR®).

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

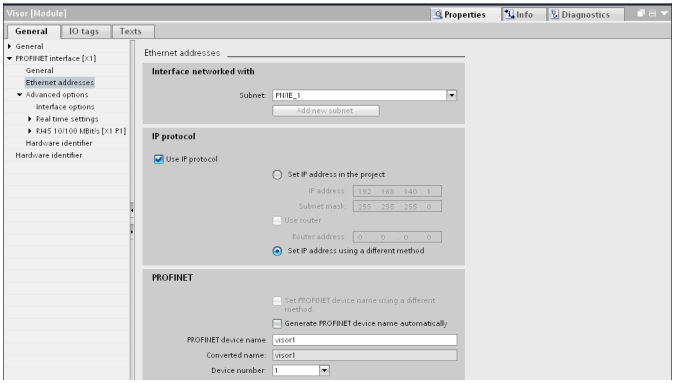


Fig. 36: Set name in project

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."

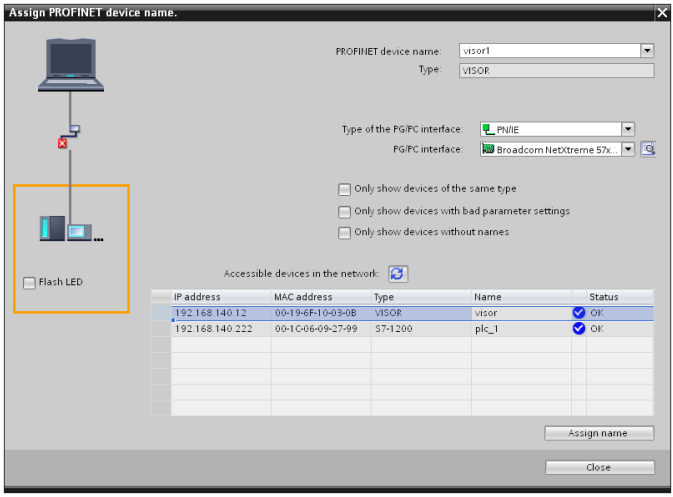


Fig. 37: Writing a name to VISOR®

9.1.5 Load the project onto the PLC

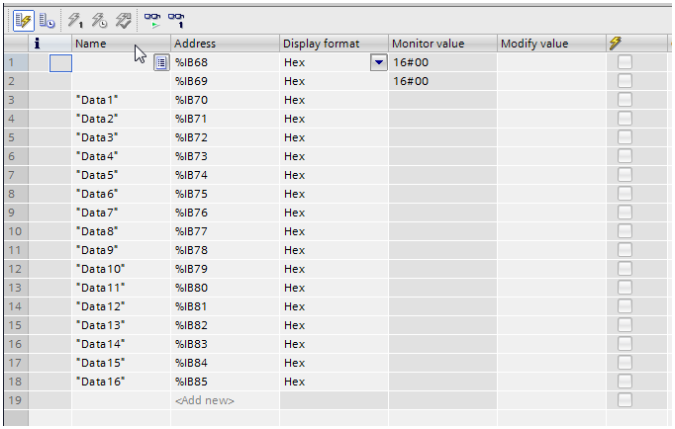
To finish the configuration and save changes of the project: 1. translate and 2. transfer / write to the PLC

Fig. 38: 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:

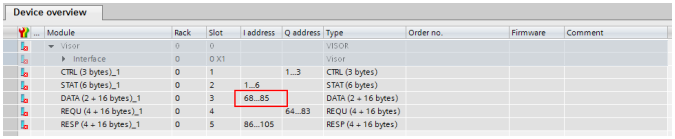
Step 1) The start address for an input variable can be taken from "Device Overview".



	Name	Address	Display format	Monitor value	Modify value
1		%IB68	Hex	16#00	
2		%IB69	Hex	16#00	
3	"Data1"	%IB70	Hex		
4	"Data2"	%IB71	Hex		
5	"Data3"	%IB72	Hex		
6	"Data4"	%IB73	Hex		
7	"Data5"	%IB74	Hex		
8	"Data6"	%IB75	Hex		
9	"Data7"	%IB76	Hex		
10	"Data8"	%IB77	Hex		
11	"Data9"	%IB78	Hex		
12	"Data10"	%IB79	Hex		
13	"Data11"	%IB80	Hex		
14	"Data12"	%IB81	Hex		
15	"Data13"	%IB82	Hex		
16	"Data14"	%IB83	Hex		
17	"Data15"	%IB84	Hex		
18	"Data16"	%IB85	Hex		
19		<Add new>			

Fig. 39: Table of variables

Step 2) Creating a tag table in the PLC



Device overview

	Module	Rack	Slot	I address	Q address	Type	Order no.	Firmware	Comment
	VISOR	0	0			VISOR			
	Interface	0	0 X1			VISOR			
	CTRL (3 bytes)_1	0	1		1...3	CTRL (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. 40: Device overview

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

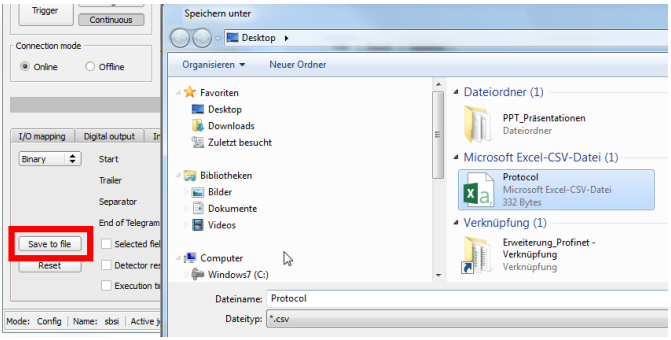
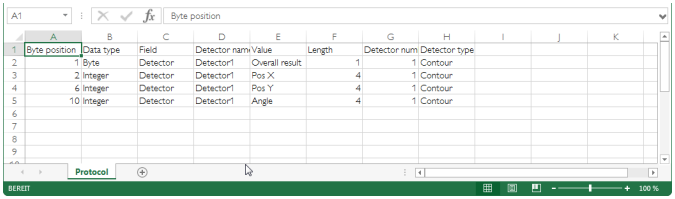


Fig. 41: Output format saved as CSV file

Step 4) Opening the file with the text program

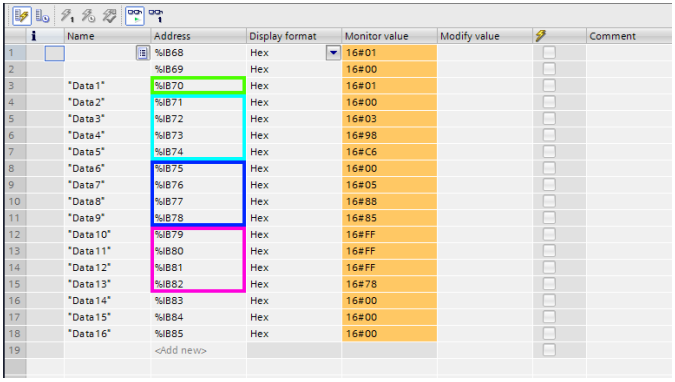


Byte position	Data type	Field	Detector name	Value	Length	Detector number	Detector type
1	Byte	Detector	Detector1	Overall result	1	1	Contour
2	Integer	Detector	Detector1	Pos X	4	1	Contour
4	Integer	Detector	Detector1	Pos Y	4	1	Contour
10	Integer	Detector	Detector1	Angle	4	1	Contour

Fig. 42: 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 controller) (Page 33)

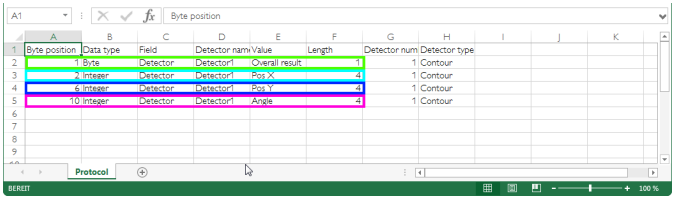
Step 5) The result is the following assignment between the input data of the PLC



	Name	Address	Display format	Monitor value	Modify value	Comment
1		%IB68	Hex	16#01		
2		%IB69	Hex	16#00		
3	"Data 1"	%IB70	Hex	16#01		
4	"Data 2"	%IB71	Hex	16#00		
5	"Data 3"	%IB72	Hex	16#03		
6	"Data 4"	%IB73	Hex	16#98		
7	"Data 5"	%IB74	Hex	16#C6		
8	"Data 6"	%IB75	Hex	16#00		
9	"Data 7"	%IB76	Hex	16#05		
10	"Data 8"	%IB77	Hex	16#88		
11	"Data 9"	%IB78	Hex	16#85		
12	"Data 10"	%IB79	Hex	16#FF		
13	"Data 11"	%IB80	Hex	16#FF		
14	"Data 12"	%IB81	Hex	16#FF		
15	"Data 13"	%IB82	Hex	16#78		
16	"Data 14"	%IB83	Hex	16#00		
17	"Data 15"	%IB84	Hex	16#00		
18	"Data 16"	%IB85	Hex	16#00		
19		<Add new>				

Fig. 43: Input data PLC

... and the configured protocol:



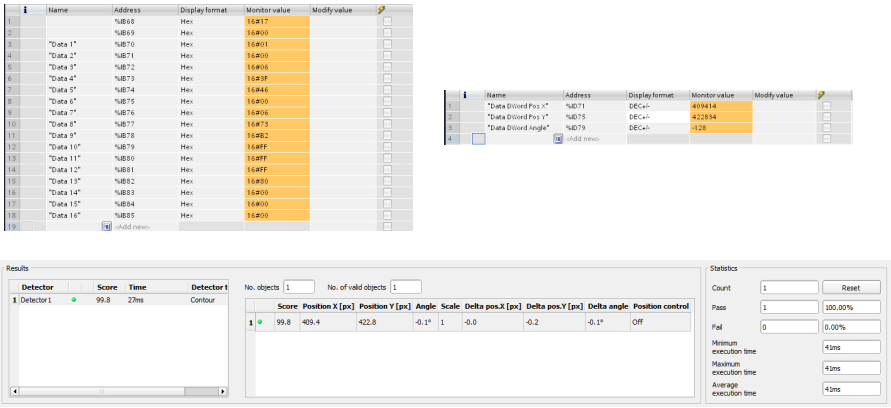
Byte position	Data type	Field	Detector name	Value	Length	Detector number	Detector type
1	Byte	Detector	Detector1	Overall result	1	1	Contour
2	Integer	Detector	Detector1	Pos X	4	1	Contour
4	Integer	Detector	Detector1	Pos Y	4	1	Contour
10	Integer	Detector	Detector1	Angle	4	1	Contour

Fig. 44: 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).



9.2 PLC example programs

The following PLC example programs show some basic functions.

PLC example 1: Trigger when VISOR® ready

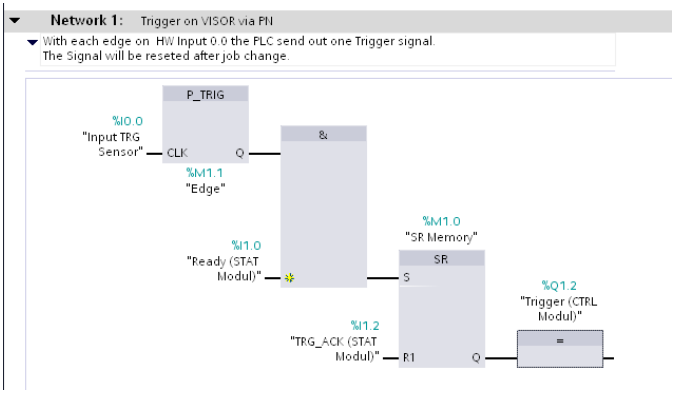


Fig. 45: Trigger when VISOR® ready (without error handling)

PLC example 2: Send Job number to VISOR®

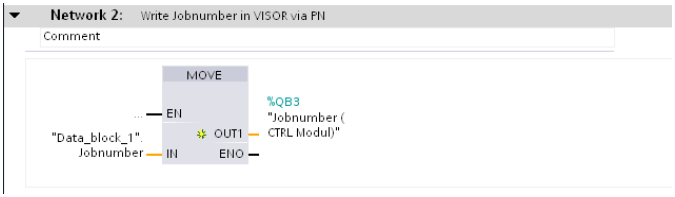


Fig. 46: Send job number

PLC example 2.1: Job change when VISOR® ready

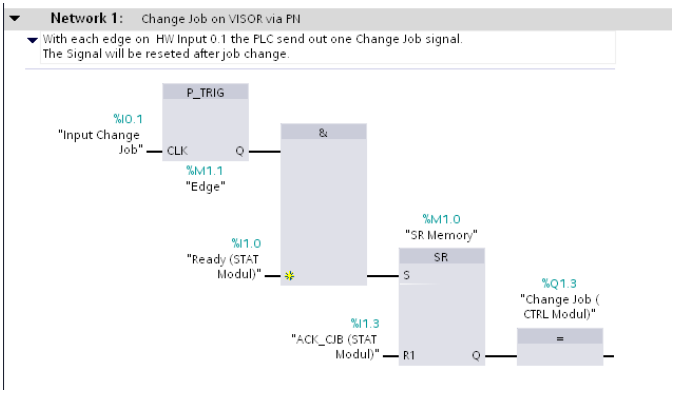


Fig. 47: Job change when VISOR® ready (without error handling)

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

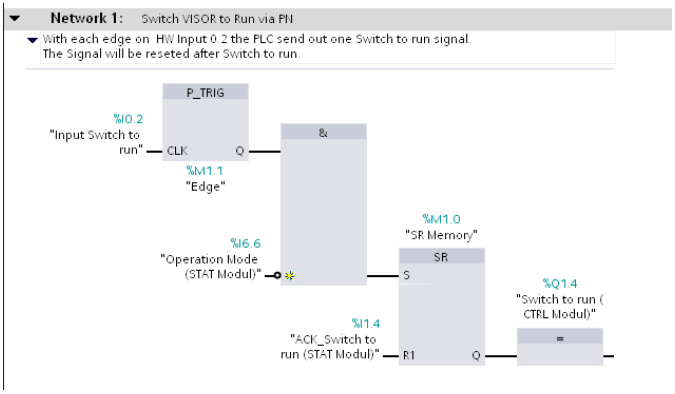


Fig. 48: Switch to Run when VISOR® in configuration mode (without error handling)

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

Variable "Data Array" (type: Array of Byte) Length (34 bytes) = User data (32) + 2 bytes (header)

(Module "Data" with 32 bytes: User data + 1 byte: Image ID + 1 byte: Result data overrun = 34 bytes)

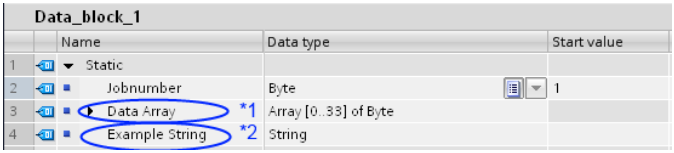


Fig. 49: Data block for data transfer

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.

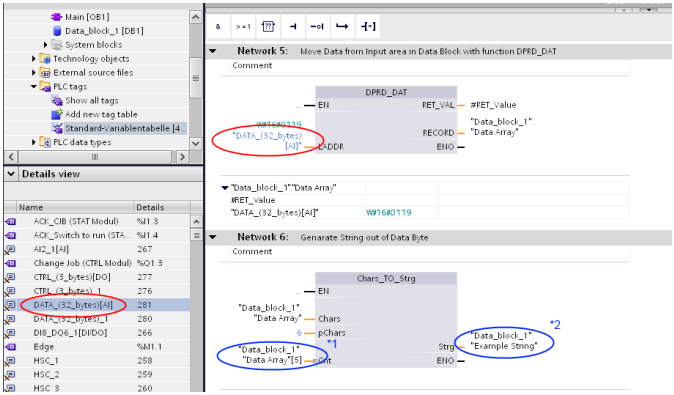


Fig. 50: Data transfer

PLC example 4.2: VISOR® telegram settings

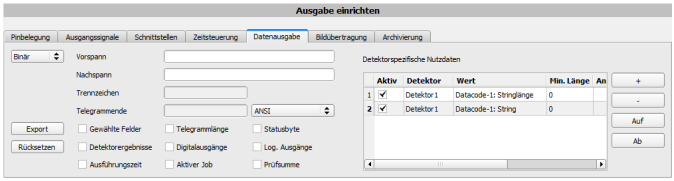


Fig. 51: Settings for sample telegram in VISOR®

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).

10.2 Rockwell Studio 5000 configuration example

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

10.2.1 Create new project

Create the project in: Create/New Project.

Fig. 52: EtherNet/IP Create new project

Select the right PLC type and assign a name.

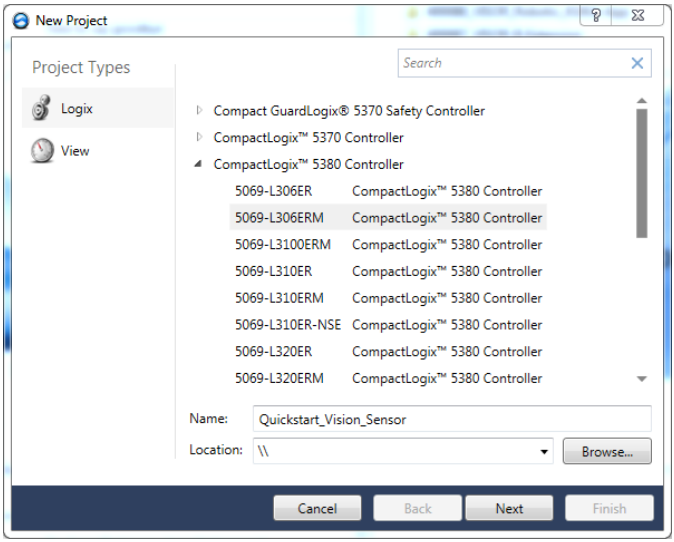


Fig. 53: EtherNet/IP Select the PLC type.

You can apply the default settings. Click on "Finish" to create the project.

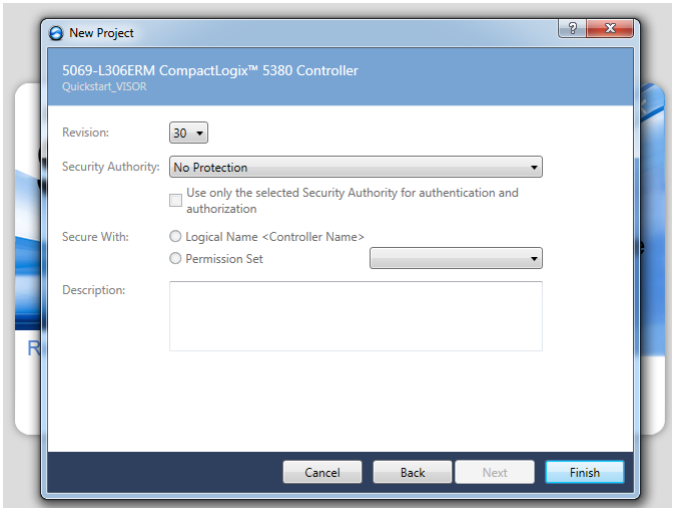


Fig. 54: EtherNet/IP Apply the default settings.

10.2.2 Select EDS file

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. The EDS file can be installed in: Tools/EDS Hardware Installation Tool.

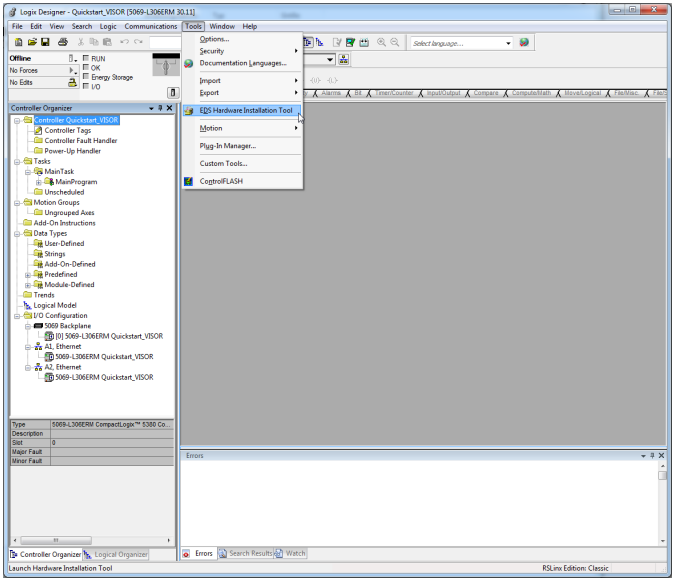


Fig. 55: Project view, Tool, EDS Hardware Installation Tool

Confirm information with Next.

Fig. 56: Confirming information

Select "Register an EDS File(s)" in the options.

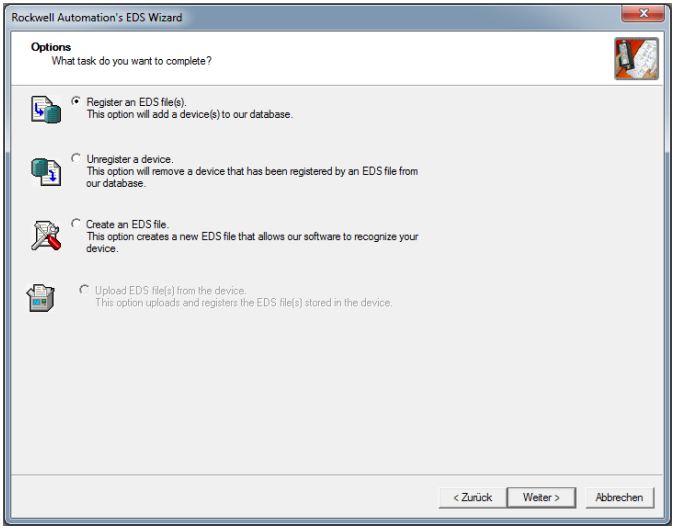


Fig. 57: Register an EDS File(s)

Select "Register a single file" and specify the path to the EDS file.
The EDS file can be found in the installation path for the VISOR® in: ...\SensoPart\VISOR® Vision-Sensor\Tools\EtherNet/IP and is also available for download at www.sensopart.com.

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

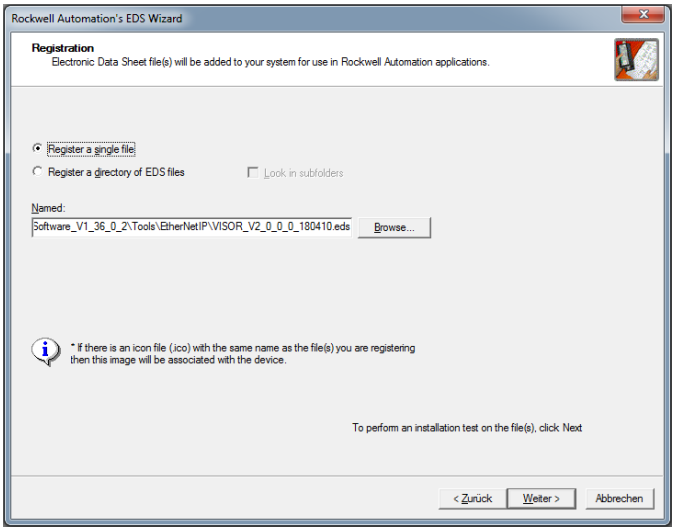


Fig. 58: Select EDS file

Confirm the EDS file test.

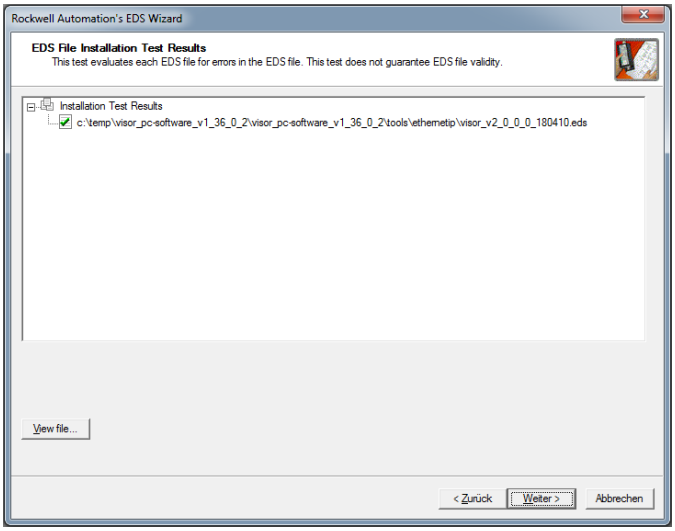


Fig. 59: EDS file test

If necessary, select an icon or continue with the default icon.

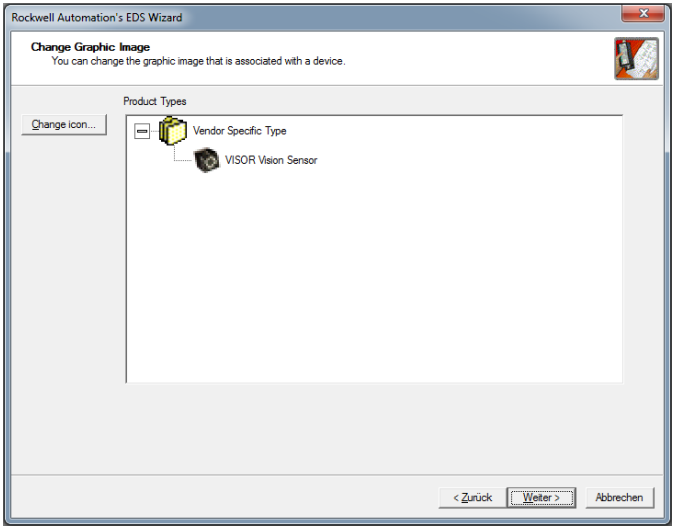


Fig. 60: Icon

Confirm the installation.

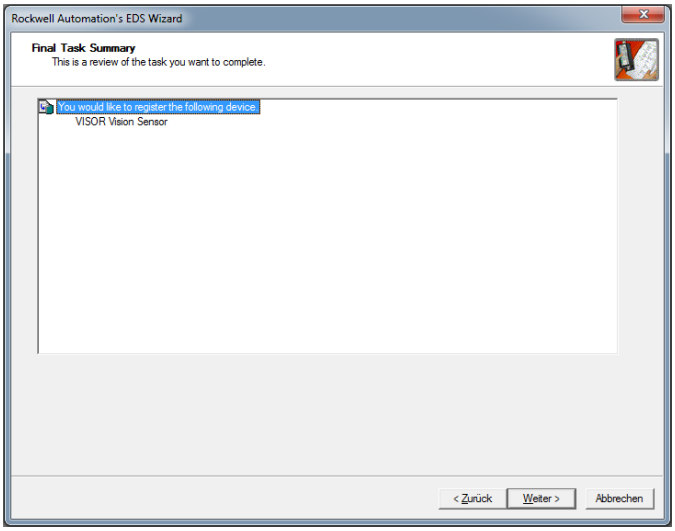


Fig. 61: Confirming the installation

Click on Finish to successfully complete the installation.

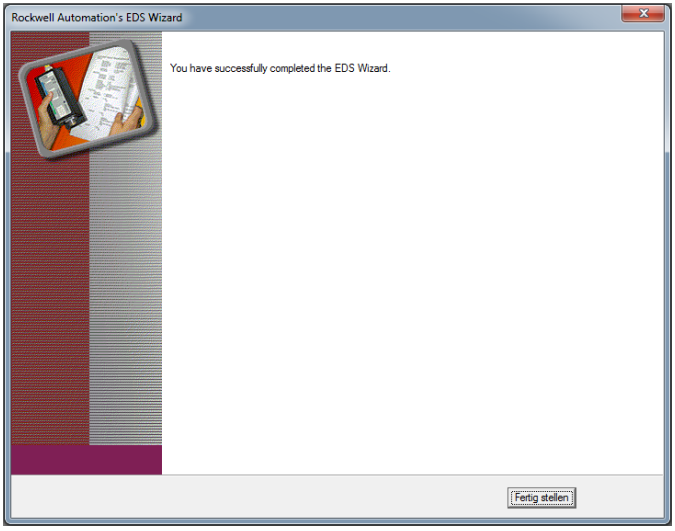
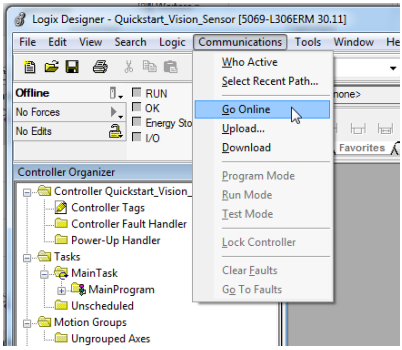


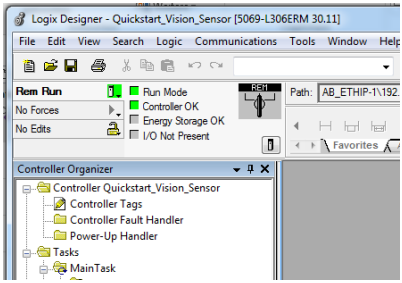
Fig. 62: Finishing the installation

10.2.3 Load the project onto the PLC

Go online with the project. To do this, click on Communications/Go Online.

NOTE:
Before this, the project path must be configured correctly.





Right-click on the network connection you want to create a new module.

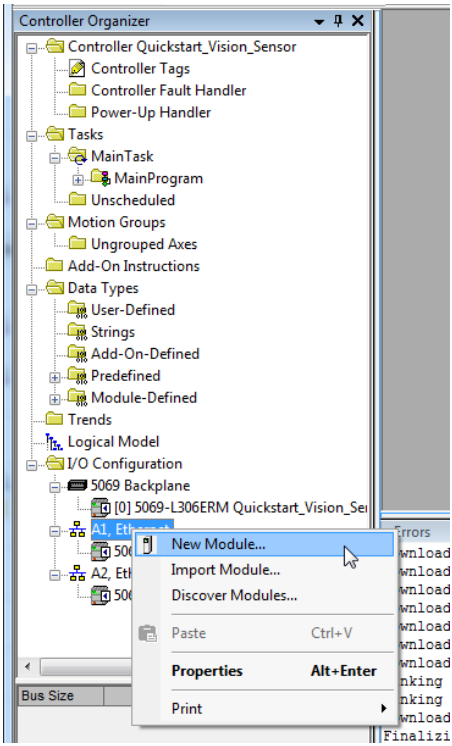
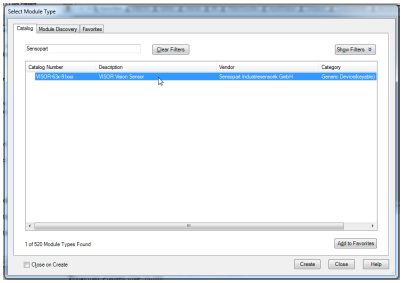
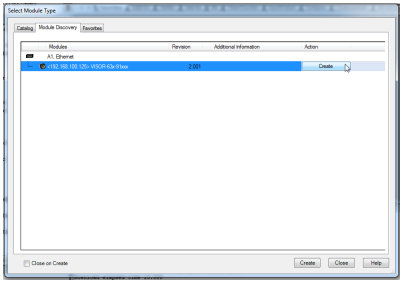


Fig. 63: Creating a new module

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

NOTE:
The software must already be online in order to be able to use the online search option.





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

Configure the device name and IP address for the VISOR®

The device name will be used as a variable name for the data later on. The IP address can be read with the SensoFind module in the VISOR® software.

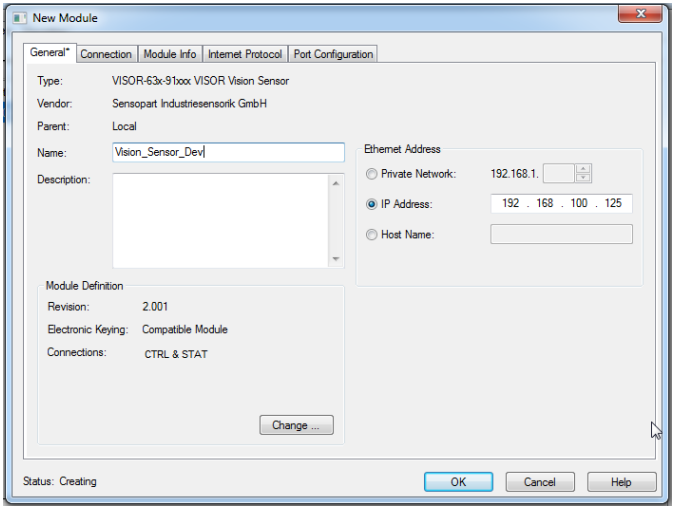
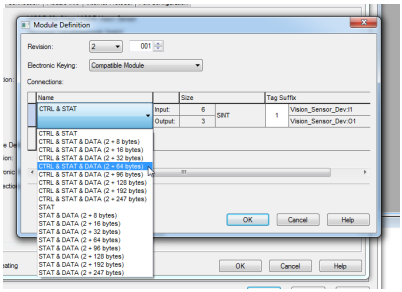
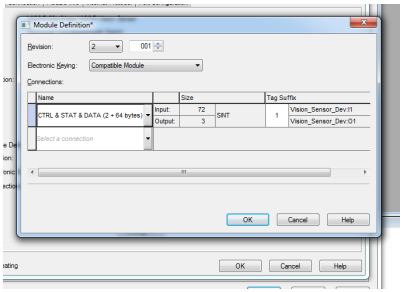


Fig. 64: Configure the device name and IP address

You can select the modules and module sizes you want with "Change ..." Additional information:
VISOR® telegrams for PROFINET and EtherNet/IP (Page 30)





In Connection, configure the refresh rate time(RPI) you want.

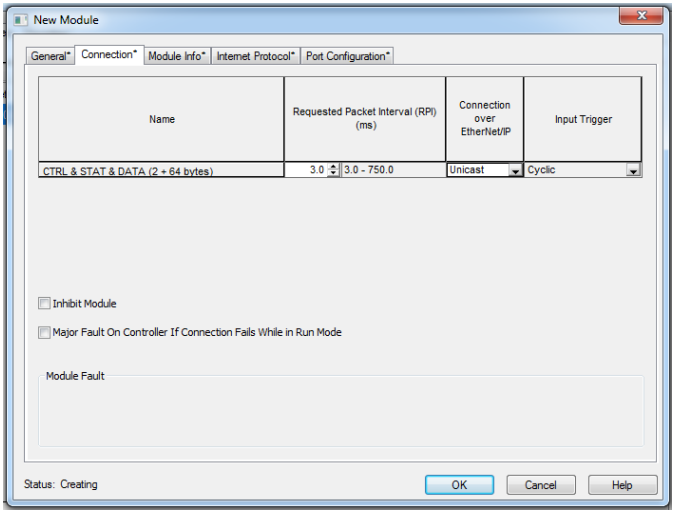


Fig. 65: Set the refresh rate.

Click on OK to finish configuring the settings for the node.

Download the project onto the PLC with Communications/Download.

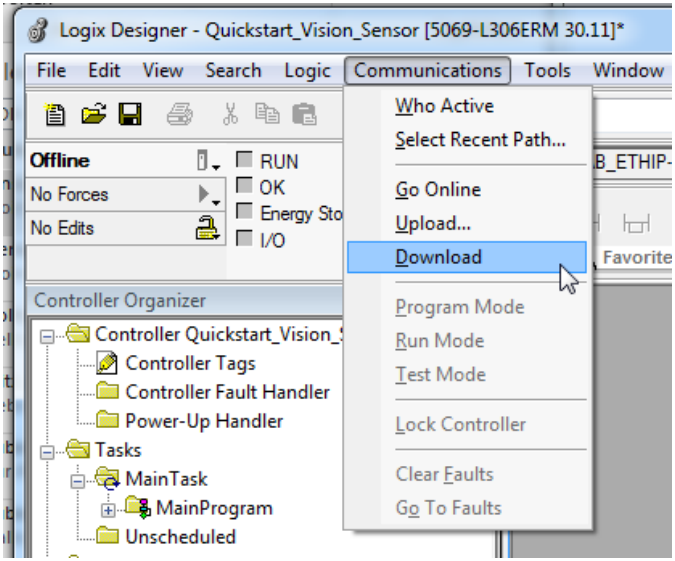


Fig. 66: Download

Check all information and confirm with Download.

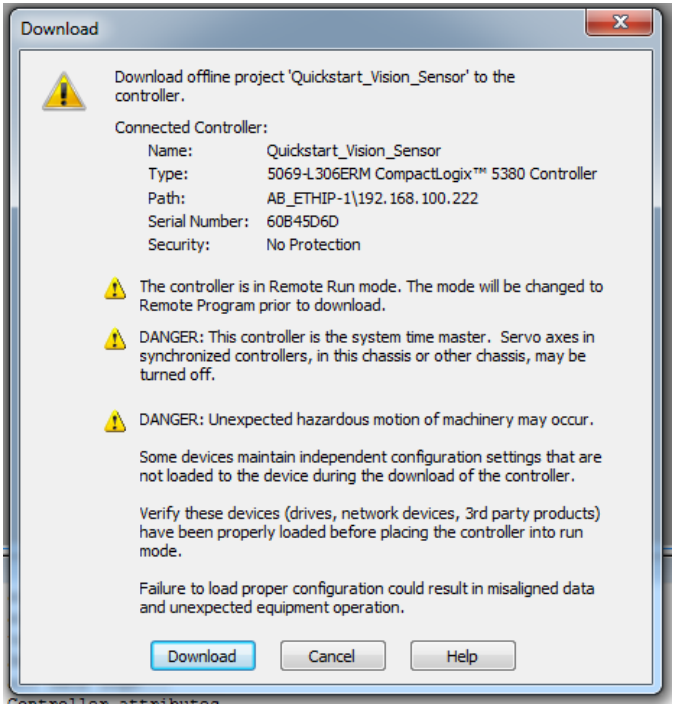


Fig. 67: Information

After a successful download, the sensor will have the "Running" status.

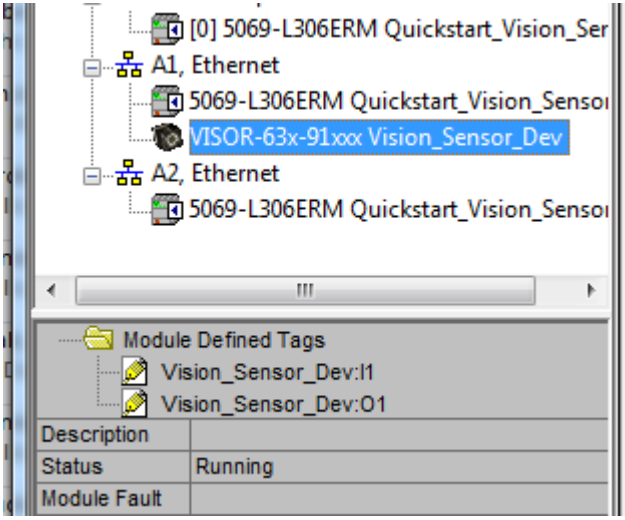


Fig. 68: Running

10.2.4 Mapping of output data

The input data is mapped as follows. (selection of module CNTL + STAT + data (2+69))

Vision-Sensor-Dev …. I1.Data[0] – …. I1.Data [5] Status module (Module 2: "Status" (from VISOR® to controller) (Page 31))

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[71]

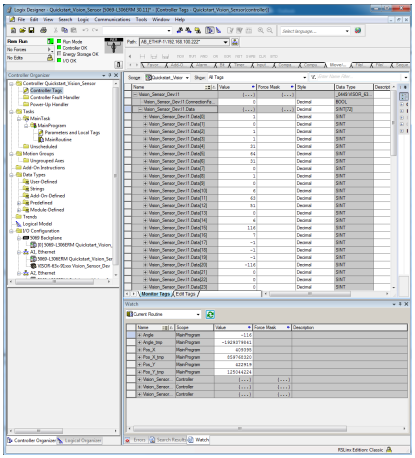
The data specified in Output/Telegram is inserted here. Additional information: Defining telegrams / data output in SensoConfig (Page 16)

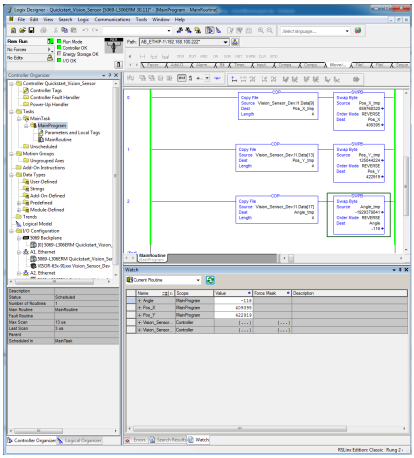
Name	Value	Force Mask	Style	Da
Vision_Sensor_Dev:I1.Data	{...}	{...}	Decimal	SI
Vision_Sensor_Dev:I1.Data[0]	1		Decimal	SI
Vision_Sensor_Dev:I1.Data[1]	0		Decimal	SI
Vision_Sensor_Dev:I1.Data[2]	0		Decimal	SI
Vision_Sensor_Dev:I1.Data[3]	1		Decimal	SI
Vision_Sensor_Dev:I1.Data[4]	6		Decimal	SI
Vision_Sensor_Dev:I1.Data[5]	0		Decimal	SI
Vision_Sensor_Dev:I1.Data[6]	6		Decimal	SI
Vision_Sensor_Dev:I1.Data[7]	0		Decimal	SI
Vision_Sensor_Dev:I1.Data[8]	0		Decimal	SI
Vision_Sensor_Dev:I1.Data[9]	0		Decimal	SI
Vision_Sensor_Dev:I1.Data[10]	0		Decimal	SI
Vision_Sensor_Dev:I1.Data[11]	0		Decimal	SI
Vision_Sensor_Dev:I1.Data[12]	0		Decimal	SI
Vision_Sensor_Dev:I1.Data[13]	0		Decimal	SI
Vision_Sensor_Dev:I1.Data[14]	0		Decimal	SI
Vision_Sensor_Dev:I1.Data[15]	0		Decimal	SI
Vision_Sensor_Dev:I1.Data[16]	0		Decimal	SI

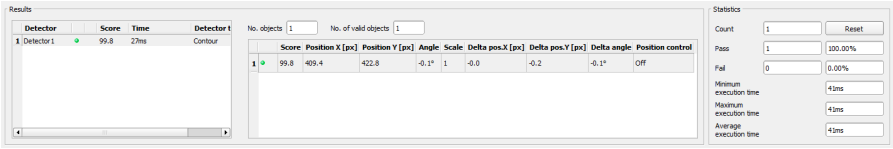
Fig. 69: Output data

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).







10.3 PLC example programs

The following PLC example programs show some basic functions.

PLC example 1: Trigger when VISOR® ready

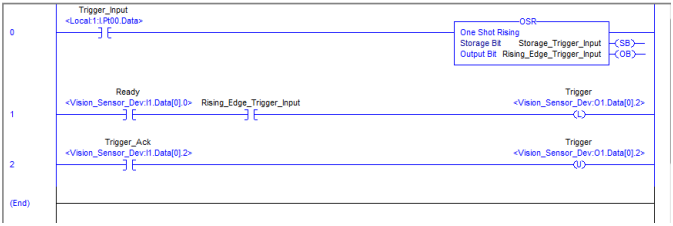


Fig. 70: Trigger when VISOR® ready (without error handling)

PLC example 2: Job change when VISOR® ready

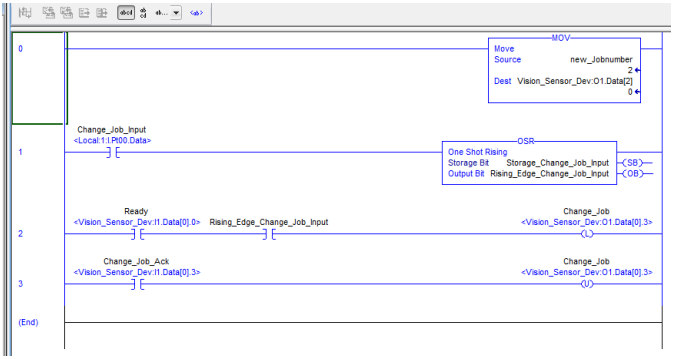


Fig. 71: Job change when VISOR® ready (without error handling)

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

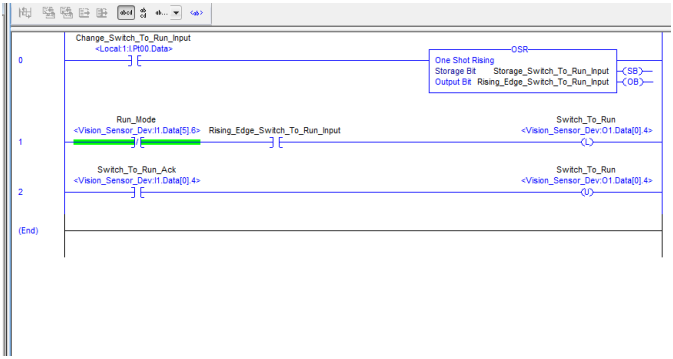


Fig. 72: Switch to Run when VISOR® in configuration mode (without error handling)

11 Appendix

11.1 Telegram, Data Output tab

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 "Data Output" of the setup "Output".

The following settings are possible:

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

11.1.1 Overview telegrams:

VISOR® General

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

VISOR® Check

- **Trigger** (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** (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.
- **Job change** (ASCII / Binary)
  The "Job change" telegram will trigger a job change on the VISOR® vision sensor.
- **Job change permanent** (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** (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 operating distance** (ASCII / binary)
  The "Auto operating distance" telegram can be used to have the operating distance for the job be automatically determined.
- **Set operating distance** (ASCII / binary)
  The "Set operating distance" telegram can be used to change the operating distance for the job.
- **Read operating distance** (ASCII / binary)
  The "Read operating distance" telegram can be used to read the current operating distance for the job.
- **Auto shutter speed** (ASCII / binary)
  The "Auto shutter speed" telegram can be used to have the shutter time for the job be automatically determined.
- **Set shutter speed** (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** (ASCII / Binary)
  With the telegram "Read shutter speed", the set shutter speed of the job can be read.
- **Set gain** (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** (ASCII / Binary)
  With the telegram "Read gain", the set gain of the job can be read.
- **Set parameter** (ASCII / Binary)
  With the telegram "Set parameter", the detector parameters can be adjusted, e.g. reference strings, detector thresholds.
- **Read parameter** (ASCII / Binary)
  With the telegram "Read parameter", the set parameters of the detectors can be read.
- **Set ROI** (ASCII / Binary)
  With the telegram "Set ROI", the position of the selected detector can be changed.
- **Read ROI** (ASCII / Binary)
  With the telegram "Read ROI", the position of the selected detector can be read.
- **Read job list** (ASCII / Binary)
  The "Read job list" telegram can be used to output a list of all available jobs on the VISOR® vision sensor.

- **Read detector list** (ASCII / Binary)
  With the telegram "Read detector list", a list of all detectors in the current job will be displayed.
- **Teach detector** (ASCII / Binary)
  The "Teach detector" telegram will result in the specified detector being re-taught (available only for Pattern matching and Contour comparison).
- **Set trigger delay** (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** (ASCII / Binary)
  With the telegram "Read trigger delay", the set delay for starting a trigger can be read.
- **Save job permanently** (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 point list** (ASCII / Binary)
  With the telegram "Calibration: Clear point list", the point list is initialized.
- **Calibration: Add world point** (ASCII / Binary)
  With the telegram "Calibration: Add world point", a world point (fiducial or point pair) of the point list is added. The telegram can be used for the calibration method Point pair list (Robotics) and Calibration plate (Robotics).
- **Calibration: Point list** (ASCII / Binary)
  With the telegram "Calibration: Point list", the calibration is carried out using the point list in the current job.
- **Calibration: Validate point list** (ASCII / Binary)
  With the telegram "Calibration: Validate point list", the calibration is validated using the point list.
- **Calibration: Calibration plate** (ASCII / Binary)
  With the telegram "Calibration: Calibration plate", the calibration is carried out using the calibration plate.
- **Set fiducial** (ASCII / Binary)
  With the telegram "Set fiducial", the fiducials are set using the point list in the current job.
- **Calibration: Copy calibration** (ASCII / Binary)
  With the telegram "Calibration: Copy calibration", the calibration of the current job is copied to the selected destination.
- **Calibration: Set parameter** (ASCII / Binary)
  With the telegram "Calibration: Set parameter", the parameter values for the calibration can be set.

- **Calibration: Read parameter** (ASCII / Binary)
  With the telegram "Calibration: Read parameter", the set parameter values of the calibration can be read.

VISOR® Visualization

- **Get image** (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 results** (ASCII)
  With the telegram "Update visualization results", the visualization results, e.g. image, detector information, and results, are updated.
- **Read sensor identity** (ASCII)
  With the telegram "Read sensor identity", the current firmware status as well as the hardware type can be queried.
- **Update firmware** (ASCII)
  With the telegram "Update firmware", a firmware update is started. The firmware file must first be loaded onto the VISOR® vision sensor.
- **Read job set** (ASCII)
  The "Import 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 "Back up job set" telegram can be used to read the VISOR® vision sensor's job set.

Data output

(ASCII / Binary)

This section provides information about the data output. In particular which format the individual results receive.

Telegrams: Availability and supported interfaces (Page 84)

11.1.2 Telegrams: Availability and supported interfaces

Telegramm	ALL			OB		CR			RO		Schnittstellen				Ab Version
	Standard	Advanced	Professional	Standard	Advanced	Standard	Advanced	Professional	Advanced	Professional	Ethernet TCP IN (2006)	PROFINET	EtherNet/IP	Service Port (1998)	
VISOR® Allgemein															
Statistik zurücksetzen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.18.
VISOR® Kontrolle															
Trigger	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0.
Erweiterter Trigger	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.6.
Jobwechsel	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0.
Jobwechsel Permanent	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.18.
Jobwechsel nach Jobname	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.0
VISOR® Jobeinstellungen															
Auto Fokus	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.0
Fokus setzen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.0
Fokus lesen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.0
Auto Verschlusszeit	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.0
Verschlusszeit setzen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0.
Verschlusszeit-wert lesen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0.
Verstärkung setzen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.6.
Verstärkungs-wert lesen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.6.
Parameter setzen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0.
Parameter lesen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0.
ROI setzen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0.
ROI lesen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0.
Jobliste lesen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.18.

Telegramm	ALL			OB		CR			RO		Schnittstellen				Ab Version
	Standard	Advanced	Professional	Standard	Advanced	Standard	Advanced	Professional	Advanced	Professional	Ethernet TCP IN (2006)	PROFINET	EtherNet/IP	Service Port (1998)	
VISOR® Jobeinstellungen															
Detektorliste lesen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.18.
Detektor ein-lernen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0.
Trigger-Verzö-gerung setzen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.22.
Trigger-Verzö-gerung lesen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.22.
Speichern Job Permanent	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.0
VISOR® Kalibrierung															
Punkteliste initialisieren		✓	✓						✓	✓	✓	✓	✓		1.18.
Weltpunkt hinzufügen		✓	✓						✓	✓	✓	✓	✓		1.22.
Kalibrierung Punkteliste		✓	✓						✓	✓	✓	✓	✓		1.18.
Kalibrierung validieren		✓	✓						✓	✓	✓	✓	✓		1.18.
Kalibrierung Kalibrierplatte	(✓)	(✓)	✓		(✓)				✓	✓	✓	✓	✓		1.19.
Referenzmar-ken setzen			✓						✓	✓	✓	✓	✓		1.22.
Kalibrierung kopieren			✓						✓	✓	✓	✓	✓		1.19.
Parameter setzen	(✓)	(✓)	✓		(✓)				✓	✓	✓	✓	✓		1.22.
Parameter lesen	(✓)	(✓)	✓		(✓)				✓	✓	✓	✓	✓		1.22.
VISOR® Visualisierung															
Bild holen	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				1.0.

Telegramm	ALL			OB		CR			RO		Schnittstellen				Ab Version
	Standard	Advanced	Professional	Standard	Advanced	Standard	Advanced	Professional	Advanced	Professional	Ethernet TCP IN (2006)	PROFINET	EtherNet/IP	Service Port (1998)	
VISOR® Service															
Update Visualisierungsdaten	✓	✓	✓		✓		✓	✓	✓	✓				✓	1.22.
Sensoridentität lesen	✓	✓	✓		✓		✓	✓	✓	✓				✓	1.19.
Firmware aktualisieren	✓	✓	✓		✓		✓	✓	✓	✓				✓	1.19.
Jobsatz einlesen	✓	✓	✓		✓		✓	✓	✓	✓				✓	1.19.
Jobsatz sichern	✓	✓	✓		✓		✓	✓	✓	✓				✓	1.19.

Overview telegrams: (Page 80)

11.1.3 Error codes

Error code	Description
000	Successful
001	Error
003	Invalid parameter data
005	Invalid telegram
006	Input parameters with invalid size or invalid value
007	File does not exist
008	Recorder off
009	Matching image of requested type not found
010	Invalid file name or length
011	Invalid data length
012	Not allowed due to job set mismatch
013	Failed to start new job from job set
016	Firmware version mismatch
018	Calibration plate data not available
020	More than one vis file present
021	Sensor type does not match for vis file
030	Calibration not enabled
031	Calibration copy error
032	Mismatched input conditions for destination job
033	Calibration / validation error
034	Invalid number of points
035	Calibration error: Add point, e.g. last job result failed
036	Invalid fiducial

Error code	Description
037	Job set protection error: "Permanent" job change is not allowed
038	Parameter values are not available to write / read
039	Sensor is in configuration mode. The telegram was rejected
040	Write / read error for parameter value
041	No matching job found

11.1.4 Serial communication ASCII

Reset statistic (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Reset statistics (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	R	Reset Statistics
2	S	
3	T	
Example:	RST	
Reset statistics (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	R	Reset Statistics
2	S	
3	T	
4	P F	P successful (Pass) F Fail
Example:	RSTP	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		Telegrams: Availability and supported interfaces

	(Page 84)
End of telegram:	Max. 4 bytes (optional)

Trigger (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Trigger (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	T	Trigger (simple trigger, in-port)
2	R	
3	G	
Example:	TRG	
Trigger (ASCII) response string from sensor		
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 F	P successful (Pass) F Fail
Example:	TRGP	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		Yes
Accepted when Ready is low:		No
Status of Ready signal during processing:		Low
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
End of telegram:		Max. 4 bytes (optional)

Extended trigger (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Extended trigger (ASCII) request string to sensor		
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 (ASCII) response string from sensor		
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 successful (Pass) F Fail
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 config mode:	Yes
Accepted when Ready is low:	No
Status of Ready signal during processing:	Low
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
End of telegram:	Max. 4 bytes (optional)

Job change (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Job change (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	C	Change Job
2	J	
3	B	
4 - 6	X	Job number
Example:	CJB005	
Job change (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	C	Change Job
2	J	
3	B	
4	P F	P successful (Pass) F Fail
5	T F	Triggered Freerun
6 - 8	X	Job number
Example 1: Example 2:	CJBPT005 CJBFF005	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes

Status of Ready signal during processing:	Low
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
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 84)

Overview telegrams: (Page 80)

Job change, permanent (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	C	Change Job Permanent
2	J	
3	P	
4 - 6	X	Job number
Example:	CJP005	
Job change, permanent (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	C	Change Job Permanent
2	J	
3	P	
4	P F	P successful (Pass) F Fail
5	T F	Triggered Freerun
6 - 8	X	Job number
Example 1: Example 2:	CJPPT005 CJPFF005	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during		Low

processing:	
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
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 84)

Overview telegrams: (Page 80)

Job change by job name (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	C	Change Job by Name
2	J	
3	N	
4	X	Request version
5 - 7	X	Job name length
8 - n	X	Job name
Example:	CJN1005Myjob	
Job change by job name (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	C	Change Job by Name
2	J	
3	N	
4	P F	P successful (Pass) F Fail
5 - 7	X	Error codes (Page 87)
8	X	Trigger mode T - Trigger F - Free run
Example:	CJNP000T	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No

Accepted when Ready is low:	Yes
Status of Ready signal during processing:	Low
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
End of telegram:	Max. 4 bytes (optional)

Auto operating distance (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Auto operating distance (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	A	Auto Focus
2	F	
3	C	
4	1	Request version
5	X	Unit 0 - 1/1000 millimeters 4 - steps
6	X	Operating distance selection 0 - False 1 - True
	NOTE: The following byte sequence is only relevant if "Operating distance selection" has been set to 1.	
7 - 9	X	Operating distance 0 - Maximum score 1 - Min. Operating distance 2 - Max. operating distance 3 - Operating distance average 4 - Operating distance median 5 - Maximum score and all planes
10	X	Distance range selection 0 - False 1 - True
	NOTE: The following byte sequence is only relevant if "Distance range selection" has been set to 1.	
11 - 18	X	Close distance

19 - 26	X	Far distance
Example:	Example 1: AFC1000 Example 2: AFC1410051000000000000003450	
Auto operating distance (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	A	Auto Focus
2	F	
3	C	
4	P F	P successful (Pass) F Fail
5 - 7	X	Error codes (Page 87)
8 - 10	X	Number of operating distances
	NOTE: The following byte sequence is repeated for each operating distance. The byte numbers shift accordingly.	
11 - 18	X	Operating distance
19 - 26	X	Score
Example:	AFCP000002000000950000009000093000000089000	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
End of telegram:		Max. 4 bytes (optional)

Set operating distance (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Set operating distance (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	S	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	Operating distance
Example:	SFC111400000010	
Set operating distance (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	S	Set Focus
2	F	
3	C	
4	P F	P successful (Pass) F Fail
5 - 7	X	Error codes (Page 87)

8 - 15	X	Current operating distance
Example:	SFCP00000000050	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
End of telegram:		Max. 4 bytes (optional)

Read operating distance (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read operating distance (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	G	Get Focus
2	F	
3	C	
4	1	Request version
5	X	Unit 0 - 1/1000 millimeters 4 - steps
Example:	GFC10	
Read operating distance (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	G	Get Focus
2	F	
3	C	
4	P F	P successful (Pass) F Fail
5 - 7	X	Error codes (Page 87)
8 - 15	X	Operating distance
Example:	GFCP00000092500	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes

Status of Ready signal during processing:	No change
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
End of telegram:	Max. 4 bytes (optional)

Auto shutter speed (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Auto shutter speed (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	A	Auto shutter
2	S	
3	H	
4	1	Request version
Example:	ASH1	
Auto shutter speed (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	A	Auto shutter
2	S	
3	H	
4	P F	P successful (Pass) F Fail
5 - 7	X	Error codes (Page 87)
8 - 15	X	Auto shutter speed value
16 - 23	X	Score
Example:	ASHP00000001780000057500	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during		Low

processing:	
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
End of telegram:	Max. 4 bytes (optional)

Set shutter speed (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Set shutter speed (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	S	Set Shutter Speed
2	S	
3	P T	Permanent Temporary Permanent change affect all parameters, including those that were previously only changed temporarily.
4 - 5	X	Number of digits of the shutter speed value, e.g. 04
6 - 9	X	New shutter speed value in microseconds, e.g. 8000 = 8 ms
Example:	SSP048000	
Set shutter speed (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	S	Set Shutter Speed
2	S	
3	P T	Permanent Temporary Permanent change affect all parameters, including those that were previously only changed temporarily.
4	P F	P successful (Pass) F Fail
Example:	SSPP	
Additional information:		
Accepted in run mode:	Yes	

Accepted in config mode:	No
Accepted when Ready is low:	Yes
Status of Ready signal during processing:	Low
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
End of telegram:	Max. 4 bytes (optional)

Read shutter speed (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read shutter speed value (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	G	Get Shutter Speed (from active job)
2	S	
3	H	
Example:	GSH	
Read shutter speed value (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	G	Get Shutter Speed
2	S	
3	H	
4	P F	P successful (Pass) F Fail
5	X	Shutter speed value, length
6 ... n	X	Shutter speed value
Example Run Mode: Example Configuration Mode:	GSHP41200 GSHF0	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		Telegrams: Availability and supported

	interfaces (Page 84)
End of telegram:	Max. 4 bytes (optional)

Set gain (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Set gain (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	S	Set Gain
2	G	
3	A	
4	0 1	0: Temporary 1: Permanent Permanent change affect all parameters, including those that were previously only changed temporarily.
5 - 9	X	New gain value (in value * 1000), e.g. 2.0 = 02000
Example:	SGA102000	
Set gain (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	S	Set Gain
2	G	
3	A	
4	P F	P successful (Pass) F Fail
5 - 9	X	Current gain value
Example:	SGAP02000	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is		Yes

low:	
Status of Ready signal during processing:	No change
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
End of telegram:	Max. 4 bytes (optional)

Read gain value (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read gain value (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	G	Get Gain
2	G	
3	A	
Example:	GGA	
Read gain value (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	G	Get Gain
2	G	
3	A	
4	P F	P successful (Pass) F Fail
5 - 9	X	Current gain value (value *1000), e.g. 1.0 = 01000
Example:	GGAP01000	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
End of telegram:		Max. 4 bytes (optional)

Set parameter (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Set parameter (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	S	Set Parameter
2	P	
3	P T	P Permanent T Temporary Permanent change affect all parameters, including those that were previously only changed temporarily.
4 - 6	X	Detector number
7 - 9	X	Command: Reference string / set value *1) see below
10 - 14	X	Length of reference string / value in Bytes (n)
15 ... n	X	Reference string / value
Example:	SPP0010010044196	
Set parameter (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	S	Set Parameter
2	P	
3	P T	P Permanent T Temporary Permanent change affect all parameters, including those that were previously only changed temporarily.
4	P F	P Pass F Fail

5	S	SI08 - Signed Integer 08 UI08 - Unsigned Integer 08 SI16 - Signed Integer 16 UI16 - Unsigned Integer 16
6	T	SI32 - Signed Integer 32 UI32 - Unsigned Integer 32 SI40 - Signed Integer 40 UI40 - Unsigned Integer 40
7	R	FLOT - Float DOBL - Double STRG - String BOOL - Boolean
8	G	SP08 - Special Signed 8 UDEF - Undefined IARR - Integer Array ZERO - Default Zero Parameter
Example:	SPPPSTRG	
Additional information:		
Accepted in run mode:	Yes	
Accepted in config mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	Low	
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)	
End of telegram:	Max. 4 bytes (optional)	

*1) Byte No. 7: Command: Set reference string

Detector	Function	Command
Alignment Pattern matching	Threshold Min Threshold Max ResultOffsetPos_X	001 002 031

Detector	Function	Command
	ResultOffsetPos_Y ResultOffsetAngle	032 033
Alignment Contour comparison	Threshold Min Threshold Max ResultOffsetPos_X ResultOffsetPos_Y ResultOffsetAngle	001 002 031 032 033
Alignment Edge detection	Transition probe 1 Transition probe 2 Transition probe 3 Score min probe 1 Score min probe 2 Score min probe 3	101 102 103 104 105 106
Pattern matching	Threshold Min Threshold Max ResultOffsetPos_X ResultOffsetPos_Y ResultOffsetAngle	001 002 031 032 033
Contour comparison	Threshold Min Threshold Max ResultOffsetPos_X ResultOffsetPos_Y ResultOffsetAngle	001 002 031 032 033
Grayscale	Threshold Min Threshold Max GrayMin GrayMax GrayInvert	001 002 101 102 103
Contrast	Threshold Min Threshold Max	001 002
Brightness	Threshold Min Threshold Max	001 002
Barcode	Reference String	101
Data code	Reference String	101
OCR	Reference String	101
Color value	ColorMinChannel1	101

Detector	Function	Command
	ColorMaxChannel1 ColorInvertChannel1 ColorMinChannel2 ColorMaxChannel2 ColorInvertChannel2 ColorMinChannel3 ColorMaxChannel3 ColorInvertChannel3	102 103 104 105 106 107 108 109
Color area	ColorMinChannel1 ColorMaxChannel1 ColorInvertChannel1 ColorMinChannel2 ColorMaxChannel2 ColorInvertChannel2 ColorMinChannel3 ColorMaxChannel3 ColorInvertChannel3	101 102 103 104 105 106 107 108 109
BLOB	GrayAbsoluteMin GrayAbsoluteMax GrayAbsoluteInvert	101 102 103

Read parameter values (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read parameter (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	G	Get Parameter Value
2	P	
3	A	
4 - 6	X	Detector number e.g. 001
7 - 9	X	Command: Read reference string / value *1) see below
Example:	GPA001001	
Read parameter (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	G	Get Parameter Value
2	P	
3	A	
4	P F	P successful (Pass) F Fail

5	S	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
6	T	
7	R	
8	G	
9 - 13	X	Length of reference strings / value (n) e.g. 00005
14 ... n	X	Reference string / value
Example:	GPAPSTRG00005Test1	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
End of telegram:		Max. 4 bytes (optional)

*1) Byte No. 7: Command: Read reference string / value

Detector	Function	Command
Alignment Pattern matching	Threshold Min Threshold Max ResultOffsetPos_X ResultOffsetPos_Y ResultOffsetAngle	001 002 031 032 033
Alignment Contour comparison	Threshold Min Threshold Max ResultOffsetPos_X ResultOffsetPos_Y ResultOffsetAngle	001 002 031 032 033
Alignment Contour comparison	Transition probe 1 Transition probe 2 Transition probe 3 Score min probe 1 Score min probe 2 Score min probe 3	101 102 103 104 105 106
Pattern matching	Threshold Min Threshold Max ResultOffsetPos_X ResultOffsetPos_Y ResultOffsetAngle	001 002 031 032 033
Contour comparison	Threshold Min Threshold Max ResultOffsetPos_X ResultOffsetPos_Y ResultOffsetAngle	001 002 031 032 033
Grayscale	Threshold Min Threshold Max GrayMin GrayMax GrayInvert	001 002 101 102 103
Contrast	Threshold Min Threshold Max	001 002
Brightness	Threshold Min Threshold Max	001 002
Barcode	Reference String	101
Data code	Reference String	101

Detector	Function	Command
OCR	Reference String	101
Color value	ColorMinChannel1 ColorMaxChannel1 ColorInvertChannel1 ColorMinChannel2 ColorMaxChannel2 ColorInvertChannel2 ColorMinChannel3 ColorMaxChannel3 ColorInvertChannel3	101 102 103 104 105 106 107 108 109
Color area	ColorMinChannel1 ColorMaxChannel1 ColorInvertChannel1 ColorMinChannel2 ColorMaxChannel2 ColorInvertChannel2 ColorMinChannel3 ColorMaxChannel3 ColorInvertChannel3	101 102 103 104 105 106 107 108 109
BLOB	GrayAbsoluteMin GrayAbsoluteMax GrayAbsoluteInvert	101 102 103

Set ROI (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Set ROI (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	S	Set ROI
2	R	
3	P T	P = Permanent T = Temporary Permanent change affect all parameters, including those that were previously only changed temporarily.
4 - 11	X	ROI info length in Bytes from Byte 4 to end e.g. 00000055
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 / Y-radius (value in pixels * 1000), e.g. 40 pixels = 00040000
51 - 58	X	Angle (not at circle / ellipse) (in ° * 1000), e.g. 180° = 00180000
Example:	SRP00000055001000200160000001200000008000000040000000180000 Length 55, detector= 1, yellow search range, rectangle, center X= 160, center Y= 120, half width= 80, half height= 40	
Set ROI (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	S	Set ROI
2	R	
3	P T	Permanent Temporary Permanent change affect all parameters, including those that were previously only changed temporarily.
4	P F	P successful (Pass) F Fail
Example:	SRPP	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		Low
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
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 ROI (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read ROI (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	G	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 for position control
Example:	GRI00100	
Read ROI (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	G	Get ROI
2	R	
3	I	
4	P F	P successful (Pass) F Fail
5 - 12	X	ROI info length in Bytes, from Byte 5 to end string
13 - 15	X	Detector no. e.g. 001
16 - 17	X	ROI Index = 00 for yellow search range

		= 01 for red teach range = 02 for 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 / Y-radius (value in pixels * 1000)
52 - 59	X	Angle (not at circle / ellipse) (value in ° * 1000)
Example:	GRIP000000550010002001600000012000000080000000400000009000 0 (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 config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		Low
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
End of telegram:		Max. 4 bytes (optional)

Read job list (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read job list (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	G	Get Job List
2	J	
3	L	
Example:	GJL	
Read job list (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	G	Get Job List
2	J	
3	L	
4	P F	P successful (Pass) F Fail
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 +	X	Number of subsequent bytes. A description

3		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 + 8	X	Date of creation of job n.
k + 9 ... k + 16	X	Date of last change of job n.
Example:	GJLP001001001007testjob010DefaultJob004Test2014112720141128	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
End of telegram:		Max. 4 bytes (optional)

Read detector list (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read detector list (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	G	Get Detector List
2	D	
3	L	
Example:	GDL	
Read detector list (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	G	Get Detector List
2	D	
3	L	
4	P F	P successful (Pass) F Fail
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 comparison 005 - Grayscale 006 - Contrast

		007 - Brightness 010 - Wafer 011 - OCR 013 - Data code 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 config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
End of telegram:		Max. 4 bytes (optional)

Teach detector (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Teach detector (ASCII) request string to sensor		
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 Permanent change affect all parameters, including those that were previously only changed temporarily.
8	X	0 = no trigger 1 = Trigger
Example:	TED00111	
Teach detector (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	T	Teach detector
2	E	
3	D	
4	P F	P successful (Pass) F Fail
Example:	TEDP	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No

Accepted when Ready is low:	Yes
Status of Ready signal during processing:	Low
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
End of telegram:	Max. 4 bytes (optional)

Set trigger delay (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Set trigger delay V1 (ASCII) request string to sensor		
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 V1 (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	S	Set Trigger Delay
2	T	
3	D	
4	P F	P successful (Pass) F Fail
5 - 7	X	Error codes (Page 87)
Example:	STDP000	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No

Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
End of telegram:		Max. 4 bytes (optional)

Read trigger delay (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read trigger delay V1 (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	G	Get Trigger Delay
2	T	
3	D	
4	1	Request version
Example:	GTD1	
Read trigger delay V1 (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	G	Get Trigger Delay
2	T	
3	D	
4	P F	P successful (Pass) F Fail
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 config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change

Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
End of telegram:		Max. 4 bytes (optional)

Save job permanently (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Save job permanently (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	S	Store Job Permanently
2	J	
3	P	
Example:	SJP	
Save job permanently (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	S	Store Job Permanently
2	J	
3	P	
4	P F	P successful (Pass) F Fail
Example:	SJPP	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		Low
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
End of telegram:		Max. 4 bytes (optional)

Calibration: Initialize point list (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Initialize point list (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	C	Calibration: Clear Data
2	C	
3	D	
Example:	CCD	
Initialize point list (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	C	Calibration: Clear Data
2	C	
3	D	
4	P F	P successful (Pass) F Fail
Example:	CCDP	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
End of telegram:		Max. 4 bytes (optional)

Calibration: Add world point (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Calibration: Add world point (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	C	Calibration: Add World Point
2	A	
3	W	
4	1	Request version
5	X	1: Fiducials only Calibration plate (Robotics) 4: World point and image point Point pair list (Robotics)
6 - 10	0	Constant (5 bytes)
11 - 18	X	World X (in mm * 1000)
19 - 26	X	World Y (in mm * 1000)
27 - 34	0	Constant (8 bytes)
Example:	CAW10000100100000002000000000000 (World X = 100 mm; World Y = 200mm)	
Calibration: Add world point (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	C	Calibration: Add World Point
2	A	
3	W	
4	P F	P successful (Pass) F Fail
5 - 7	X	Error codes (Page 87)
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 config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
End of telegram:		Max. 4 bytes (optional)

Calibration: Point list (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Calibration with point list (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	C	Calibration: Calibrate by Point List
2	C	
3	L	
4	X	0: Temporary 1: Permanent
Example:	CCL1	
Calibration with point list (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	C	Calibration: Calibrate by Point List
2	C	
3	L	
4	P F	P successful (Pass) F Fail
5 - 9	X	Current highest point pair index
10 - 17	X	Deviation calibration, RMSE (root mean square error)
18 - 25	X	Deviation calibration, mean
26 - 33	X	Deviation calibration, max.
34 - 41	X	Deviation calibration, min.
Example:	CCLP000101234567812345678123456781234567812345678	
Additional information:		
Accepted in run mode:		Yes

Accepted in config mode:	No
Accepted when Ready is low:	Yes
Status of Ready signal during processing:	No change
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
End of telegram:	Max. 4 bytes (optional)

Calibration: Validate point list (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Calibration: Validate point list (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	C	Calibration: Validate by Point List
2	V	
3	L	
Example:	CVL	
Calibration: Validate point list (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	C	Calibration: Validate by Point List
2	V	
3	L	
4	P F	P successful (Pass) F Fail
5 - 9	X	Current highest point pair index
10 - 17	X	Deviation calibration, RMSE (root mean square error)
18 - 25	X	Deviation calibration, mean
26 - 33	X	Deviation calibration, max.
34 - 41	X	Deviation calibration, min.
Example:	CVLP000101234567812345678123456781234567 8	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No

Accepted when Ready is low:	Yes
Status of Ready signal during processing:	No change
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
End of telegram:	Max. 4 bytes (optional)

Calibration: Calibrate plate (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Calibration: Calibrate plate (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	C	Calibration: Calibrate by Plate
2	C	
3	P	
4	1	Request version
5	X	0: Temporary 1: Permanent Permanent change affect all parameters, including those that were previously only changed temporarily.
6	X	0: No fiducials are used. The origin of the measurement coordinate system is identical to the origin of the calibration coordinate system. 1: No fiducials are used. Measurement coordinate system is identical to device 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 measurement coordinate system
Example:	CCP1110	
Calibration: Calibrate plate (ASCII) response string from sensor		
Byte no.	Content	Meaning

1	C	Calibration: Calibrate by Plate
2	C	
3	P	
4	P F	P successful (Pass) F Fail
5 - 7	X	Error codes (Page 87)
8 - 12	X	Current number of detected calibration points
13 - 20	X	Deviation calibration, RMSE (root mean square error)
21 - 28	X	Deviation calibration, mean
29 - 36	X	Deviation calibration, max.
37 - 44	X	Deviation calibration, min.
45 - 52	X	X delta (in user unit * 1000)
53 - 60	X	Y delta (in user unit * 1000)
61 - 68	0	Reserved
69 - 76	0	Reserved
77 - 84	0	Reserved
85 - 92	X	Delta Gamma (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 config mode:		No

Accepted when Ready is low:	Yes
Status of Ready signal during processing:	No change
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
End of telegram:	Max. 4 bytes (optional)

Calibration: Set fiducial (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Calibration: Set fiducial (ASCII) request string to sensor		
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 (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	C	Calibration: Set Fiducial
2	S	
3	F	
4	P F	P successful (Pass) F Fail
5 - 7	X	Error codes (Page 87)
8 - 15	X	X value
16 - 23	X	Y value
24 - 31	X	Z value
32 - 39	X	Red X value
40 - 47	X	Red Y value
48 - 55	X	Red Z value
56 - 63	X	Deviation fiducials, mean

64 - 71	X	Deviation fiducials, max.
72 - 79	X	Deviation fiducials, min.
Example:	CSFP 000 00001001 00002002 00003003 00004004 00005005 00006006 00001001 0000202 00003003	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
End of telegram:		Max. 4 bytes (optional)

Calibration: Copy calibration (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Calibration: Copy calibration (ASCII) request string to sensor		
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 (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	C	Calibration: Copy Calibration
2	C	
3	C	
4	P F	P successful (Pass) F Fail
5 - 7	X	Error codes (Page 87)
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 config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
End of telegram:		Max. 4 bytes (optional)

Calibration: Set parameter (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Calibration: Set parameter (ASCII) request string to sensor		
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 (Page 153)
9 - 16	X	Length of the following data
17 ... n	X	Parameter value
Example:	CSP11002000000019	
Calibration: Set parameter (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	C	Calibration: Set Parameter
2	S	
3	P	
4	P F	P successful (Pass) F Fail
5 - 7	X	Error codes (Page 87)
Example:	CSPP000	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No

Accepted when Ready is low:	Yes
Status of Ready signal during processing:	No change
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
End of telegram:	Max. 4 bytes (optional)

Parameter name	**Parameter number**	**Parameter value**	**Length**	**Format**	**Status**
Unit (user unit)	004	0: mm 1: cm 2: m 3: inch 4: au	1 Byte	Characters	no change
Z Offset	021	(in user unit * 1000)	8 Byte	Signed integer	no change
Focal length	022	(in mm *1000)	8 Byte	Signed integer	invalid
Fiducial 1	024	in each case: 0: X value (in user unit * 1000) 1: Y value (in user unit * 1000) 2: Z value (in user unit * 1000)	8 bytes per value	Signed integer	no change
Fiducial 2	025				
Fiducial 3	026				
Fiducial 4	027				

Calibration: Read parameter (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Calibration: Read parameter (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	C	Calibration: Get Parameter
2	G	
3	P	
4	1	Request version
5 - 7	X	Parameter number (Page 155)
Example:	CGP1001	
Calibration: Read parameter (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	C	Calibration: Get Parameter
2	G	
3	P	
4	P F	P successful (Pass) F Fail
5 - 7	X	Error codes (Page 87)
8 - 10	X	Parameter number (Page 155)
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 config mode:		No

Accepted when Ready is low:	Yes
Status of Ready signal during processing:	No change
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
End of telegram:	Max. 4 bytes (optional)

Parameter name	Parameter number	Value	Length	Format
Status calibration	001	0: Invalid 1: Valid	1 Byte	Signed integer
Calibration method	002	0: None 1: Scaling (Measurement) 2: Point pair list (Robotics) 3: Calibration plate (Measurement) 4: Calibration plate (Robotics)	1 Byte	Characters
Unit (user unit)	004	0: mm 1: cm 2: m 3: inch 4: au	1 Byte	Characters
Internal sensor parameters	010	0: Focal length (in mm * 1000) 1: Kappa value (*1000) 2: X pixel pitch (in µm * 1000) 3: Y pixel pitch (in µm * 1000) 4: X coordinate origin (in pixels * 1000) 5: Y coordinate origin (in pixels * 1000)	8 bytes per value	Signed integer

Parameter name	Parameter number	Value	Length	Format
		6: Image size X (pixels) 7: Image size Y (pixels)		
External parameters	011	0: X axis offset (in user unit * 1000) 1: Y axis offset (in user unit * 1000) 2: Z axis offset (in user unit * 1000) 3: X rotation (in degrees * 1000) 4: Y rotation (in degrees * 1000) 5: X rotation (in degrees * 1000)	8 bytes per value	Signed integer
Transformation calibration coordinate system	012	0: X axis offset (in user unit * 1000) 1: Y axis offset (in user unit * 1000) 2: Z axis offset (in user unit * 1000) 3: X rotation (in degrees * 1000) 4: Y rotation (in degrees * 1000) 5: X rotation (in degrees * 1000)	8 bytes per value	Signed integer
Transformation measurement coordinate system	014	0: X axis offset (in user unit * 1000) 1: Y axis offset (in user unit * 1000) 2: Z axis offset (in user unit * 1000) 3: X rotation (in degrees * 1000) 4: Y rotation (in degrees * 1000) 5: X rotation (in degrees * 1000)	8 bytes per value	Signed integer

Parameter name	Parameter number	Value	Length	Format
Z Offset	021	Value in user unit (see unit * 1000)	8 Byte	Signed integer
Focal length	022	(in mm *1000)	8 Byte	Signed integer
Fiducial 1	024	in each case: 0: X value (in user unit * 1000) 1: Y value (in user unit * 1000) 2: Constant 0	8 bytes per value	Signed integer
Fiducial 2	025			
Fiducial 3	026			
Fiducial 4	027			

Get image (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Get image (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	G	Get Image
2	I	
3	M	
4	X	0 - Last image 1 - Last poor image 2 - Last good image
Example:	GIM1	
Get image (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	G	Get Image
2	I	
3	M	
4	P F	P successful (Pass) F Fail
5	X	Error codes (Page 87)
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 message string
20 ... n	X	Binary image data (rows * columns)
Example:	GIMP0004800640...	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		Low
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
End of telegram:		Max. 4 bytes (optional)

Update visualization data (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Update visualization data (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	U	Update Visualization Results
2	V	
3	R	
4	1	Request version
5	X	Image: 0 - No image is being created 1 - Gray value image / RGB image is being created 2 - Gray value image / Bayerpattern is being created
6	X	Result XML: 0 - Result file will be not be created 1 - Result file is being created
7	X	Statistic XML: 0 - Statistic file will not be created 1 - Statistic file is being 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 000: visu000
Example:	UVR11110000	
Update visualization data (ASCII) response string from sensor		
Byte no.	Content	Meaning

1	U	Update Visualization Results
2	V	
3	R	
4	P F	P successful (Pass) F Fail
5 - 7	X	Error codes (Page 87)
8	X	Data available: 0: New data available when ready.txt is written 1: No new data available.
9 - 11	X	Directory number 000: visu000
Example:	UVRP0000000	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
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	Detector type

Name		Value	Description
		contrast brightness gray caliper blob ocr data code barcode	
	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 comparison	Yes	Yes	Yes	200
Contrast	Yes	No	No	0
Brightness	Yes	No	No	0
Grayscale	Yes	No	No	0
Caliper	Yes	No	No	0
BLOB	Yes	No	No	1000
OCR	Yes	No	No	1
Data code	Yes	No	No	5
Barcode	Yes	No	No	5

Read sensor identity (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read sensor identity (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	G	Read sensor identity
2	S	
3	I	
4	1	Request version
Example:	GSI1	
Read sensor identity (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	G	Read sensor identity
2	S	
3	I	
4	P F	P Pass F Fail
5 - 7	X	Error codes (Page 87)
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:	GSIP0000221.19.3.2; [[[Nicht definierte Variablevisor_object.PD V20-OB-A2-W12]]]	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes

Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
End of telegram:	Max. 4 bytes (optional)

Update firmware (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Update firmware (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	U	Update Firmware
2	F	
3	W	
4	1	Request version
Example:	UFW1	
Update firmware (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	U	Update Firmware
2	F	
3	W	
4	P F	P Pass F Fail
5 - 7	X	Error codes (Page 87)
Example:	UFWP000	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is low:		Yes
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)
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 84)

Overview telegrams: (Page 80)

Import job set (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	S	Read job set
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.bjs" is used.
Example:	SJS1012myjobset.bjs	
Import job set (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	S	Read job set
2	J	
3	S	
4	P F	P Pass F Fail
5 - 7	X	Error codes (Page 87)
8 - 10	X	Active job number in the loaded job set
Example:	SJSP000001	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No

Accepted when Ready is low:	No
Status of Ready signal during processing:	Low
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
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.

Get job set (ASCII)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Back up job set (ASCII) request string to sensor		
Byte no.	Content	Meaning
1	G	Get job set
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.bjs" is used.
Example:	GJS1012myjobset.bjs	
Back up set (ASCII) response string from sensor		
Byte no.	Content	Meaning
1	G	Get job set
2	J	
3	S	
4	P F	P Pass F Fail
5 - 7	X	Error codes (Page 87)
Example:	GJSP000	
Additional information:		
Accepted in run mode:		Yes
Accepted in config mode:		No
Accepted when Ready is		Yes

low:	
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)
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.

Data output in ASCII

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Output data (ASCII), dynamically put together based on the user settings in the software: SensoConfig/Output/Data output.

Basic string construction:

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

Output data (ASCII):

<OPTIONAL FIELDS>				
Parameter	**Description**	**Length ASCII [Byte]**	**Data type**	**Available for**
Selected fields	This checkbox shows all selected fields. The checkbox for selected fields is not shown.	16	ASCII: The output order is from left to right and from top to bottom, i.e., for each checkbox that is enabled, a byte is set beginning with the LSB. Binary: The output order is from left to right and from top to bottom, i.e., for each checkbox that is enabled, a bit is set beginning with the least significant one.	All types
Telegram		1 ... 10	ASCII: E.g., output string with 10	All types

<OPTIONAL FIELDS>				
Parameter	**Description**	**Length ASCII [Byte]**	**Data type**	**Available for**
length	Number of characters, incl. the characters for the telegram length itself.		characters; telegram length: 10 + 2 characters (one byte per whole number place) = 12	
Status byte	Represents the trigger mode.	3	ASCII: PPF = Trigger; PFP = Free run Binary: 0X06 0x00 = Trigger mode 0X05 0x00 = Free run mode	All types
Detector results	Output of overall result for each detector.	4 ... 261	ASCII: Byte 1 = AND conjunction of all detectors Byte 2 = Overall position tracking result Byte 3 = Overall current job result Followed by the number of detectors; one byte per whole number 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	ASCII: Byte 1 Number of active outputs (logic gate result assigned)	All types

<OPTIONAL FIELDS>				
Parameter	**Description**	**Length ASCII [Byte]**	**Data type**	**Available for**
			Followed by bytes 2 – 7; one byte per output P = Detector pass F = Detector fail 0 = Inactive output (gap between two active outputs) Binary: Bytes 1 and 2: Number of active Outputs Bytes 3 – n: Outputs, bit-coded	
Log. Outputs	Returns the logic gate result for each logic output.	1 ... 259	ASCII: Starting from byte 1 Number of active outputs (logic gate result assigned); 1 byte per whole number place Following bytes: One byte per logic output P = Detector pass F = Detector fail 0 = Inactive output (gap between two active outputs) Binary:	All types

<OPTIONAL FIELDS>				
Parameter	**Description**	**Length ASCII [Byte]**	**Data type**	**Available for**
			Byte 1 and byte 2: Number of active log. Outputs Byte 3 – n All active logic outputs,	
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

Value	Description	ASCII length [bytes]	Data type	Available for
"All evaluations" counter	Total number of checks	1 ... 11	Signed integer	GENERAL
Pass parts counter	Number of checks with "OK" result	1 ... 11	Signed integer	GENERAL
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	Boolean	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

Value	Description	ASCII length [bytes]	Data type	Available for
Score	[%]	1 ... 6	Signed integer	All detectors
Overall result	Boolean detector result	1	Boolean	All detectors
Execution time	Execution time of individual detector in [msec].	1 ... 11	Signed integer	All detectors

Position

Value	Description	ASCII length [bytes]	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	

Value	Description	ASCII length [bytes]	Data type	Available for
				AF
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	
Angle	Orientation of found object, 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) [°],	1 ... 7	Signed integer	

Value	Description	ASCII length [bytes]	Data type	Available for
	value range: -180° to 180° 0° = East, counterclockwise			
Delta angle	Angle between taught object and found object, 1/1000 [°]	1 ... 7	Signed integer	
Position check		1	Boolean	

Measurement

Value	Description	ASCII length [bytes]	Data type	Available for
Height	Height of geometric element [user unit], Height ≥ 0, height ≤ width	1 ... 11	Signed integer	

Value	Description	ASCII length [bytes]	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 with holes, 1/1000 [pixels]	1 ... 11	Signed integer	
Distance	Calculated distance [user unit]	1 ... 11	Signed integer	

Identification

Value	Description	ASCII length [bytes]	Data type	Available for
String length	Length of read code [bytes]	1 ... 3	Signed integer	
String	Contents of read code. Depending on the code, the string length may vary. If a fixed string length is desired, the minimum string length (detector-specific user data) and the maximum string length (detector settings) must be set to the same value (e.g. 127).	0 ... 255	String	
String comparison	Content check for the read information. The contents of the read information are checked using regular expressions (Reiter Referenzstring)	1	Boolean	
Truncated	Code complete or truncated F = Code complete P = Code truncated	1	Boolean	

Identification - quality

Value	Description	ASCII length [bytes]	Data type	Available for
Quality - overall	Evaluation of printing quality according to international standard ISO/IEC 15416. The overall quality indicates the final rating. [x].			
Quality - individual	Evaluation of printing quality according to international standard ISO/IEC 15416. Printing quality rating.			
Min. Quality	Used to check whether the minimum required quality is being met			

Color

Value	Description	ASCII length [bytes]	Data type	Available for
Color value: • 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

Value	Description	ASCII length [bytes]	Data type	Available for
Number of objects	Maximum number of objects that must be detected [units]	1 ... 5	Signed integer	
Number of valid objects	Used to check whether the number of objects found falls within a specific range. [Good/Bad or 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	 (only Edge detection) 
Number of valid search stripes	Used to check whether the number of search stripes found falls within a specific range. [Good/Bad or units]	1 ... 3	Signed integer	 (only Edge detection) 
Too many BLOBs		1	Boolean	

Extended

Value	Description	ASCII length [bytes]	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	3 ... 4		(only Contour comparison)
Eccentricity	Numerical eccentricity Value range of 0.0 to 1.0	n	Signed integer	
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		A
Reference string met	The output string matches the reference string.	1	Boolean	A
Contrast	Code contrast Value range of 0 to 100 [%]	n		
Correction	Number of modules corrected by error corrections	n		

Value	Description	ASCII length [bytes]	Data type	Available for
	[units]			
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	
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				
Max. signal threshold				
Inverted signal threshold				

Value	Description	ASCII length [bytes]	Data type	Available for
Deviation, inside	Returns the largest deviation between the BLOB contour and the contour of the geometric element (deviation inside the fitted circle). [user unit]			
Deviation, outside	Returns the largest deviation between the BLOB contour and the contour of the geometric element (deviation outside the fitted circle). [user unit]			
Deviation, mean	Returns the mean of the absolute "inside" and "outside" deviation values between the BLOB contour and the contour of the geometric element.			
Axial ratio	Ratio of the long to the short axis (a / b)			
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				

<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**
Trailer	User-defined, up to a max. of 8 characters	0 ... 8	Unsigned Int	All types

11.1.5 Serial communication BINARY

Reset statistic (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Reset statistics (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x05	Telegram length
5	Unsigned Char	0x04	Reset Statistics
Reset statistics (binary) response string from sensor			
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 87)
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			Low
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

Trigger (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Trigger (binary) request string to sensor			
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 (binary) response string from sensor			
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 87)
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			Yes
Accepted when Ready is low:			No
Status of Ready signal during processing:			Low
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

Extended trigger (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Extended trigger (binary), request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length
5	Unsigned Char	0x13	Extended trigger command, trigger with index, for assigning triggers to corresponding result data, via port 2006)
6	Unsigned Char	0xXX	Length of following data (n)
7 ... n	Unsigned Char	0xXX	Data
Extended trigger (binary) response string from sensor			
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 87)
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 config mode:			Yes
Accepted when Ready is low:			No
Status of Ready signal during processing:			Low
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

Job change (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Job change (binary) request string to sensor			
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 (binary) response string from sensor			
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 87)
8	Unsigned Char	0xXX	Trigger Mode 0 = Triggered 1 = Freerun
9	Unsigned Char	0xXX	Job no. XX = 1 - n
Additional information:			
Accepted in run mode:		Yes	
Accepted in config mode:		No	
Accepted when Ready is low:		Yes	
Status of Ready signal during processing:		Low	
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)	

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 84)

Overview telegrams: (Page 80)

Job change, permanent (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x06	Telegram length
5	Unsigned Char	0x22	Change Job Permanent
6	Unsigned Char	0xXX	Job no. XX = 1 - n
Job change, permanent (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x09	Telegram length
5	Unsigned Char	0x22	Change Job Permanent
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
8	Unsigned Char	0xXX	Trigger Mode 0 = Triggered 1 = Freerun
9	Unsigned Char	0xXX	Job no. XX = 1 - n
Additional information:			
Accepted in run mode:		Yes	
Accepted in config mode:		No	
Accepted when Ready is low:		Yes	
Status of Ready signal during processing:		Low	
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)	

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 84)

Overview telegrams: (Page 80)

Job change by job name (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x07	Telegram length
5	Unsigned Char	0x2C	Job change by job name
6	Unsigned Char	0xXX	Request version
7	Unsigned Char	0xXX	Job name length
8 - n	Unsigned Char	0xXX	Job name
Job change by job name (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x0B	Telegram length
5	Unsigned Char	0x31	Job change by job name
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
8	Unsigned Char	0xXX	Trigger mode T - Trigger F - Free run
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			Low

Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)

Auto operating distance (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Auto operating distance (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x08	Telegram length
5	Unsigned Char	0x32	Auto Focus
6	Unsigned Char	0xXX	Request version
7	Unsigned Char	0xXX	Unit 0 - 1/1000 millimeters 4 - steps
8	Unsigned Char	0xXX	Operating distance selection 0 - False 1 - True
			NOTE: The following byte sequence is only relevant if "Operating distance selection" has been set to 1.
9	Unsigned Char	0xXX	Operating distance 0 - Maximum score 1 - Min. Operating distance 2 - Max. operating distance 3 - Operating distance average 4 - Operating distance median 5 - Maximum score and all planes
10	Unsigned Char	0xXX	Distance range selection 0 - False 1 - True
			NOTE: The following byte sequence is only relevant if "Distance range selection" has been set to 1.
11	Int	0xXX	Close distance

12	Int	0xXX	Far distance
Auto operating distance (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length
5	Unsigned Char	0x32	Auto Focus
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
8 - 11	Unsigned Int	0xXX	Number of operating distances
	NOTE: The following byte sequence is repeated for each operating distance. The byte numbers shift accordingly.		
12 - 15	Int	0xXX	Operating distance
16 - 19	Int	0xXX	Score
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

Set operating distance (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Set operating distance (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x0D	Telegram length
5	Unsigned Char	0x31	Set focus
6	Unsigned Char	0xXX	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	Operating distance
Set operating distance (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x0B	Telegram length
5	Unsigned Char	0x31	Set focus
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
8 - 11	Int	0xXX	Current operating distance
Additional information:			

Accepted in run mode:	Yes
Accepted in config mode:	No
Accepted when Ready is low:	Yes
Status of Ready signal during processing:	No change
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)

Read operating distance (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read operating distance (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x7	Telegram length
5	Unsigned Char	0x30	Get Focus
6	Unsigned Char	0xXX	Request version
7	Unsigned Char	0xXX	Unit 0 - 1/1000 millimeters 4 - steps
Read operating distance (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x0B	Telegram length
5	Unsigned Char	0x31	Get Focus
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
8 - 11	Int	0xXX	Current operating distance
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

Auto shutter speed (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Auto shutter speed (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x06	Telegram length
5	Unsigned Char	0x07	Auto shutter speed
6	Unsigned Char	0xXX	Request version
Auto shutter speed (binary) response string from sensor			
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 87)
8 - 11	Int	0xXX	Auto shutter speed value
12 - 15	Int	0xXX	Score
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			Low
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

Set shutter speed (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Set shutter speed (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x09	Telegram length
5	Unsigned Char	0x0E 0x0F	Set shutter speed temporarily Set shutter speed permanently Permanent change affect all parameters, including those that were previously only changed temporarily.
6 - 9	Unsigned Int	0xXX	Shutter speed value
Set shutter speed (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x07	Telegram length
5	Unsigned Char	0x0E 0x0F	Set shutter speed temporarily Set shutter speed permanently Permanent change affect all parameters, including those that were previously only changed temporarily.
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			Low

Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)

Read shutter speed (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read shutter speed (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x05	Telegram length
5	Unsigned Char	0x17	Get Shutter Speed
Read shutter speed (binary) response string from sensor			
1 - 4	Unsigned Int	0x0B	Telegram length
5	Unsigned Char	0x17	Get Shutter Speed
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
8 - 11	Unsigned Int	0xXX	Shutter speed value
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode::			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

Set gain value (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Set gain (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x0A	Telegram length
5	Unsigned Char	0x1B	Set Gain
6	Unsigned Char	0xXX	0: Temporary 1: Permanent Permanent change affect all parameters, including those that were previously only changed temporarily.
7 - 10	Unsigned Int	0xXX	Gain value
Set gain (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x0B	Telegram length
5	Unsigned Char	0x1B	Set Gain
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
8 - 11	Unsigned Int	0xXX	Current gain value (value *1000)
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

Read gain value (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read gain value (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x05	Telegram length
5	Unsigned Char	0x1C	Get Gain
Read gain value (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x0B	Telegram length
5	Unsigned Char	0x1C	Get Gain
6	Unsigned Short	0xXX	Error codes (Page 87)
7		0xXX	
8 - 11	Unsigned Int	0xXX	Current gain value
Additional information:			
Accepted in run mode:		Yes	
Accepted in config mode:		No	
Accepted when Ready is low:		Yes	
Status of Ready signal during processing:		No change	
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)	

Set parameter (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Set parameter (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length = 9 bytes + length of reference string (n)
5	Unsigned Char	0x05 0x06	Set parameter permanent Set parameter temporarily Permanent change affect all parameters, including those that were previously only changed temporarily.
6	Unsigned Char	0xXX	Detector no., XX = 1- n
7	Unsigned Char	0xXX	Command reference string / Set value *1), see below !
8 - 9	Unsigned Short	0xXX	Length new reference string / value (n)
10 ... n	Unsigned Char	0xXX	Reference string / value
Set parameter (binary) response string from sensor (may arrive with a delay of up to 4–5 seconds)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x08	Telegram length
5	Unsigned Char	0x05 0x06	Set parameter permanent Set parameter temporary Permanent change affect all parameters, including those that were previously only changed temporarily.
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)

8	Unsigned Char	0xXX	Parameter type string
Additional information:			
Accepted in run mode:		Yes	
Accepted in config mode:		No	
Accepted when Ready is low:		Yes	
Status of Ready signal during processing:		Low	
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)	

*1) Byte No. 7: Command: Set reference string

Detector	Function	Command	Length of the following data
Alignment Pattern matching	Threshold Min	1	4
	Threshold Max	2	4
	ResultOffsetPos_X	31	5
	ResultOffsetPos_Y	32	5
	ResultOffsetAngle	33	5
Alignment Contour comparison	Threshold Min	1	4
	Threshold Max	2	4
	ResultOffsetPos_X	31	5
	ResultOffsetPos_Y	32	5
	ResultOffsetAngle	33	5
Alignment Edge detection	Transition probe 1	101	4
	Transition probe 2	102	4
	Transition probe 3	103	4
	Score min probe 1	104	4
	Score min probe 2	105	4
	Score min probe 3	106	4

Detector	Function	Command	Length of the following data
Pattern matching	Threshold Min	1	4
	Threshold Max	2	4
	ResultOffsetPos_X	31	5
	ResultOffsetPos_Y	32	5
	ResultOffsetAngle	33	5
Contour comparison	Threshold Min	1	4
	Threshold Max	2	4
	ResultOffsetPos_X	31	5
	ResultOffsetPos_Y	32	5
	ResultOffsetAngle	33	5
Grayscale	Threshold Min	1	4
	Threshold Max	2	4
	GrayMin	101	4
	GrayMax	102	4
	GrayInvert	103	1
Contrast	Threshold Min	1	4
	Threshold Max	2	4
Barcode	Reference String	101	n
Data code	Reference String	101	n
OCR	Reference String	101	n
Color value	ColorMinChannel1	101	4
	ColorMaxChannel1	102	4
	ColorInvertChannel1	103	4
	ColorMinChannel2	104	4
	ColorMaxChannel2	105	4
	ColorInvertChannel2	106	4
	ColorMinChannel3	107	4
	ColorMaxChannel3	108	4
	ColorInvertChannel3	109	4
Color area	ColorMinChannel1	101	4
	ColorMaxChannel1	102	4
	ColorInvertChannel1	103	4

Detector	Function	Command	Length of the following data
	ColorMinChannel2 ColorMaxChannel2 ColorInvertChannel2 ColorMinChannel3 ColorMaxChannel3 ColorInvertChannel3	104 105 106 107 108 109	4 4 4 4 4 4
BLOB	GrayAbsoluteMin GrayAbsoluteMax GrayAbsoluteInvert	101 102 103	4 4 1

Read parameter (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read parameter (binary) request string to sensor			
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	Read reference string / value *1), see below!
Read parameter (binary) response string from sensor (may arrive with a delay of up to 4–5 seconds)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length = 10 bytes + length of reference string (n)
5	Unsigned Char	0x0A	Get Parameter
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
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 config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

*1) Byte No. 7: Command: Read reference string / value

Detector	Function	Command	**Length of the following data**
Alignment Pattern matching	Threshold Min	1	4
	Threshold Max	2	4
	ResultOffsetPos_X	31	5
	ResultOffsetPos_Y	32	5
	ResultOffsetAngle	33	5
Alignment Contour comparison	Threshold Min	1	4
	Threshold Max	2	4
	ResultOffsetPos_X	31	5
	ResultOffsetPos_Y	32	5
	ResultOffsetAngle	33	5
Alignment Edge detection	Transition probe 1 Transition probe 2 Transition probe 3 Score min probe 1 Score min probe 2 Score min probe 3	101 102 103 104 105 106	4 4 4 4 4 4
Pattern matching	Threshold Min	1	4
	Threshold Max	2	4
	ResultOffsetPos_X	31	5
	ResultOffsetPos_Y	32	5
	ResultOffsetAngle	33	5
Contour comparison	Threshold Min	1	4
	Threshold Max	2	4
	ResultOffsetPos_X	31	5
	ResultOffsetPos_Y	32	5
	ResultOffsetAngle	33	5
Grayscale	Threshold Min	1	4
	Threshold Max	2	4
	GrayMin	101	4

Detector	Function	Command	**Length of the following data**
	GrayMax GrayInvert	102 103	4 1
Contrast	Threshold Min Threshold Max	1 2	4 4
Barcode	Reference String	101	n
Data code	Reference String	101	n
OCR	Reference String	101	n
Color value	ColorMinChannel1 ColorMaxChannel1 ColorInvertChannel1 ColorMinChannel2 ColorMaxChannel2 ColorInvertChannel2 ColorMinChannel3 ColorMaxChannel3 ColorInvertChannel3	101 102 103 104 105 106 107 108 109	4 4 4 4 4 4 4 4 4
Color area	ColorMinChannel1 ColorMaxChannel1 ColorInvertChannel1 ColorMinChannel2 ColorMaxChannel2 ColorInvertChannel2 ColorMinChannel3 ColorMaxChannel3 ColorInvertChannel3	101 102 103 104 105 106 107 108 109	4 4 4 4 4 4 4 4 4
BLOB	GrayAbsoluteMin GrayAbsoluteMax GrayAbsoluteInvert	101 102 103	4 4 1

Set ROI (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Set ROI (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x18 0x20	Telegram length 0x18 for circle 0x20 for rectangle / free shape / ellipse
5	Unsigned Char	0x10 0x11	Set ROI temporarily Set ROI permanently Permanent change affect all parameters, including those that were previously only changed temporarily.
6 - 9	Unsigned Int	0xXX	ROI info length in bytes from Byte 6 to end
10	Unsigned Char	0xXX	Detector no.
11	Unsigned Char	0x00	ROI Index 00 = Yellow ROI 01 = Teach 02 = Position control
12	Unsigned Char	0xXX	ROI shape 01= Circle 02= Rectangle 03= Ellipse 04= Free shape
13 - 16	Unsigned Int	0xXX	ROI parameter: Center X (value in pixels * 1000)
17 - 20	Unsigned Int	0xXX	ROI parameter: Center Y (value in pixels * 1000)
21 -	Unsigned	0xXX	ROI parameter: Half width / radius X (value in pixels * 1000)

24	Int		
	Only for ellipse / rectangle:		
25 - 28	Unsigned Int	0xXX	ROI parameter: Half width / radius Y (value in pixels * 1000)
29 - 32	Unsigned Int	0xXX	ROI parameter: Angle in ° (value in ° * 1000)
Set ROI (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1	Unsigned Int	0x07	Telegram length
5	Unsigned Char	0x10 0x11	Set ROI temporarily Set ROI permanently Permanent change affect all parameters, including those that were previously only changed temporarily.
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			Low
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)
Parameter:			The parameters are given in the coordinate system of the alignment and not in the coordinate system of the image.

Read ROI (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read ROI (binary) request string to sensor			
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	ROI Index 00 = Yellow ROI 01 = Teach 02 = Position control
Read ROI (binary) response string from sensor			
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 87)
8 - 11	Unsigned Int	0xXX	ROI info length in bytes from Byte 8
12	Unsigned Char	0xXX	Detector no.
13	Unsigned Char	0x00	ROI Index 00 = Yellow ROI 01 = Teach 02 = Position control
14	Unsigned Char	0xXX	ROI shape 01= Circle

			02= Rectangle 03= Ellipse 04= Free shape
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 * 1000)
	Only for ellipse / rectangle:		
27 - 30	Unsigned Int	0xXX	ROI parameter: Half width / radius Y (value in pixels * 1000)
31 - 34	Unsigned Int	0xXX	ROI parameter: Angle in ° (value in ° * 1000)
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			Low
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

Read job list (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read job list (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x05	Telegram length
5	Unsigned Char	0x14	Get Job List
Read job list (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length
5	Unsigned Char	0x14	Get Job List
6	Unsigned Short	0xXX	Error codes (Page 87)
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	Unsigned	0xXX	Number of subsequent bytes. A description for job n

+ 3	Char		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 + 8	Unsigned Int	0xXX	Date of creation of job n.
k + 9 ... k + 16	Unsigned Int	0xXX	Date of last change of job n.
Additional information:			
Accepted in run mode:		Yes	
Accepted in config mode:		No	
Accepted when Ready is low:		Yes	
Status of Ready signal during processing:		No change	
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)	

Read detector list (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read detector list (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x05	Telegram length
5	Unsigned Char	0x15	Get Detector List
Read detector list (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length
5	Unsigned Char	0x18	Get Detector List
6	Unsigned Short	0xXX	Error codes (Page 87)
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 + 3	Unsigned Char	0xXX	001 - Pattern matching 004 - Contour comparison 005 - Grayscale 006 - Contrast

			007 - Brightness 010 - Wafer 011 - OCR 013 - Data code 014 - Barcode 017 - Busbar 018 - Color value 019 - Color area 020 - Color list 021 - Caliper 022 - BLOB
Additional information:			
Accepted in run mode:		Yes	
Accepted in config mode:		No	
Accepted when Ready is low:		Yes	
Status of Ready signal during processing:		No change	
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)	

Teach detector (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Teach detector (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x08	Telegram length
5	Unsigned Char	0x18	Teach detector
6	Unsigned Char	0xXX	0 = Alignment >= 1 Detectors
7	Unsigned Char	0xXX	0: Temporary 1: Permanent
8	Unsigned Char	0xXX	0 = No trigger 1 = Trigger
Teach detector (binary) response string from sensor			
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 87)
Additional information:			
Accepted in run mode:		Yes	
Accepted in config mode:		No	
Accepted when Ready is low:		Yes	
Status of Ready signal during processing:		No change	
Supported interfaces:		Telegrams: Availability and supported interfaces (Page 84)	

Set trigger delay (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Set trigger delay (binary) request string to sensor			
Byte no..	Data type	Content	Meaning
1 - 4	Unsigned Int	0xB	Telegram length
5	Unsigned Char	0x27	Set trigger delay
6	Unsigned Char	0xXX	Request version
7	Unsigned Char	0xXX	0: Temporary 1: Permanent
8 - 11	Unsigned Int	0xXX	Trigger delay in msec (max. 3000 msec) in encoder steps (max. 65535 steps)
Set trigger delay (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x07	Telegram length
5	Unsigned Char	0x1A	Set trigger delay
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			Low
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

Get trigger delay (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Read trigger delay (binary) request string to sensor			
Byte no..	Data type	Content	Meaning
1 - 4	Unsigned Int	0x6	Telegram length
5	Unsigned Char	0x28	Read trigger delay
6	Unsigned Char	0xXX	Request version
Read trigger delay (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xB	Telegram length
5	Unsigned Char	0x28	Read trigger delay
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
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 config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

Save job permanently (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Save job permanently (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x5	Telegram length
5	Unsigned Char	0xD	Save job permanently
Save job permanently (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x0B	Telegram length
5	Unsigned Char	0x31	Save job permanently
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			Low
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

Calibration: Initialize point list (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Initialize point list (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x05	Telegram length
5	Unsigned Char	0x1F	Calibration: Clear Data
Initialize point list (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x07	Telegram length
5	Unsigned Char	0x1F	Calibration: Clear Data
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

Calibration: Add world point (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Calibration: Add world point (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x0F	Telegram length
5	Unsigned Char	0x26	Calibration: Add World Point (CAW)
6	Unsigned Char	0xXX	Request version
7	Unsigned Char	0xXX	1: Fiducials only Calibration plate (Robotics) 4: World point and image point Point pair list (Robotics)
9 - 10	Unsigned Short	0xXX	Constant (5 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	0xXX	Constant (8 bytes)
Calibration: Add world point (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x11	Telegram length
5	Unsigned Char	0x26	Calibration: Add World Point (CAW)
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
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 config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

Calibration: Point list (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Calibration: Point list (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x06	Telegram length
5	Unsigned Char	0x1E	Calibration: Calibrate by Point List
6	Unsigned Char	0xXX	0: Temporary 1: Permanent
Calibration: Point list (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x19	Telegram length
5	Unsigned Char	0x1E	Calibration: Calibrate by Point List
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
8 - 9	Unsigned Short	0xXX	Current highest point pair index
10 - 13	Unsigned Int	0xXX	Deviation calibration, RMSE (root mean square error)
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 config mode:			No
Accepted when Ready is low:			Yes

Status of Ready signal during processing:	No change
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)

Calibration: Validate point list (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Calibration: Validate point list (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x05	Telegram length
5	Unsigned Char	0x20	Calibration: Validate by Point List
Calibration: Validate point list (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x19	Telegram length
5	Unsigned Char	0x20	Calibration: Validate by Point List
6	Unsigned Short	0xXX	Error codes (Page 87)
8 - 9	Unsigned Short	0xXX	Current highest point pair index
10 - 13	Unsigned Int	0xXX	Deviation calibration, RMSE (root mean square error)
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 config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			Telegrams: Availability and

	supported interfaces (Page 84)

Calibration: Calibration plate (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Calibration: Calibration plate (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x09	Telegram length
5	Unsigned Char	0x24	Calibration: Calibrate by Plate
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	0: Temporary 1: Permanent
8	Unsigned Char	0xXX	0 - No fiducials are used. The origin of the measurement coordinate system is identical to the origin of the calibration coordinate system. 1 - No fiducials are used. Measurement coordinate system is identical to the device coordinate system. 2 - Uses world system, fiducial job 3 - Uses world system, fiducials command CAW
9	Unsigned Char	0xXX	0: Calibration internal and external sensor parameters 1: Validation of calibration 2: Calibration internal sensor parameters 5: Calibration transformation measurement coordinate system
Calibration: Calibration plate (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x3D	Telegram length
5	Unsigned	0x24	Calibration: Calibrate by Plate

	Char		
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
8 - 9	Unsigned Short	0xXX	Current number of detected calibration points
10 - 13	Unsigned Int	0xXX	Deviation calibration, RMSE (root mean square error)
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	X delta (in user unit * 1000)
30 - 33	Unsigned Int	0xXX	Y delta (in user unit * 1000)
34 - 37	Unsigned Int	0	Reserved
38 - 41	Unsigned Int	0	Reserved
42 - 45	Unsigned Int	0	Reserved
46 - 49	Unsigned Int	0xXX	Delta Gamma (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 config mode:	No
Accepted when Ready is low:	Yes
Status of Ready signal during processing:	No change
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)

Calibration: Set fiducial (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Calibration: Set fiducial (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x08	Telegram length
5	Unsigned Char	0x1E	Calibration: Set Fiducial
6	Unsigned Char	0xXX	Request version
7	Unsigned Char	0xXX	0: Temporary 1: Permanent
Calibration: Set fiducial (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x37	Telegram length
5	Unsigned Char	0x2B	Calibration: Set Fiducial
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
8 - 11	Unsigned Int	0xXX	X value
12 - 15	Unsigned Int	0xXX	Y value
16 - 19	Unsigned Int	0xXX	Z value
18 - 21	Unsigned Int	0xXX	Red X value
22 - 25	Unsigned Int	0xXX	Red Y value
26 -29	Unsigned Int	0xXX	Red Z value
30 - 33	Unsigned Int	0xXX	Deviation fiducials, mean
34 - 37	Unsigned Int	0xXX	Deviation fiducials, max.

38 - 41	Unsigned Int	0xXX	Deviation fiducials, min.
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

Calibration: Copy calibration (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Calibration: Copy calibration (binary) request string to sensor			
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 (binary) response string from sensor			
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 87)
8	Unsigned Char	0xXX	00: Successful >0 : Job number at which the error occurs.
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			No

Accepted when Ready is low:	Yes
Status of Ready signal during processing:	No change
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)

Calibration: Set parameter (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Calibration: Set parameter (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x07	Telegram length
5	Unsigned Char	0x29	Calibration: Set Parameter
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	0: Temporary 1: Permanent
8	Unsigned Char	0xXX	Parameter number (Page 242)
9 - 12	Unsigned Int	0xXX	Length of the following data
13 ... n	Unsigned Char	0xXX	Parameter value (Page 242)
Calibration: Set parameter (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length
5	Unsigned Char	0x29	Calibration: Set Parameter
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			No
Accepted when Ready is low:			Yes

Status of Ready signal during processing:	No change
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)

Parameter name	Parameter number	Parameter value	Length	Format	Status
Unit (user unit)	004	0: mm 1: cm 2: m 3: inch 4: au	1 Byte	Characters	no change
Z Offset	021	(in user unit * 1000)	8 Byte	Signed integer	no change
Focal length	022	(in mm *1000)	8 Byte	Signed integer	invalid
Fiducial 1	024	in each case: 0: X value (in user unit * 1000) 1: Y value (in user unit * 1000) 2: Z value (in user unit * 1000)	8 bytes per value	Signed integer	no change
Fiducial 2	025				
Fiducial 3	026				
Fiducial 4	027				

Calibration: Read parameter (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Calibration: Read parameter (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x07	Telegram length
5	Unsigned Char	0x2A	Calibration: Get Parameter
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	Parameter number (Page 244)
Calibration: Read parameter (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length
5	Unsigned Char	0x2A	Calibration: Get Parameter
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
8	Unsigned Char	0xXX	Parameter number (Page 244)
9 - 12	Unsigned Int	0xXX	Length of the following data
13 ... n	Unsigned Char	0xXX	Parameter value (Page 244)
Additional information:			
Accepted in run mode:			Yes
Accepted in config mode:			No
Accepted when Ready is low:			Yes

Status of Ready signal during processing:	No change
Supported interfaces:	Telegrams: Availability and supported interfaces (Page 84)

Parameter name	Parameter number	Parameter value	Length	Format
Status calibration	001	0: Invalid 1: Valid	1 Byte	Signed integer
Calibration method	002	0: None 1: Scaling (Measurement) 2: Point pair list (Robotics) 3: Calibration plate (Measurement) 4: Calibration plate (Robotics)	1 Byte	Characters
Unit (user unit)	004	0: mm 1: cm 2: m 3: inch 4: au	1 Byte	Characters
Internal sensor parameters	005	0: Focal length (in mm * 1000) 1: Kappa value (*1000) 2: X pixel pitch (in µm * 1000) 3: Y pixel pitch (in µm * 1000) 4: X coordinate origin (in pixels * 1000) 5: Y coordinate origin (in pixels * 1000) 6: Image size X (pixels) 7: Image size Y (pixels)	8 bytes per value	Signed integer

Parameter name	Parameter number	Parameter value	Length	Format
External parameters	011	0: X axis offset (in user unit * 1000) 1: Y axis offset (in user unit * 1000) 2: Z axis offset (in user unit * 1000) 3: X rotation (in degrees * 1000) 4: Y rotation (in degrees * 1000) 5: X rotation (in degrees * 1000)	8 bytes per value	Signed integer
Transformation calibration coordinate system	012	0: X axis offset (in user unit * 1000) 1: Y axis offset (in user unit * 1000) 2: Z axis offset (in user unit * 1000) 3: X rotation (in degrees * 1000) 4: Y rotation (in degrees * 1000) 5: X rotation (in degrees * 1000)	8 bytes per value	Signed integer
Transformation measurement coordinate system	014	0: X axis offset (in user unit * 1000) 1: Y axis offset (in user unit * 1000) 2: Z axis offset (in user unit * 1000) 3: X rotation (in degrees * 1000) 4: Y rotation (in degrees * 1000) 5: X rotation (in degrees * 1000)	8 bytes per value	Signed integer
Z Offset	021	(in user unit * 1000)	8 Byte	Signed integer
Focal length	022	(in mm *1000)	8 Byte	Signed

Parameter name	Parameter number	Parameter value	Length	Format
				integer
Fiducial 1	024	in each case: 0: X value (user unit * 1000) 1: Y value (user unit * 1000) 2: Constant 0	8 bytes per value	Signed integer
Fiducial 2	025			
Fiducial 3	026			
Fiducial 4	027			

Get image (BINARY)

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Get image (binary) request string to sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x06	Telegram length
5	Unsigned Char	0x03	Get image
6	Unsigned Char	0xXX	0 - Last image 1 – Last failed image 2 – Last good image
Get image (binary) response string from sensor			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length e.g. 00 04 B0 0D (Dez. 307213)
5	Unsigned Char	0x03	Get image
6 - 7	Unsigned Short	0xXX	Error codes (Page 87)
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	0xXX	Number of rows

	Short		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 config mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			Low
Supported interfaces:			Telegrams: Availability and supported interfaces (Page 84)

Data output in BINARY

Telegrams: Availability and supported interfaces (Page 84)

Overview telegrams: (Page 80)

Output data (BINARY), dynamically put together based on the user settings in the software: SensoConfig/Output/Data output.

Basic string construction:

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

Output data (BINARY):

<OPTIONAL FIELDS>				
Parameter	**Description**	**Length Binary [Byte]**	**Data type**	**Available for**
Selected fields	This checkbox shows all selected fields. The checkbox for selected fields is not selected fields.	2	ASCII: The output order is from left to right and from top to bottom, i.e., for each checkbox that is enabled, a byte is set beginning with the LSB. Binary: The output order is from left to right and from top to bottom, i.e., for each checkbox that is enabled, a bit is set beginning with the least significant one.	All types
Telegram length	Number of characters, incl.	2	ASCII: E.g., output string with 10 characters; telegram length: 10 + 2 characters (one byte per whole number	All types

<OPTIONAL FIELDS>				
Parameter	**Description**	**Length Binary [Byte]**	**Data type**	**Available for**
	the characters for the telegram length itself.		place) = 12	
Status byte	Represents the trigger mode.	2	ASCII: PPF = Trigger; PFP = Free run Binary: 0X06 0x00 = Trigger mode 0X05 0x00 = Free run mode	All types
Detector results	Output of overall result for each detector.	3 ... 35	ASCII: Byte 1 = AND conjunction of all detectors Byte 2 = Overall position tracking result Byte 3 = Overall current job result Followed by the number of detectors; one byte per whole number 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.	n	ASCII: 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 Binary [Byte]**	**Data type**	**Available for**
			Binary: Bytes 1 and 2: Number of active Outputs Bytes 3 – n: Outputs, bit-coded	
Log. Outputs	Returns the logic gate result for each logic output.	n	ASCII: Starting from byte 1 Number of active outputs (logic gate result assigned); 1 byte per whole number place Following bytes: One byte per logic output P = Detector pass F = Detector fail 0 = Inactive output (gap between two active outputs) Binary: Byte 1 and byte 2: Number of active log. Outputs Byte 3 – n All active logic outputs,	All types
Execution time	Returns the	4	Signed integer	All types

<OPTIONAL FIELDS>				
Parameter	**Description**	**Length Binary [Byte]**	**Data type**	**Available for**
	execution time for the last evaluation.			
Active job	Returns the job for the last evaluation.	1	Unsigned Int u8	All types

<PAYLOAD>

Overview of detector-specific payload values

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 checks with "OK" result	4	Signed integer	GENERAL
Fail parts counter	Number of inspections with result "Error"	4	Signed integer	GENERAL

Value	Description	Length Binary [Byte]	Data type	Available for
Timeout	Indicates that the maximum cycle time has been exceeded.	1	Boolean	GENERAL
Recording	Indicates the number of image acquisition repetitions for the last evaluation Only in combination with repeat mode.	4	Int	GENERAL
String	This field can be used to enter a constant string into the data output.	0 ... 5	String	GENERAL

Base values

Value	Description	Length Binary [Byte]	Data type	Available for
Score	[%]	4	Signed integer	All detectors
Overall result	Boolean detector result	1	Boolean	All detectors
Execution time	Execution time of individual detector in [msec].	4	Signed integer	All detectors

Position

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	
Delta pos. X	X position delta between the taught object and the found object, 1/1000 [user unit]	4	Signed integer	
Delta pos. Y	Y position delta between the taught object and the found object, 1/1000 [user unit]	4	Signed integer	
Angle	Orientation of found object, 1/1000 [°]	4	Signed integer	
Angle (45)	Orientation of bounding box for found code [°], value range: -45° to 45°	4	Signed integer	

Value	Description	Length Binary [Byte]	Data type	Available for
Angle (180)	Orientation of object width (long axis) [°], value range: -90° to 90° 0° = East, counterclockwise	4	Signed integer	
Angle (360)	Orientation of object width (long axis) [°], value range: -180° to 180° 0° = East, counterclockwise	4	Signed integer	
Delta angle	Angle between taught object and found object, 1/1000 [°]	4	Signed integer	
Position check		1	Boolean	

Measurement

Value	Description	Length Binary [Byte]	Data type	Available for
Height	Height of geometric element [user unit], Height ≥ 0, height ≤ width	4	Signed integer	
Width	Width of geometric element [user unit] Width ≥ 0, width ≥ 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 with holes, 1/1000 [pixels]	4	Signed integer	
Distance	Calculated distance [user unit]	4	Signed integer	

Identification

Value	Description	Length Binary [Byte]	Data type	Available for
String length	Length of read code [bytes]	4	Signed integer	
String	Contents of read code. Depending on the code, the string length may vary. If a fixed string length is desired, the minimum string length (detector-specific user data) and the maximum string length (detector settings) must be set to the same value (e.g. 127).	n	String	
String comparison	Content check for the read information. The contents of the read information are checked using regular expressions (Reiter Referenzstring)	1	Boolean	
Truncated	Code complete or truncated F = Code complete P = Code truncated	1	Boolean	

Identification - quality

Value	Description	Length Binary [Byte]	Data type	Available for
Quality - overall	Evaluation of printing quality according to international standard ISO/IEC 15416. The overall quality indicates the final rating. [x].			
Quality - individual	Evaluation of printing quality according to international standard ISO/IEC 15416. Printing quality rating.			
Min. Quality	Used to check whether the minimum required quality is being met			

Color

Value	Description	Length Binary [Byte]	Data type	Available for
Color value: • Red,	Value for color parameter	4	Signed integer	

Value	Description	Length Binary [Byte]	Data type	Available for
green, blue • Hue, saturation, lightness • Luminance, a, b				
Color distance	Distance of the current color versus the taught-in color	4	Signed integer	

Counting / number

Value	Description	Length Binary [Byte]	Data type	Available for
Number of objects	Maximum number of objects that must be detected [units]	4	Signed integer	
Number of valid objects	Used to check whether the number of objects found falls within a specific range.	4	Signed integer	

Value	Description	Length Binary [Byte]	Data type	Available for
	[Good/Bad or units]			
Number of search stripes	Number of parallel search stripes into which the width of the search range is divided. [units]	4	Signed integer	(only Edge detection)
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	(only Edge detection)
Too many BLOBs		1	Boolean	

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.	4		(only Contour comparison)

Value	Description	Length Binary [Byte]	Data type	Available for
	Value range of 0.5 to 2			
Eccentricity	Numerical eccentricity Value range of 0.0 to 1.0	4	Signed integer	
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		
Reference string met	The output string matches the reference string.	1	Boolean	
Contrast	Code contrast Value range of 0 to 100 [%]	4		
Correction	Number of modules corrected by error corrections [units]	4		
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	4	Signed integer	

Value	Description	Length Binary [Byte]	Data type	Available for
	deviates from a circle, the greater the compactness value will be.			
Center of gravity X	X coordinate of centroid, 1/1000	4	Signed integer	
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. signal threshold				
Max. signal threshold				
Inverted signal threshold				
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		
Deviation, outside	Returns the largest deviation between the BLOB contour and the contour of the geometric element (deviation outside the fitted circle).	4		

Value	Description	Length Binary [Byte]	Data type	Available for
	[user unit]			
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		
Axial ratio	Ratio of the long to the short axis (a / b)	4		
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				

<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**
Trailer	Characters appended at the end of the string	0 ... 8		All types

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.
For example, via Ethernet, the following four bytes are received: 000,000,139,115
Formula for recalculating: (HiWordByte*256 + HiLowByte) *65536 + HiByte*256 + LoByte = 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 (= real score value)
Angle data or other negative values are represented in two's complement.

Wir schauen nach vorn

Gestern, heute und in Zukunft

SENSOPART

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

Frankreich
SensoPart France SARL
77420 Champs sur Marne
Tel. +33 164 730061
info@sensopart.fr

Großbritannien
SensoPart UK Limited
Pera Business Park, Nottingham Road,
Melton Mowbray, Leicestershire, LE13 0PB
Tel. +44 1664 561539
uk@sensopart.com

USA
SensoPart Inc.
Perrysburg OH 43551
Tel. +1 866 2827610
usa@sensopart.com

China
SensoPart (Shanghai) Co. Ltd.
201803 Shanghai
Tel. +86 21 69017660
china@sensopart.cn