



VISOR<sup>®</sup>

Communications manual

Software version 2.12

## Copyright

---

Any reproduction or reprinting of this document, as well as any corresponding storage in databases or retrieval systems, or the publication, in any form whatsoever, even in part, or any imitation of the illustrations, drawings and design is permitted only on the basis of prior written authorization from SensoPart Industriesensorik GmbH.

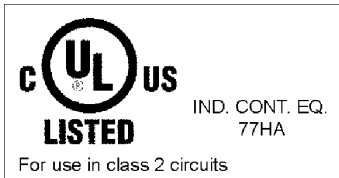
No liability whatsoever shall be accepted for printing errors or mistakes made in the preparation of this document. Subject to delivery options and technical changes.

- Translation of Original operating instructions -

Initial publication 01 / 2019

First publication 06/2025

SensoPart Industriesensorik GmbH  
Nägelsestr. 16  
79288 Gottenheim  
Germany



For use in NFPA 79 Applications only



## Table of contents

---

<b>1 Information on this document</b>	<b>5</b>
1.1 What the symbols mean	5
1.2 Additional documents	6
1.3 Document version	6
<b>2 Network settings</b>	<b>7</b>
2.1 Integrating the VISOR® into the network / gateway	7
2.2 Direct connection of the VISOR® vision sensor	7
2.3 Connection of the VISOR® vision sensor via network	7
2.4 Used Ethernet ports	8
2.5 Access to VISOR® through network	9
2.6 Access to VISOR® through the Internet / World Wide Web	10
<b>3 Timing diagrams for VISOR® communication</b>	<b>11</b>
<b>4 Ethernet TCP/IP, port 2005 / 2006</b>	<b>15</b>
4.1 Example: Communication via one port, unidirectional	16
4.2 Example: Communication via two ports, bidirectional	20
4.3 Example: Communication via one port, bidirectional	22
4.4 More examples	22
4.4.1 Example: Job change from PC / controller to VISOR® (CJB)	23
4.4.2 Example: Set shutter speed (SST/SSP) / read (GSH)	25
<b>5 Service / Visualization</b>	<b>28</b>
5.1 Backup creation	28
5.2 Visualization	28
<b>6 VISOR® telegrams for PROFINET and EtherNet/IP</b>	<b>29</b>
6.1 Module 1: "Control" (From PLC to VISOR®)	29
6.2 Module 2: "Status" (from VISOR® to PLC)	31
6.3 Module 3: "Data" (from VISOR® to PLC)	33
6.4 Module 4: "Request" (From PLC to VISOR®)	34
6.5 Module 5: "Response" (From VISOR® to PLC)	35
6.6 Start / end criteria for each telegram	36
<b>7 Telegrams</b>	<b>37</b>
7.1 Overview	37
7.2 Availability and supported interfaces	41
7.3 Sequences	45
7.3.1 Trigger Request Sequence	46
7.3.2 Change job request sequence	47
7.3.3 Switch to Run sequence	47
7.3.4 Sequence for requests via request/response module	48
7.3.5 Calibration via telegrams	48
7.3.5.1 Automated calibration: Point pair list (Robotics)	49
7.3.5.2 Automated calibration: Calibration plate (Robotics)	50
7.3.5.3 Automated calibration: Hand-Eye calibration (Robotics)	53
7.3.5.4 Automated calibration: Base-Eye calibration (Robotics)	54
7.3.6 Calibration routine VISOR® Robotic +Z	54
7.4 Description Telegrams ASCII	57
7.4.1 General	57
7.4.2 Control	60
7.4.3 Job settings	70

7.4.4 Calibration .....	112
7.4.5 Visualization .....	132
7.4.6 Service (available only on port 1998 and in ASCII format) .....	134
7.5 Description Telegrams BINARY .....	141
7.5.1 General .....	141
7.5.2 Control .....	143
7.5.3 Job settings .....	152
7.5.4 Calibration .....	186
7.5.5 Visualization .....	205
7.6 Error codes .....	205
<b>8 Telegram</b> .....	<b>208</b>
8.1 Description of data output ASCII .....	208
8.2 Description of data output BINARY .....	224

# 1 Information on this document

## 1.1 What the symbols mean

### Warnings



**CAUTION or WARNING or 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

	Data code		Contour
	Barcode		Pattern matching
	OCR		Contour 3D
	Brightness		BLOB
	Contrast		Target Mark 3D
	Contour check		Shape find
	Grayscale		Caliper
	Color value		Caliper, circle
	Color area		Result processing
	Classification (AI)		
	Color list		

### Alignment

	Pattern matching
	Edge detection
	Contour comparison
	Circle

## **1.2 Additional documents**

---

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

- VISOR<sup>®</sup> Operating manual
- VISOR<sup>®</sup> User Manual
- VISOR<sup>®</sup> Communications manual
- VISOR<sup>®</sup> Web user manual

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

## **1.3 Document version**

---

This manual describes the VISOR<sup>®</sup> Vision software version 2.12.

## 2 Network settings

---

### 2.1 Integrating the VISOR<sup>®</sup> into the network / gateway

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

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

### 2.2 Direct connection of the VISOR<sup>®</sup> vision sensor



**NOTE:**

To configure the PC via the direct connection, the IP configuration of the PC must be changed. In advance, inform the administrator about changes.

1. Connect the sensor: power supply 24 V; network cable.  
→ Green PWR LED of VISOR<sup>®</sup> is on. Status LED on network adapter is active and/or LAN status shows "Connection established" (Control Panel / Network Connection / LAN Connection / Status)
2. Check whether the IP address of the PC matches the vision sensor. Sensor (Default): 192.168.100.100/24 (24 means: 24-bit subnet mask = 255.255.255.0).  
If not, check / correct the settings.
3. Open SensoFind and click the "Find" button. Select sensor.
4. Click the "Configuring" button or double-click on Vision Sensor.  
→ SensoConfig opens.
5. Parameterize VISOR<sup>®</sup> in SensoConfig.

### 2.3 Connection of the VISOR<sup>®</sup> vision sensor via network

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



**NOTE:**

To configure the VISOR<sup>®</sup> for the network, it must be integrated into the network. To establish communication between the vision sensor and the PC, the sensor requires a free IP address. Before connecting, check whether the address of the sensor is already assigned to other devices! Default: 192.168.100.100/24 (24 means: 24-bit subnet mask = 255.255.255.0)  
Contact your administrator to obtain a free IP address.

1. Connect the sensor: power supply 24 V; network cable.  
→ Green PWR LED of VISOR<sup>®</sup> is on. Status LED on network adapter is active and/or LAN status shows "Connection established" (Control Panel / Network Connection / LAN Connection / Status)
2. Open SensoFind and click the "Find" button. Select sensor.
3. Configure the IP address via the "Settings" button and enter the subnet mask. Confirm with "Set".
4. Click the "Find" button.
5. Click the "Configuring" button or double-click on Vision Sensor.  
→ SensoConfig opens.
6. Parameterize VISOR<sup>®</sup> in SensoConfig.

## 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® Vision software (PC) and the VISOR®:

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

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

Process interfaces:

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



**NOTE:**

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

## 2.5 Access to VISOR® through network

Exemplary values for IP, etc.

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

- Via SensoFind (/find)

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

Only if:

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

**PC 1**  
 IP: 192.168.20.x  
 Subnetmask: 255.255.255.0  
 Gateway: 192.168.20.1

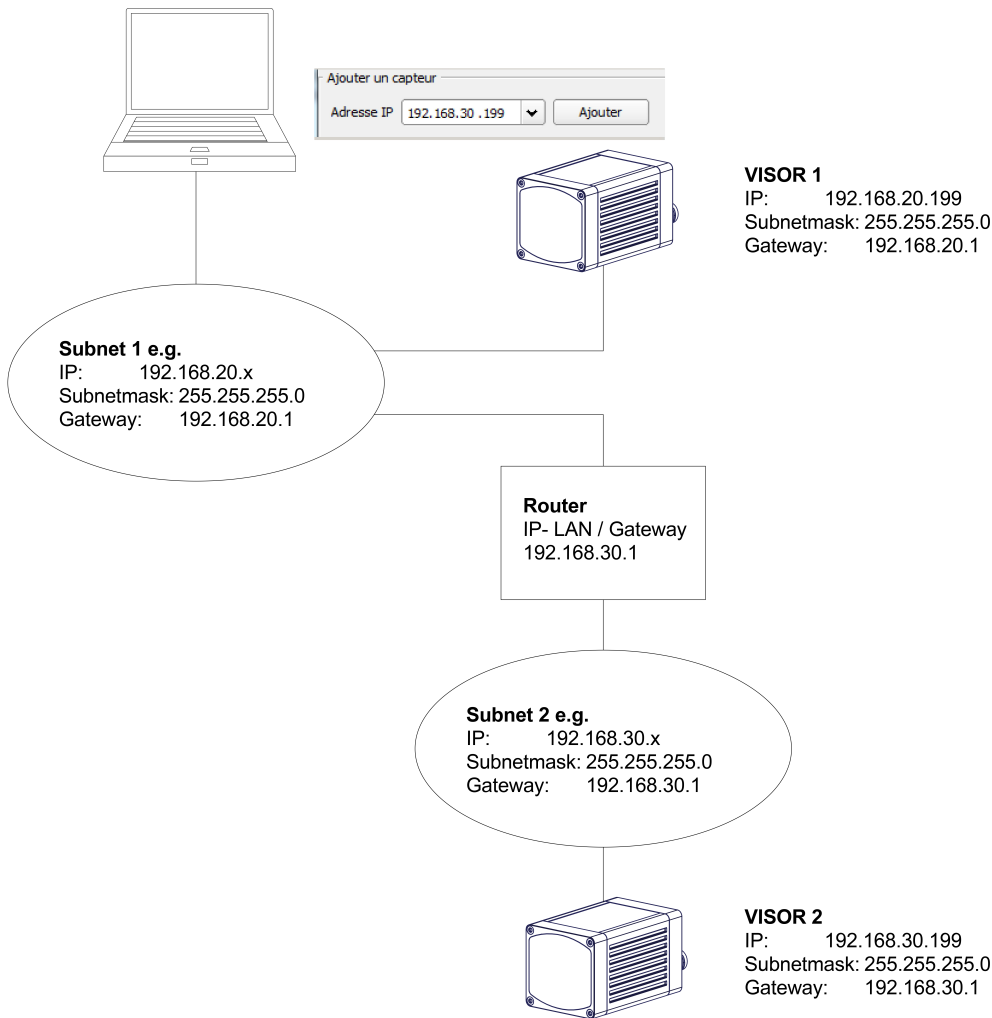


Fig. 1: Access to VISOR® through network, same or other subnet

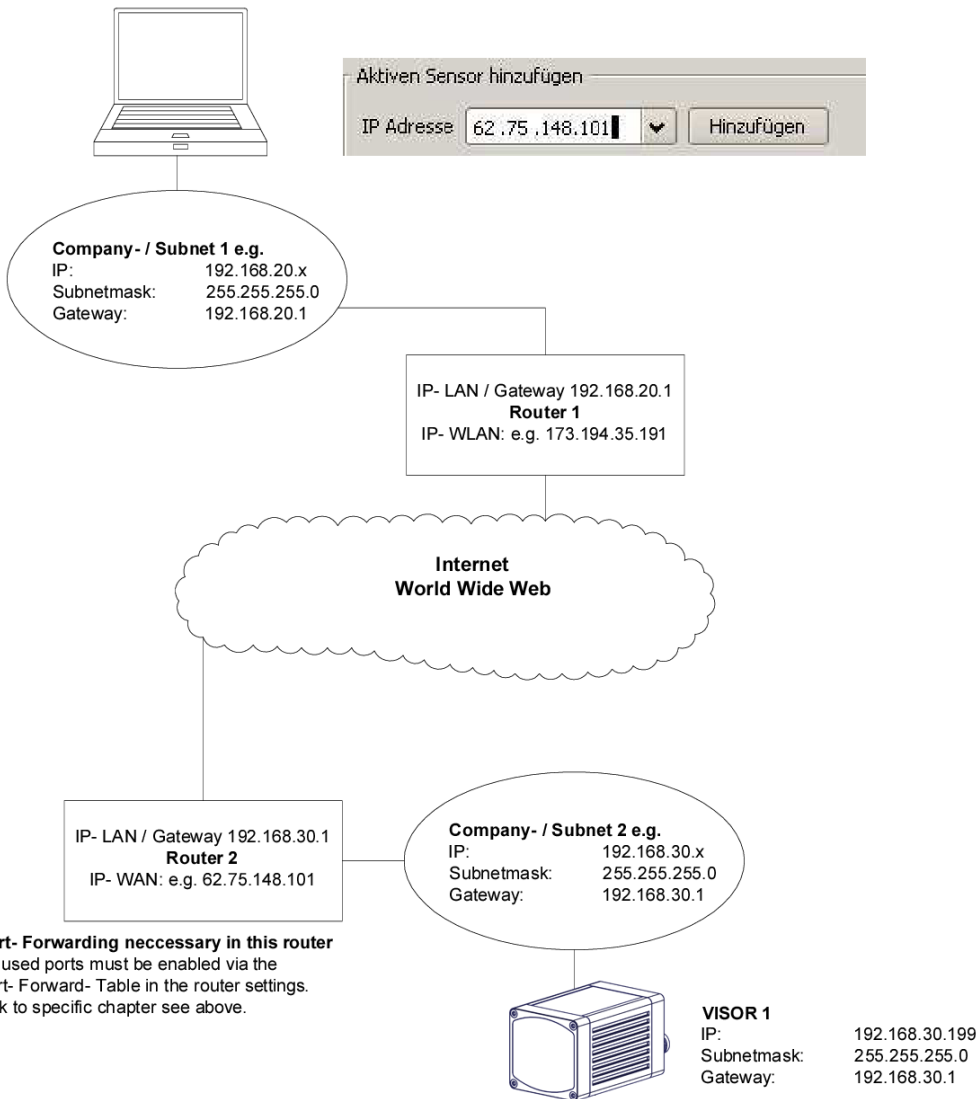
## 2.6 Access to VISOR® through the Internet / World Wide Web

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. In Router 2, enable the ports that are to be used by the sensor in the router (See Chapter: [Used Ethernet ports](#)).

**PC 1**  
 IP: 192.168.20.x  
 Subnetmask: 255.255.255.0  
 Gateway: 192.168.20.1



**Port- Forwarding necessary in this router**  
 All used ports must be enabled via the Port- Forward- Table in the router settings.  
 Link to specific chapter see above.

**Fig. 2: Access to VISOR® through the Internet / World Wide Web**

### 3 Timing diagrams for VISOR® communication

#### Case: Trigger ok

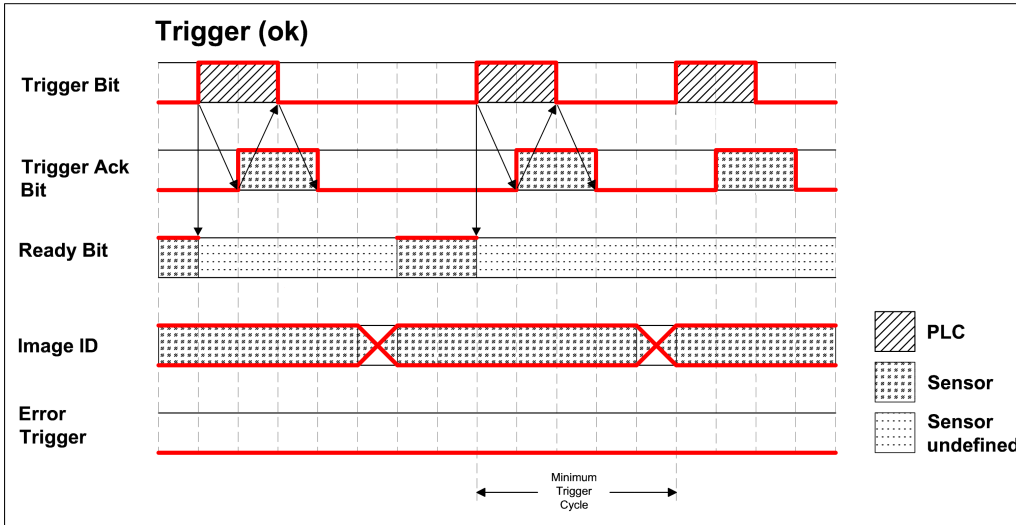


Fig. 3: Timing Trigger ok

#### Case: Trigger not possible (not ready)

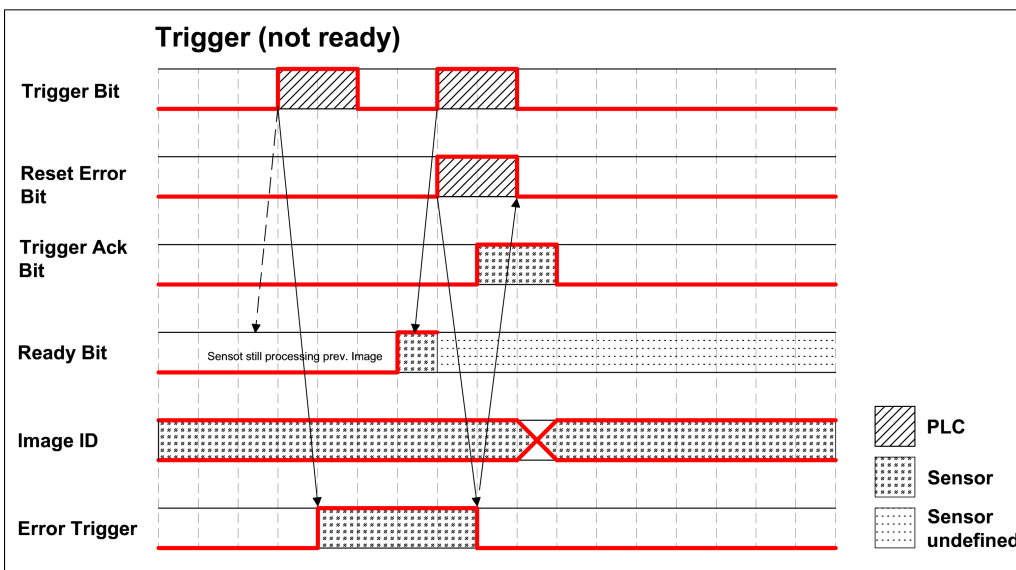
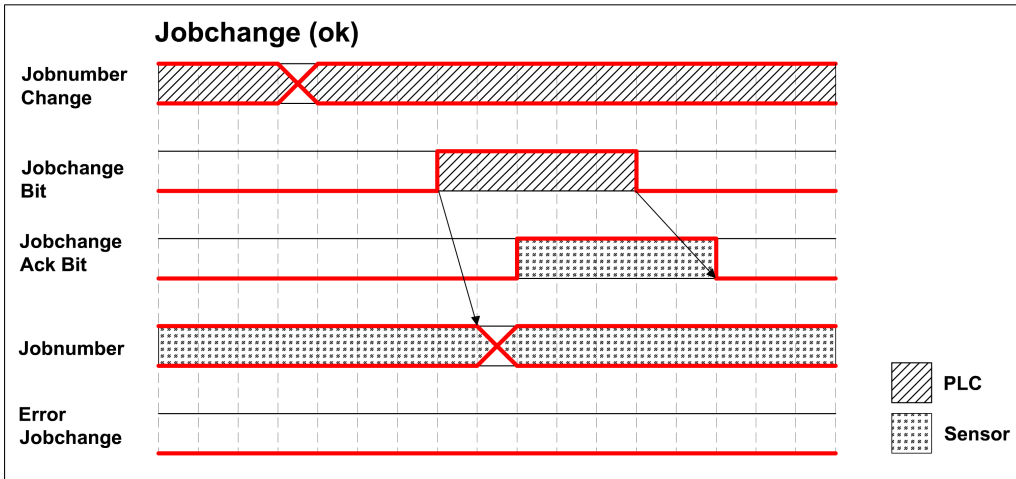


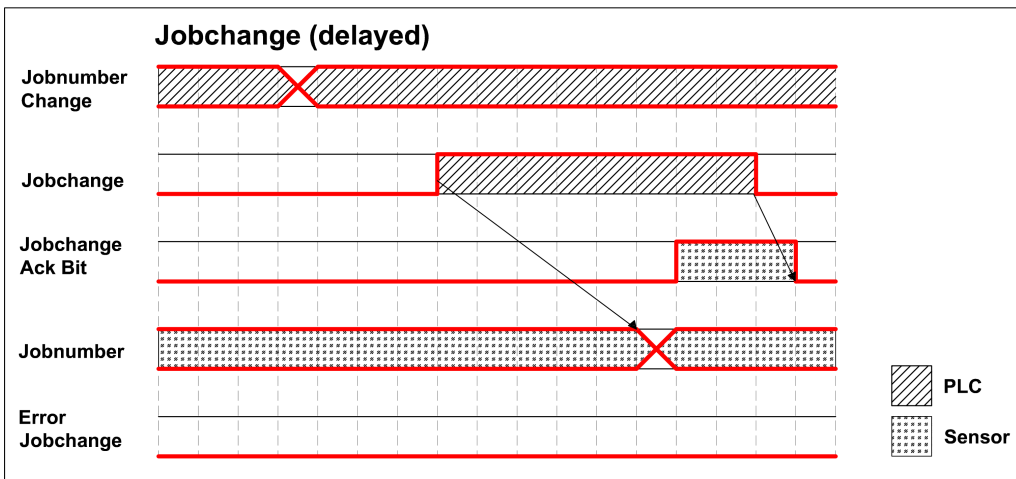
Fig. 4: Timing Trigger not ready

**Case: Job change ok**



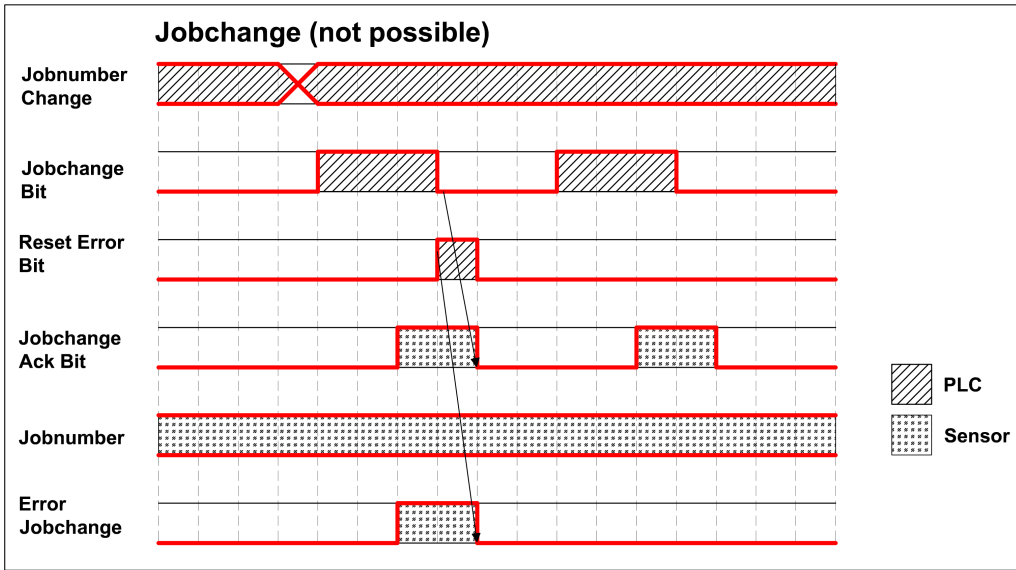
*Fig. 5: Timing Job change ok*

**Case: Job change delayed**



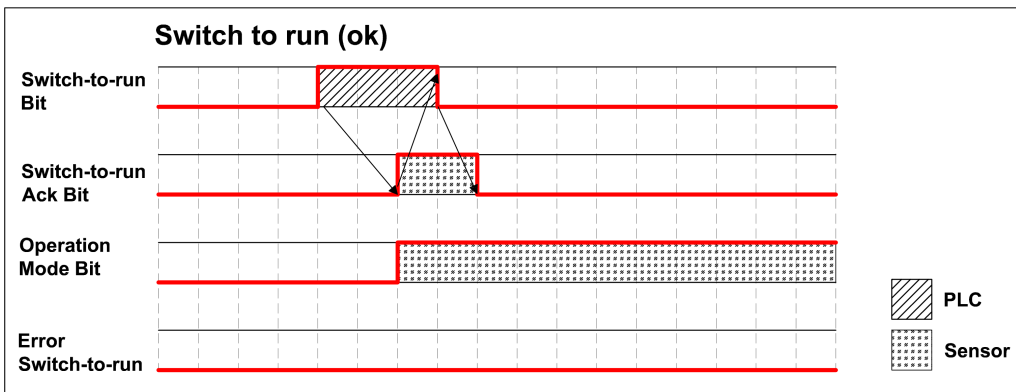
*Fig. 6: Timing Job change delayed*

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



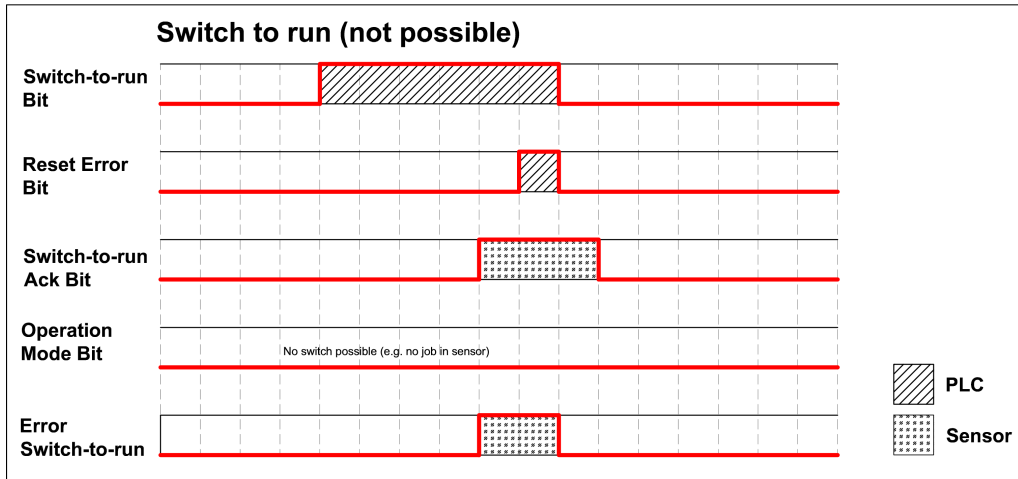
*Fig. 7: Timing Job change not possible*

**Case: Switch to run ok**



*Fig. 8: Timing Switch to run ok*

**Case: Switch to run not possible**



*Fig. 9: Switch to run not possible*

## 4 Ethernet TCP/IP, port 2005 / 2006

4.1 Example: Communication via one port, unidirectional .....	16
4.2 Example: Communication via two ports, bidirectional .....	20
4.3 Example: Communication via one port, bidirectional .....	22
4.4 More examples .....	22

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.

Data configured at Output/Telegram can be output in a separate ASCII or BINARY format.

There are three variants of how the VISOR® vision sensor can communicate with a socket client (PC, PLC, etc.).

### 1. Communication via one port, unidirectional

From the PLC, only a connection to the VISOR® vision sensor needs to be established to receive the result data. No telegrams can be sent to the vision sensor!  
 The vision sensor receives all commands (for example triggers) via the hardware inputs (e.g. from a light barrier)  
 Example: Default Port 2006  
 VISOR® is triggered by an external sensor, VISOR®: image acquisition, sends result data to PLC via port 2006.

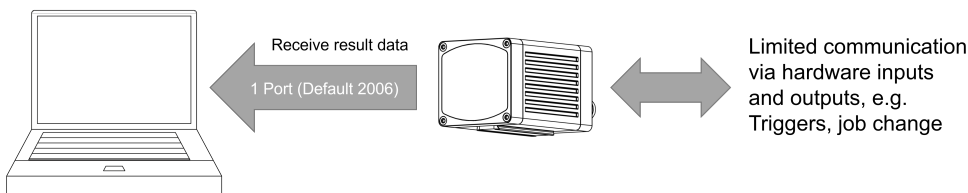


Fig. 10: Communication via one port, unidirectional

### 2. Communication via two ports, bidirectional

Two connections must be established from the PLC to the VISOR® vision sensor. All available telegrams (except service port telegrams) can be sent to the vision sensor. The telegram response also comes on this port. The result data is sent to PC, PLC etc. via an extra port.  
 Example: Default Ports 2006 and 2005  
 PLC sends trigger via port 2006; VISOR®: image acquisition, sends telegram response to PLC via port 2006, result data to PLC via port 2005.

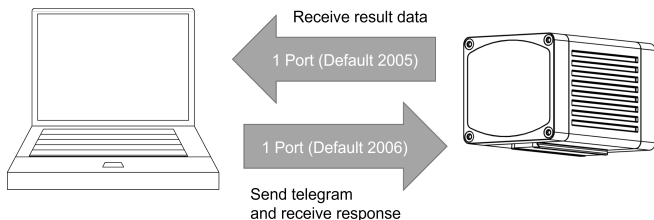


Fig. 11: Communication via two ports, bidirectional

### 3. Communication via one port, bidirectional

From the PLC, only a connection to the VISOR® vision sensor needs to be established. All available telegrams (except service port telegrams) can be sent to the vision sensor. The telegram response also comes on this port. To obtain result data, the Extended Trigger (TRX) can be used. The result data is appended to the telegram response.  
 Example: Default Port 2006

PLC sends trigger via port 2006; VISOR®: image acquisition, sends telegram response and result data to PLC via the same port.

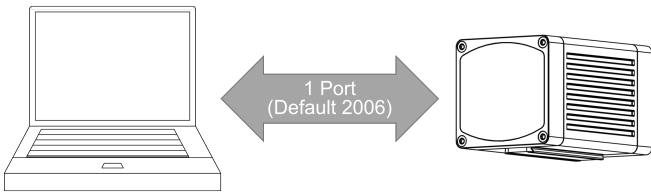


Fig. 12: Communication via one port, bidirectional

## 4.1 Example: Communication via one port, unidirectional

### Data output from VISOR® to the PC / controller via OUT port

#### Step 1

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

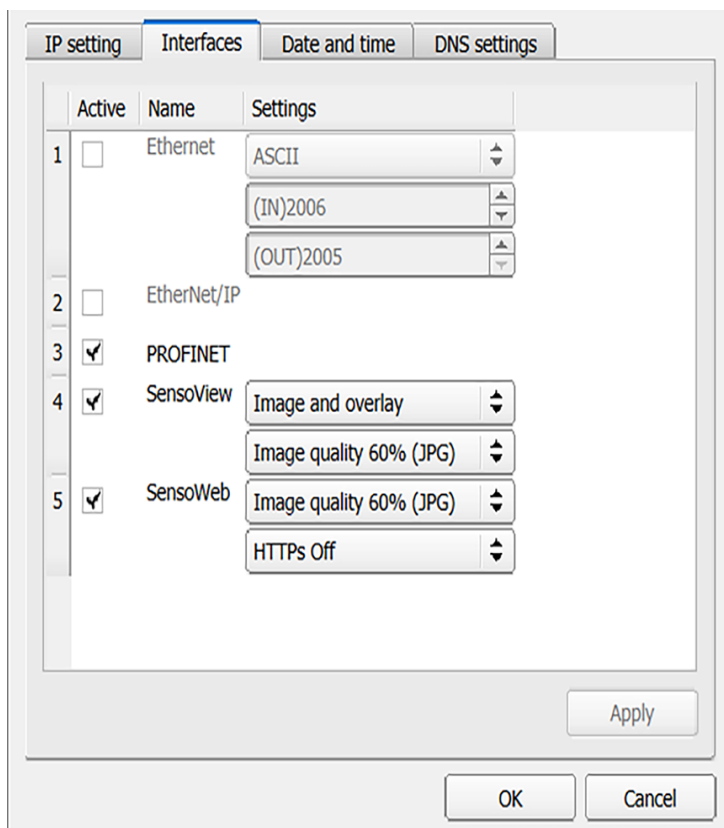


Fig. 13: Interfaces, Ethernet

In the example, the Ethernet interface is enabled in the tab: "Interfaces" by marking the checkbox. 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. The default ASCII output format is also retained.

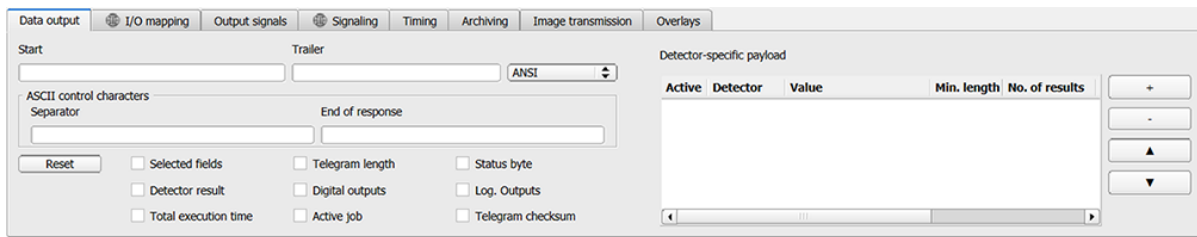
**Step 2**

In the tab "Telegram" Payload to be output via Ethernet OUT port 2005 are configured.

In this example, it is the:

- Start "("
- Separator semicolon ";"
- Trailer ")"
- Detector result from detector 1
- "Pose 3D" from Detector 1

"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.

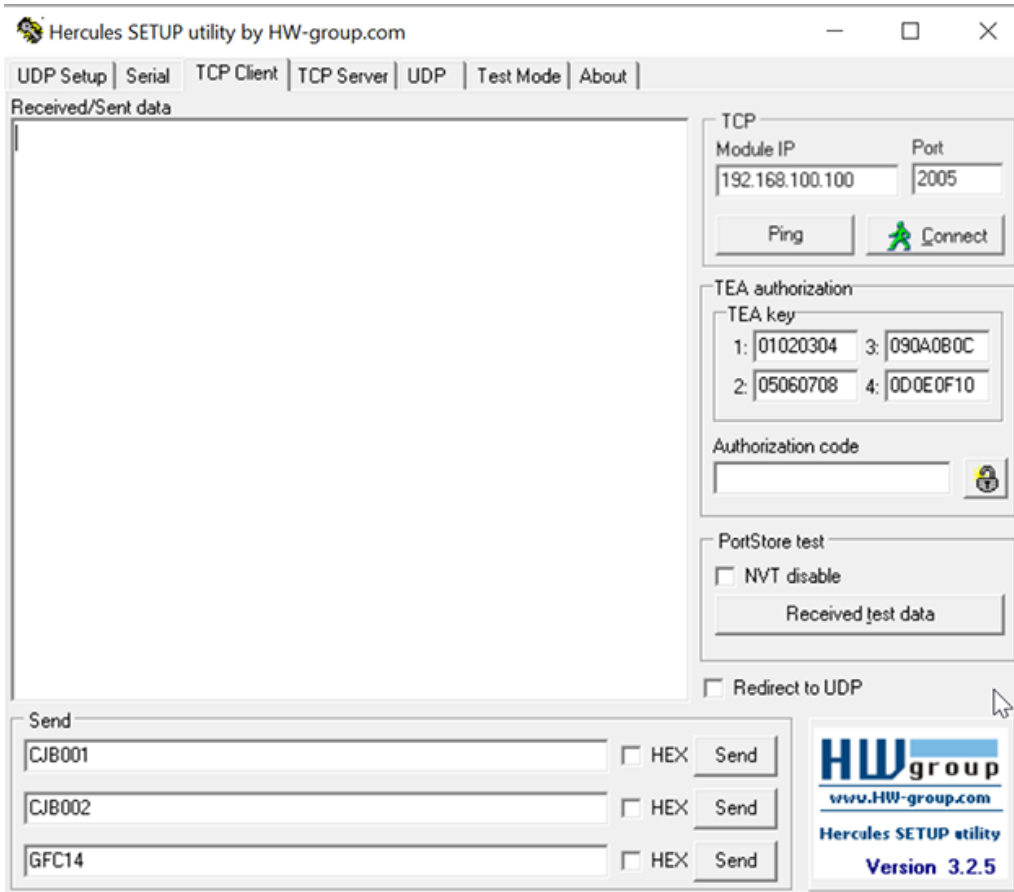


**Fig. 14:** Telegram, configure output data

**Step 3**

Sending and receiving is shown in the following example using the "Hercules Setup Utility" software (hereinafter referred to only as "Hercules") from HW-Group (HW-Group.com). With this software it is very easy to open a TCP/IP socket connection. However, other tools can also be used.

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



**Fig. 15:** Settings for TCP/IP connection using Hercules Setup Utility

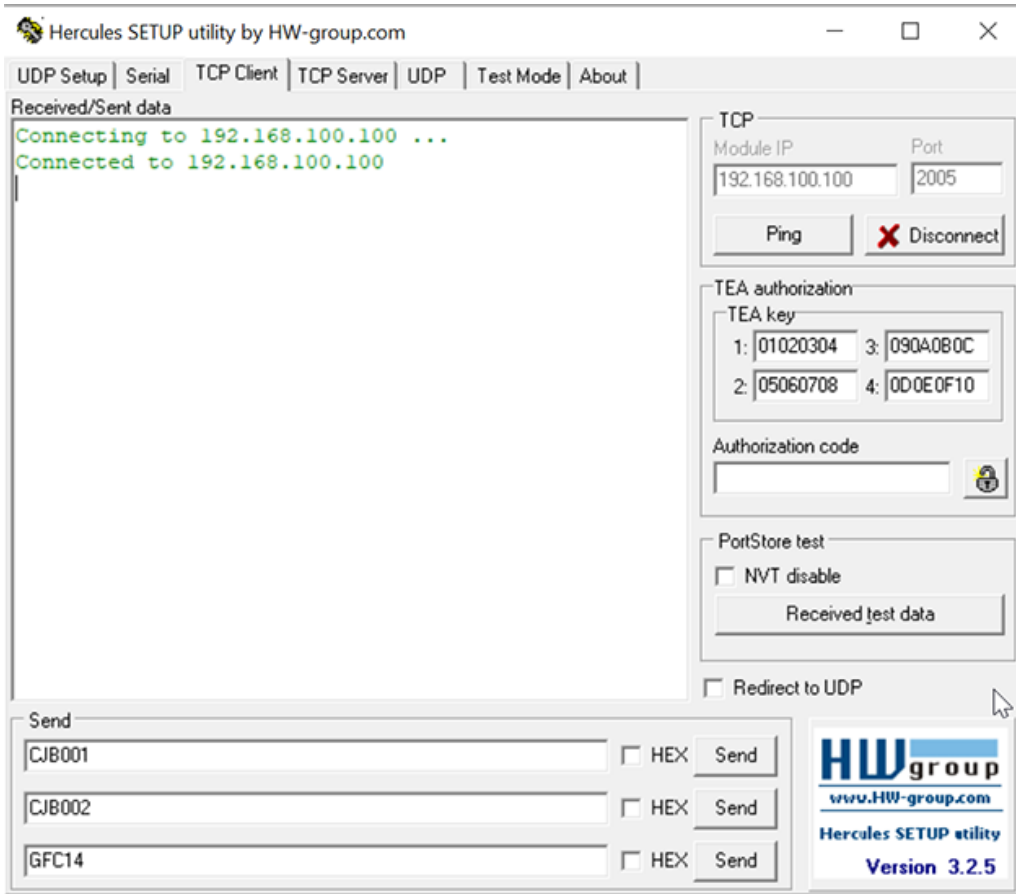
You will need to enter the IP address of the VISOR® and the correct port in order to receive data.

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

The IP address of the VISOR® vision sensor is visible in SensoFind (see the first line in the "Active sensors" window, e.g. 192.168.100.100).

**Step 4**

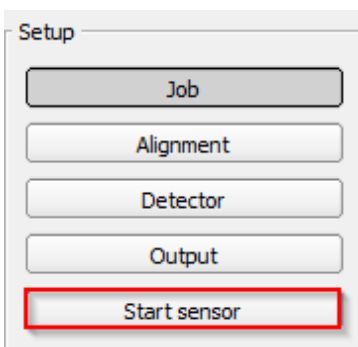
After the IP address and the port have been entered, the connection with the VISOR® vision sensor can be established via the "Connect" button. All other settings in Hercules remain at the default values. A successful connection is displayed in green letters in the main window.



**Fig. 16:** Establishing a TCP/IP connection using Hercules Setup Utility

**Step 5**

The VISOR<sup>®</sup> vision sensor can only send result data in the "Run" mode. For this purpose, the vision sensor must be started with "Starting the sensor" from the PC application SensoConfig. In the later operating case, the VISOR<sup>®</sup> is in normal operation immediately after power-up and sends data if configured.



**Fig. 17:** Starting the sensor

In this example, Trigger mode = Trigger is set, so evaluations only take place and data is sent after a trigger signal. In this example, the VISOR<sup>®</sup> vision sensor is triggered by an external sensor via the hardware inputs. These are only visible in the main window of Hercules.





### 4.3 Example: Communication via one port, bidirectional

#### Telegrams (requests) from PC / controller to VISOR® via IN port; data from VISOR® to PC / controller via IN port

In the previous **example**, a trigger was sent via port 2006 and the user data was received via port 2005. It is possible to receive the payload on port 2006 as well. For this purpose, an extended trigger (TRX) is sent instead of a normal trigger (TRG). Payload is appended to the telegram response.

With the aid of the extended trigger, the VISOR® vision sensor can additionally be sent any string, which can then be used as the file name of the image files to be archived.

If this function is not required, the telegram to be sent is TRX00. The response to an extended trigger TRX is correspondingly longer than for a normal trigger TRG.

Additional information: [Overview](#)

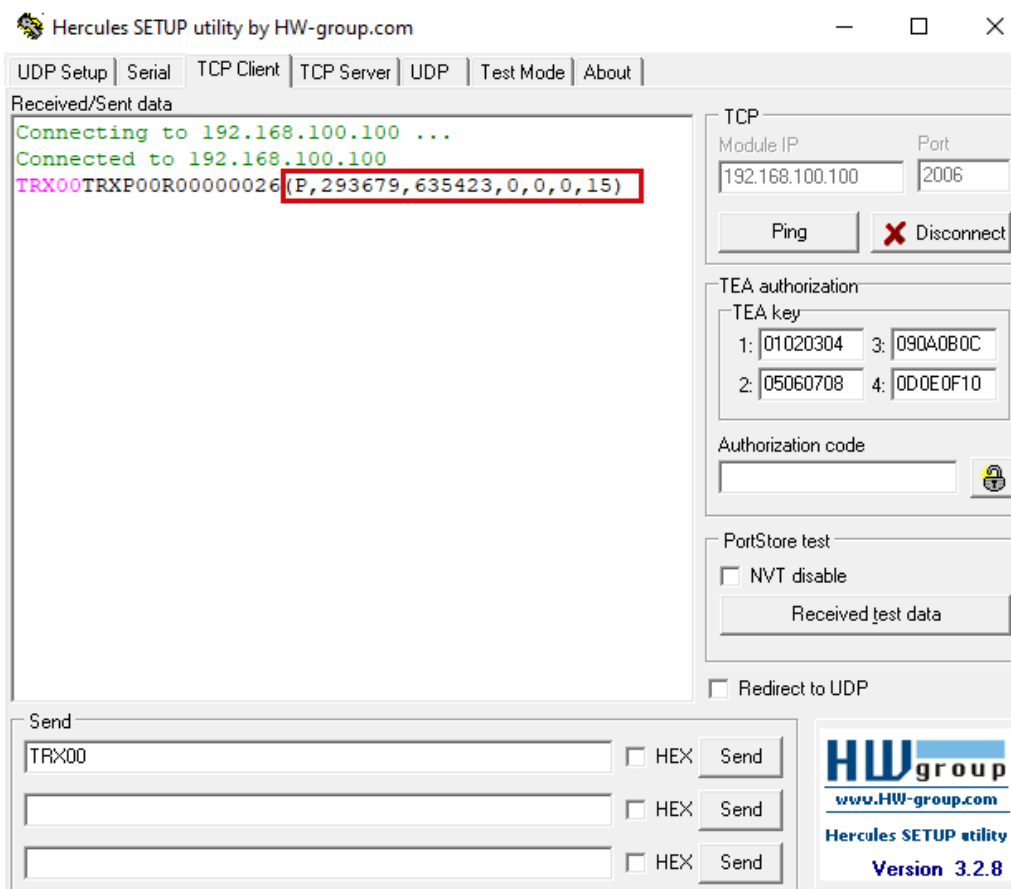


Fig. 21: Sending and receiving data on port 2006 in Hercules Setup Utility

### 4.4 More examples

For better clarity, the Trigger mode = "Trigger" is set here. The Ethernet interface must be activated (operating step Output / tab Interfaces / Ethernet, default ports, format ASCII).

For better readability of the examples all telegrams are sent in ASCII format.

### 4.4.1 Example: Job change from PC / controller to VISOR® (CJB)

#### Step 1

For this example, at least two jobs must be created on the VISOR® vision sensor. To create a new job based on an existing job, you can use the "Duplicate" function.

In SensoFind or SensoView, the last job executed is displayed in the lower left corner, among others.

One way of easily checking the job change is, for example, to use significantly different shutter speeds so that it is obvious from the brightness of the image that a job change has taken place (e.g. job 1 very dark, job 2 very bright). Alternatively, different settings could be selected for the data output.

After two jobs have been created, start the vision sensor. Leave SensoConfig open to check if a job change has occurred.

#### Step 2

Open the Hercules application.

To trigger a job change via Ethernet (TCP/IP), the telegram Job change by index (CJB) can be used for example. The telegram CJB is followed by the three-digit index of the job number from 001 to 255. To change to job 1, "CJB001" must be sent, for job 2 "CJB002" etc.

Additional information: [Overview](#)

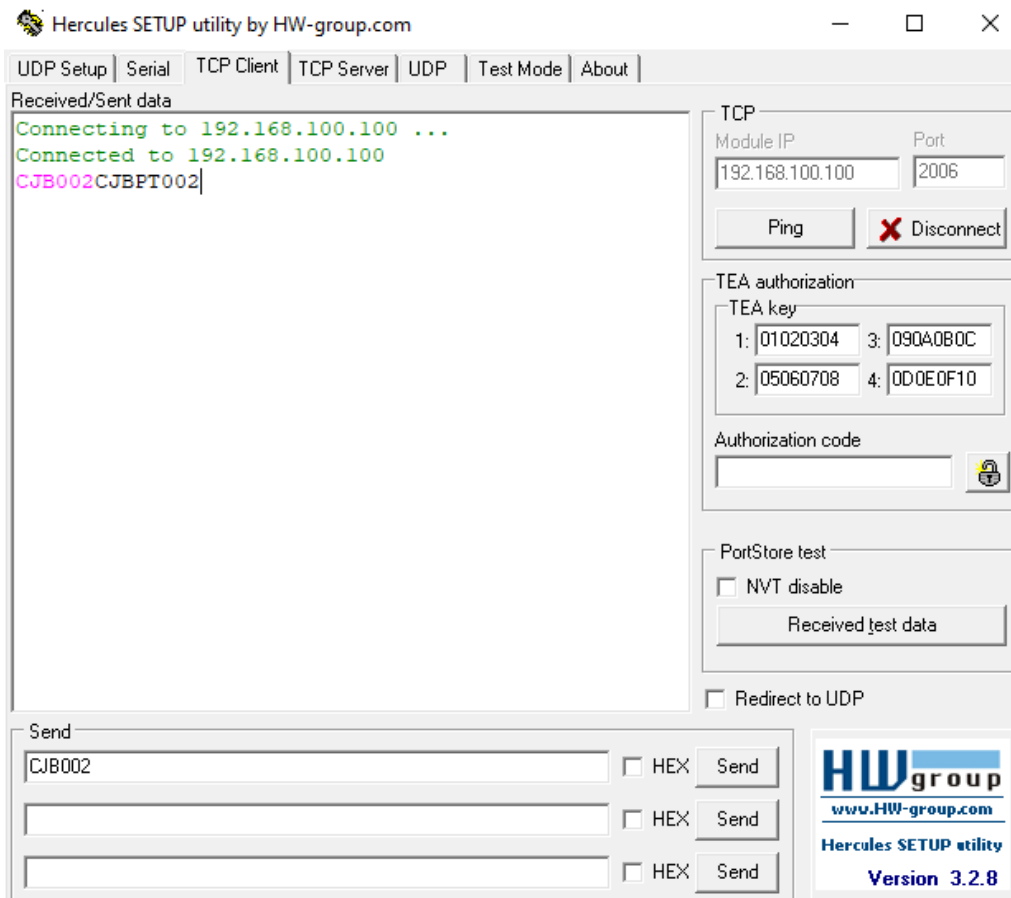
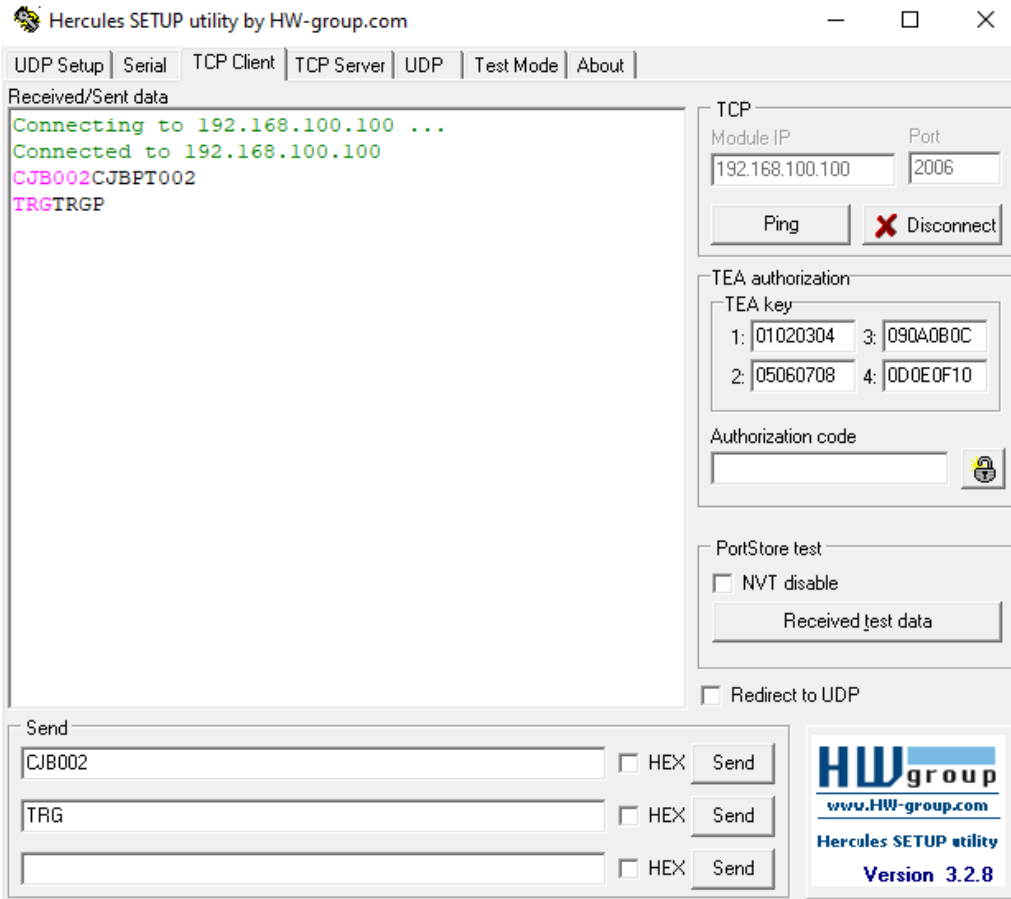


Fig. 22: Sending the "Job change" telegram in Hercules Setup Utility

The sent telegram is displayed in pink again. The telegram response shows that the VISOR® has successfully accepted the request and that job 2 will be executed on the next trigger.

In SensoConfig, SensoView and SensoConfig Web the last executed job is displayed. After a successful job change, the old job is still displayed in the software. After a trigger the display is updated. Therefore after the request CJB a TRG must be sent so that the VISOR® executes a new evaluation.



**Fig. 23:** Sending the "Trigger" telegram after a job change in Hercules Setup Utility

If the data output has been set differently, it can also be detected from the sent detector-specific user data that a job change has taken place. In the following screenshot, leader, trailer and separator have been defined differently for Job 1 and Job 2:

Job 1:

- Start "("
- Separator ";"
- Trailer ")"

Job 2:

- Start "<"
- Separator "/"
- Trailer ">"

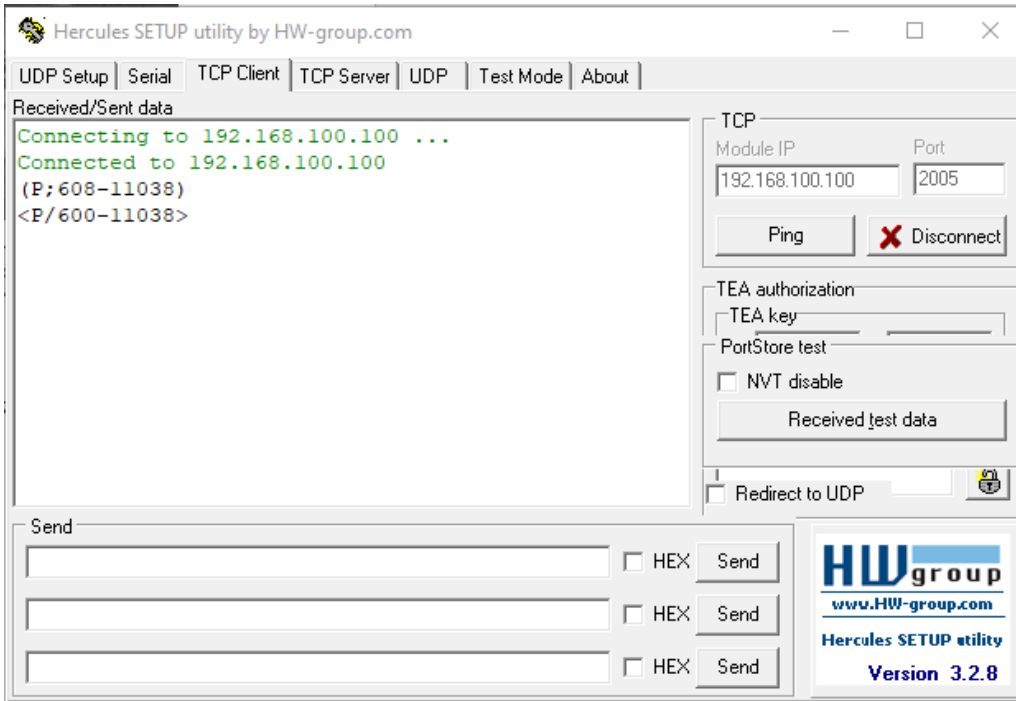


Fig. 24: Comparison of different payload data after "Job change" telegram in Hercules Setup Utility

**4.4.2 Example: Set shutter speed (SST/SSP) / read (GSH)**

To change the shutter speed of the vision sensor, the telegram "Set shutter speed" (temporary until next device restart: SST / permanent: SSP) is used.

To read the current value of the shutter speed, the telegram "Get shutter speed" (GSH) is used.

Additional information: [Overview](#)

In the following example, the shutter speed is permanently set to 4.25 ms. The millisecond value must be multiplied by a factor of 1000. With a target new shutter speed of 4.25 ms, the value to be sent is thus 4250. Thus, a four-digit number is sent for the parameter "Number of digits of shutter speed value": SSP044250.

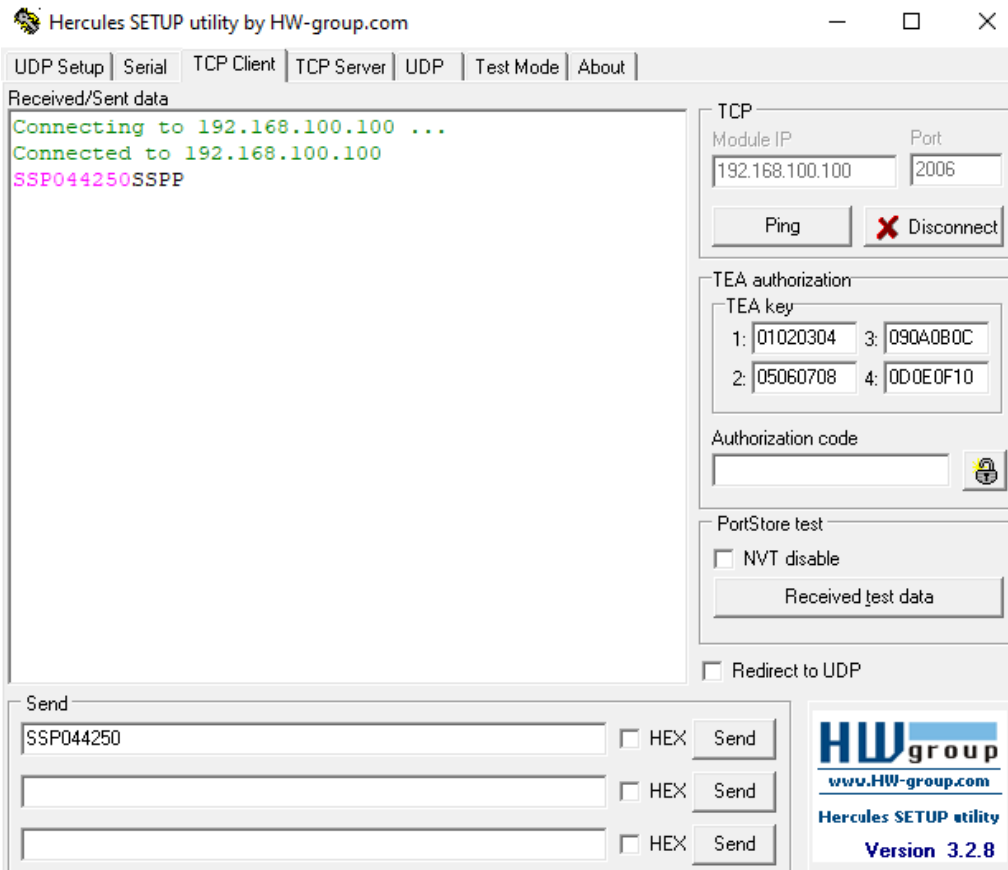


Fig. 25: Sending the "Set parameter" telegram in Hercules Setup Utility

The telegram Set shutter speed - permanent (SSPP) is sent as a response from VISOR®. The telegram was accepted.

The GSH telegram can now be used to check whether the set shutter speed value was really set.

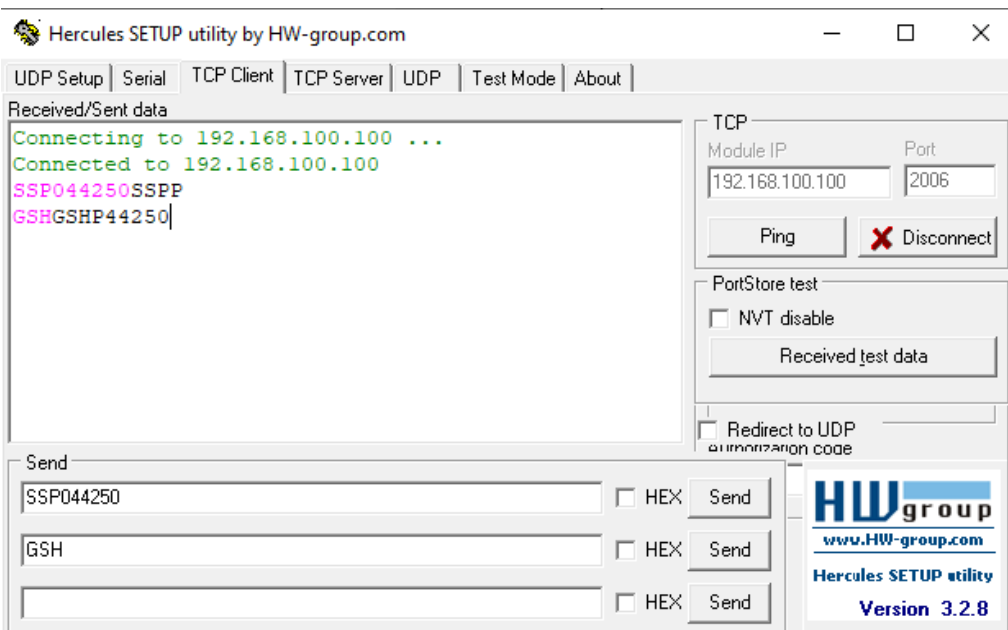


Fig. 26: Sending the "Read shutter speed value" telegram in Hercules Setup Utility

The response string shows that the command was accepted successfully, that a four-digit shutter speed value was sent and that this has the value 4250. This value must now be divided by 1000 to arrive at the correct value:  $4250 / 1000 = 4.25$  ms.

Alternatively, you can check whether the shutter speed has been set in SensoConfig: Job / Image acquisition / Shutter speed parameter.

## 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

- **Load jobset** ([ASCII](#))  
The "Load jobset" telegram can be used to change the VISOR® vision sensor's job set. The jobset file must first be loaded onto the VISOR®.
- **Get jobset** ([ASCII](#))  
The "Get jobset" telegram can be used to read the VISOR® vision sensor's jobset.

### 5.2 Visualization

---

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

Additional information: [Update visualization data \(ASCII\)](#)

## 6 VISOR® telegrams for PROFINET and EtherNet/IP

6.1 Module 1: "Control" (From PLC to VISOR®)	29
6.2 Module 2: "Status" (from VISOR® to PLC)	31
6.3 Module 3: "Data" (from VISOR® to PLC)	33
6.4 Module 4: "Request" (From PLC to VISOR®)	34
6.5 Module 5: "Response" (From VISOR® to PLC)	35
6.6 Start / end criteria for each telegram	36

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

Name in PLC "CTRL (3 bytes)"


Byte Offset	Bit Adr..	Name	Data type	Meaning
0	0	Reset error	1 bit	Reset Error clears the 4 bit error code in the "Status" module. Rising edge (False → True) clears error code.
	1	Trigger Disable	1 bit	This bit is used to disable the trigger. Valid for Trigger mode Trigger and Free run. <ul style="list-style-type: none"> <li>False (0): Trigger activated.</li> <li>True (1): Trigger deactivated.</li> </ul> If the digital input "Trigger enable" is used, both conditions (digital input "Hardware Trigger" and "Trigger Disable Bit") must be set to "Enable" to accept a trigger.
	2	Trigger	1 bit	Rising edge (False → True): Trigger is executed immediately. If the trigger could not be executed, the Trigger acknowledge Bit stays false and "Error status" module has the error code "1: Failure trigger request". See also Timing diagram, Chapter <a href="#">Case: Trigger not possible (not ready)</a>
	3	Change job	1 bit	Rising edge (false → true): Switch to job with number "Job number" from Control module. When executing this request, delays may occur. After a successful job change, the "Job number" byte in the "Status" module shows the same value as in the Control module. If the job change could not be executed due to error (due to an error, e.g. wrong job number), the "Error status" module has the error code "2: Failure change job" (and Ready stays false!). See also Timing diagram, Chapter <a href="#">Case: Job change not possible (e.g. wrong job number)</a>
	4	Switch-to-Run	1 bit	Rising edge (False → True): "Switch-to-Run" is executed. Success or failure of Switch to Run request is shown in the "Error status" module (error code "3: Failure Switch to run request") and bit "Operation Mode". See also Timing diagram, Chapter <a href="#">Case: Switch to run not possible</a>
5-7		Reserve		

Byte Offset	Bit Adr..	Name	Data type	Meaning
1		Reserve	1 byte	
2		Job number	U8	Job number to be switched to, on the rising edge of the change job bit. Binary value 1-255 for "Job number change". 0 stands for "No switching", even if the Change Job Bit changes.

[Timing diagrams for VISOR® communication](#)

## 6.2 Module 2: “Status” (from VISOR® to PLC)

Name in PLC "STAT (6 bytes)"

Byte Offset	Bit Adr.	Name	Data type	Description
0	0	Ready	1 bit	VISOR® ready for next evaluation. Ready=1.   <b>Attention:</b> The Ready bit is exclusively reserved for indicating the readiness of the VISOR® vision sensor for the next evaluation. It is not suitable for indicating that an evaluation has been completed or the results of an evaluation are available!
	1	Reserve	1 bit	
	2	Trigger acknowledge	1 bit	Acknowledge (confirmation) for successful trigger request (via Trigger Bit in Control module). Acknowledge is deleted as a response to the deletion of the trigger bit. If the trigger could not be executed, the Trigger Acknowledge Bit stays false.
	3	Change Job acknowledge	1 bit	Acknowledge (confirmation) for the Change Job Request (via Change Job Bit in Control module) – independent of its success. Acknowledge is deleted as soon as the Change Job Request Bit has been deleted. Success or failure of Change Job Request is shown in the bitfield "Error" (error code "2: Failure change job") and in the byte "Job number" in the Status module. If there are delays in executing the job change, this acknowledge bit can also be set with a delay.
	4	Switch to run acknowledge	1 bit	Acknowledge (confirmation) for the Switch to Run Request (via Switch to Run Request Bit in the Control module). Acknowledge is deleted as soon as the Request Bit is deleted. Success or failure of Switch to Run Request is shown in the bitfield "Error" (error code "3: Failure Switch to run request") and bit "Operation Mode". Acknowledge is set after SensoConfig is closed and the job has been loaded from the flash or if an error has occurred.
	5-7	Reserve		
1		Reserve	1 byte	

Byte Offset	Bit Adr.	Name	Data type	Description
2	0	Digital Results	1 bit	12 RDBU
	1		1 bit	09 RD
	2		1 bit	05 PK
	3		1 bit	06 YE
	4		1 bit	07 BK
	5		1 bit	08 GY
	6	Reserve	1 bit	This byte is filled with the results of the digital switching outputs. The bit position is fixed. The value of the output is defined in the tab: Output/Digital output, Column: "Logical expression" in SensoConfig. If not selected as result output pin, or if no valid logical expression is assigned, the value is = 0.
	7	Reserve	1 bit	
3		Job number	U8	Number of current job: Job number 1-255
4		Image ID	U8	Image ID (0 - 255) is incremented by 1 with each job execution, independent of the trigger source.
5	0-3	Error	4 bit	4 bit error code (decimal). Used to indicate errors in requests via the control module or VISOR® system errors. The error code can be reset by "Reset error" or is overwritten by the next error. In case of an archiving error (8), you can continue without a "Reset error". 0: No error 1: Trigger request error (sensor not Ready) 2: Error: Change job 3: Error: Switch-to-Run 4: Request rejected 5: Error, no job on the device 7: Focus lock time 8: Error: Archiving 15: System error
	4	Trigger mode	1 bit	1 = Free run 0 = Trigger
	5	Reserve	1 bit	
	6	Operation mode	1 bit	1 = Run 0 = Config
	7	Reserve	1 bit	

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

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

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

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

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

Byte Offset	Bit Adr.	Name	Data type	Meaning
0	1	Key	1 byte	Request key (Request counter)
1	1	Reserve	1 byte	Reserve
2	1	Reserve	1 byte	Reserve
3	1	Reserve	1 byte	Reserve
4		Request Data	Byte array	Additional information: <a href="#">Overview</a>

## 6.5 Module 5: “Response” (From VISOR® to PLC)

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

Byte Offset	Bit Adr.	Name	Data type	Description
0		Key	U8	Response key = mirrored from request
1	0	Result Data overrun	1 bit	Response data has been truncated
	1-7	Reserve	7 Bit	
2		Reserve	1 byte	
3		Reserve	1 byte	
4		Result Data	Byte array	Additional information: <a href="#">Overview</a>

## 6.6 Start / end criteria for each telegram

Telegram ("Control" module)	Start- condition (Modul "Status")	Confirmation of acceptance (Modul "Status")	Confirmation of execution (Modul "Status")
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 Telegrams

7.1 Overview .....	37
7.2 Availability and supported interfaces .....	41
7.3 Sequences .....	45
7.4 Description Telegrams ASCII .....	57
7.5 Description Telegrams BINARY .....	141
7.6 Error codes .....	205

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

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

The following settings are possible:

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

### 7.1 Overview

#### Availability and supported interfaces

#### VISOR® General

- **Reset statistics (RST)** ([ASCII](#) / [Binary](#))  
The "Reset statistics" telegram can be used to reset the VISOR® vision sensor's internal statistics counter.
- **Process image from file (PIF)** ([ASCII](#) / [Binary](#))  
With the "Process image from file" telegram, the VISOR® vision sensor can process an image file instead of a live image for simulation and test purposes.

#### VISOR® Control

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

- **Set Trigger ID (STI)** ([ASCII](#) / [Binary](#))  
 With the telegram "Set Trigger ID" a Trigger Identifier can be set. The identifier is used for the next image acquisition and can be set, for example, as a file name.
- **Set robot position (GRP)** ([ASCII](#) / [binary](#))  
 An image acquisition can be started with the telegram "Read robot position". The VISOR<sup>®</sup> provides a robot position in which the VISOR<sup>®</sup> is aligned at the desired distance and orthogonally. The telegram can be used with Hand-Eye calibration and the Base-Eye calibration.
- **Job change (CJB)** ([ASCII](#) / [Binary](#))  
 The "Job change" telegram will trigger a job change on the VISOR<sup>®</sup> vision sensor.
- **Change job permanent (CJP)** ([ASCII](#) / [Binary](#))  
 The "Job change permanent" telegram will trigger a permanent job change on the VISOR<sup>®</sup> vision sensor. The job is run again after restarting.
- **Job change by job name (CJN)** ([ASCII](#) / [Binary](#))  
 The "Job change by job name" telegram will trigger a job change on the VISOR<sup>®</sup> 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<sup>®</sup> Job settings

- **Create job template (MJT)** ([ASCII](#) / [Binary](#))  
 The "Create job template" telegram can be used to specify the job template automatically.
- **Auto Working distance (AFC)** ([ASCII](#) / [Binary](#))  
 The "Auto working distance" telegram can be used to have the working distance for the job be automatically determined.
- **Set working distance (SFC)** ([ASCII](#) / [Binary](#))  
 The "Set working distance" telegram can be used to change the working distance for the job.
- **Read working distance (GFC)** ([ASCII](#) / [binary](#))  
 The "Read working distance" telegram can be used to read the current working distance for the job.
- **Auto shutter speed (ASH)** ([ASCII](#) / [Binary](#))  
 The "Auto shutter speed" telegram can be used to have the shutter time for the job be automatically determined.
- **Set shutter speed (SSP/SST)** ([ASCII](#) / [Binary](#))  
 With the telegram "Set shutter speed", the shutter speed of the job can be changed. This telegram can, for example, be used for brightness compensation.
- **Read shutter speed (GSH)** ([ASCII](#) / [Binary](#))  
 With the telegram "Read shutter speed", the set shutter speed of the job can be read.
- **Set gain (SGA)** ([ASCII](#) / [binary](#))  
 With the telegram "Set gain", the gain of the job can be changed. This telegram can, for example, be used for brightness compensation.
- **Read gain (GGA)** ([ASCII](#) / [binary](#))  
 With the telegram "Read gain", the set gain of the job can be read.
- **Set parameters (SPP/SPT)** ([ASCII](#) / [binary](#))  
 With the telegram "Set parameter", the detector parameters can be adjusted, e.g. reference strings, detector thresholds.
- **Read parameters (GPA)** ([ASCII](#) / [binary](#))  
 With the telegram "Read parameter", the set parameters of the detectors can be read.
- **Set region of interest / ROI (SRP/SRT)** ([ASCII](#) / [binary](#))  
 With the telegram "Set ROI", the position of the selected detector can be changed.
- **Read region of interest / ROI (GRI)** ([ASCII](#) / [Binary](#))  
 With the telegram "Read ROI", the position of the selected detector can be read.

- **Set ROI content (SRC)** ([ASCII](#) / [Binary](#))  
With the telegram "Set ROI content" the image to be taught in can be defined, the mask can be set and for the detector "Contour" the taught-in contour can be set.
- **Read ROI content (GRC)** ([ASCII](#) / [Binary](#))  
With the telegram "Read ROI content" a mask, pattern or contour file can be read in.
- **Read job list (GJL)** ([ASCII](#) / [binary](#))  
The "Get job list" telegram can be used to output a list of all available jobs on the VISOR® vision sensor.
- **Read detector list (GDL)** ([ASCII](#) / [binary](#))  
With the telegram "Read detector list", a list of all detectors in the current job will be displayed.
- **Teach detector (TED)** ([ASCII](#) / [binary](#))  
The "Teach detector" telegram will result in the specified detector being re-taught (available only for Pattern matching, Contour and Barcode).
- **Set trigger delay (STD)** ([ASCII](#) / [Binary](#))  
With the telegram "Set trigger delay", a delay for starting a trigger can be set (in time (ms) or encoder steps).
- **Read trigger delay (GTD)** ([ASCII](#) / [Binary](#))  
With the telegram "Read trigger delay", the set delay for starting a trigger can be read.
- **Save Job Permanently (SJP)** ([ASCII](#) / [binary](#))  
The "Save job permanently" telegram will take all the parameters that were previously set temporarily and copy them to a job set.
- **Set image acquisition parameters (SAP)** ([ASCII](#) / [Binary](#))  
The "Set image acquisition parameters" telegram can be used to set parameter values for image acquisition.
- **Read image acquisition parameters (GAP)** ([ASCII](#) / [Binary](#))  
The "Read image acquisition parameters" telegram can be used to read the set parameter values of the image acquisition.
- **Data output - Set parameters (STP)** ([ASCII](#) / [Binary](#))  
The "Data output - Set parameters" telegram can be used to set the set parameter values of the data output.
- **Data output - Read parameters (GTP)** ([ASCII](#) / [Binary](#))  
The "Data output - Read parameters" telegram can be used to read the set parameter values of the data output.

## VISOR® Calibration

- **Calibration: Initialization (CCD)** ([ASCII](#) / [Binary](#))  
The point pair list is initialized with the telegram "Calibration: Initialize point pair list".
- **Calibration: Add world point (CAW)** ([ASCII](#) / [Binary](#))  
With the telegram "Calibration: Add world point" a world point (fiducial or point pair) is added to the point pair list. The telegram can be used for the calibration method Point pair list (Robotics) and Calibration plate (Robotics).
- **Calibration: Point pair list (CCL)** ([ASCII](#) / [Binary](#))  
With the telegram "Calibration: Point pair list" the calibration is carried out using the point pair list in the current job.
- **Calibration: Validate point pair list (CVL)** ([ASCII](#) / [Binary](#))  
With the telegram "Calibration: Validate point list", the calibration is validated using the point list.
- **Calibration: Calibration plate (CCP)** ([ASCII](#) / [Binary](#))  
With the telegram "Calibration: Calibration plate", the calibration is carried out using the calibration plate.
- **Set fiducial (CSF)** ([ASCII](#) / [binary](#))  
With the telegram "Set fiducial", the fiducials are set using the point list in the current job.
- **Calibration: Add image (CAI version 1)** ([ASCII](#) / [Binary](#))  
The "Add image version 1" telegram triggers an image acquisition and if a calibration plate is found, an image is added to the calibration object. The telegram can be used for calibration method Multi-image calibration and calibration method Calibration plate (Robotics).

- **Calibration: Add image (CAI version 2)** ([ASCII](#) / [Binary](#))  
The "Add image version 2" telegram triggers an image acquisition and if a calibration plate is found, an image is added to the calibration object. The telegram can be used for calibration method Multi-image calibration and calibration method Calibration plate (Robotics). In addition, a trigger is executed. This causes the image to be displayed and saved if necessary.
- **Calibration: Multi-Image (CMP)** ([ASCII](#) / [Binary](#))  
With the telegram "Calibration: Multi-image" a calibration is carried out and an existing calibration object is accessed.
- **Calibration: Robotics Multi-Image (CRP)** ([ASCII](#) / [Binary](#))  
With the telegram "Multi-image, robot" a calibration is carried out using the calibration plate.
- **Calibration: Copy calibration (CCC version 1)** ([ASCII](#) / [Binary](#))  
With the telegram "Calibration: Copy calibration version 1", the calibration of the current job is completely copied to the selected destination.
- **Calibration: Copy calibration (CCC version 2)** ([ASCII](#) / [Binary](#))  
With the telegram "Calibration: Copy calibration version 2", the calibration of the current job is copied to the selected destination. In addition, it is possible to select which part of the calibration is copied.
- **Calibration: Set parameters (CSP)** ([ASCII](#) / [Binary](#))  
With the telegram "Calibration: Set parameter", the parameter values for the calibration can be set.
- **Calibration: Read parameters (CGP)** ([ASCII](#) / [Binary](#))  
With the telegram "Calibration: Read parameter", the set parameter values of the calibration can be read.

## VISOR<sup>®</sup> visualization

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

## VISOR<sup>®</sup> Service (available only on port 1998 and in ASCII format)

- **Update visualization data (UVR)** ([ASCII](#))  
The "Update visualization data" telegram is used to update visualization data such as image, detector information and results.
- **Read sensor identity (GSI)** ([ASCII](#))  
With the telegram "Read sensor identity", the current firmware status as well as the hardware type can be queried.
- **Update firmware (UFW)** ([ASCII](#))  
With the telegram "Update firmware", a firmware update is started. The firmware file must first be loaded onto the VISOR<sup>®</sup> vision sensor.
- **Load jobset (SJS)** ([ASCII](#))  
The "Load jobset" telegram can be used to change the VISOR<sup>®</sup> vision sensor's job set. The jobset file must first be loaded onto the VISOR<sup>®</sup>.
- **Get jobset (GJS)** ([ASCII](#))  
The "Get jobset" telegram can be used to provide the current job set of the VISOR<sup>®</sup> vision sensor for subsequent download.

## Telegram

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

Data output ASCII

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

Data output Binary

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



**NOTE:**

The directory **/tmp** on the VISOR® vision sensor is used for transferring files from or to the VISOR®. Files can be transferred from here / to here with an SFTP client. Access data for SFTP client:  
Username: *user*, Password: *user*.



**NOTE:**

After changing a parameter using a request with permanent storage of the parameters in VISOR®, the operating voltage must not be switched off for at least 10 seconds. Otherwise the change may be lost.

## 7.2 Availability and supported interfaces

**Device variant**

<b>ALL</b>	Allround
<b>OB</b>	Object (AI)
<b>CR</b>	Code Reader
<b>RO</b>	Robotic

**Device type**

<b>S</b>	Standard
<b>A</b>	Advanced
<b>P</b>	Professional

**Interfaces**

<b>1</b>	Ethernet TCP IN (2006)
<b>2</b>	PROFINET
<b>3</b>	EtherNet/IP
<b>4</b>	Service Port (1998)

✓ available

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

Telegram	ALL		OB		CR			RO		Interfaces				From version
	A	P	S	A	S	A	P	A	P	1	2	3	4	
<b>VISOR® General</b>														
Reset statistics (RST)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.18
Process image from file (PIF)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.0
<b>VISOR® Control</b>														
Trigger (TRG)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0
Extended trigger (TRX)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.6
Trigger Robotics (TRR)		✓						✓	✓	✓	✓	✓		2.2

Telegram	ALL		OB		CR			RO		Interfaces				From version
	A	P	S	A	S	A	P	A	P	1	2	3	4	
Set Trigger ID (STI)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.2
Read robot position (GRP)		✓							✓	✓	✓	✓		2.12
Job change (CJB)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0
Job Change Permanent (CJP)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.18
Job change by name (CJN)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.0
<b>VISOR® Job settings</b>														
Make Job Template (MJT)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.10
Auto working distance (AFC)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.0
Set working distance (SFC)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.0
Read working distance (GFC)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.0
Auto Shutter Speed (ASH)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.0
Set Shutter Speed (SSP/SST)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0
Read shutter speed (GSH)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0
Set gain (SGA)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.6
Read gain value (GGA)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.6
Set Parameter (SPP/SPT)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0
Read Parameter (GPA)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0
Set ROI (SRP/SRT)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0
Read ROI (GRI)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0
Set ROI content (SRC)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.0
Read ROI content (GRC)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.0
Read job list (GJL)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.18
Read Detector List (GDL)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.18
Teach detector (TED)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.0

Telegram	ALL		OB		CR			RO		Interfaces				From version
	A	P	S	A	S	A	P	A	P	1	2	3	4	
Set trigger delay (STD)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.22
Read Trigger Delay (GTD)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		1.22
Save Job Permanently (SJP)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.0
Set image acquisition parameters (SAP)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.6
Read image acquisition parameters (GAP)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		2.6
Data output - Set parameter (STP)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	2.10
Data output - Read parameters (GTP)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	2.10
<b>VISOR® Calibration</b>														
Initialization (CCD)	✓	✓							✓	✓	✓	✓	✓	1.18
Add world point (CAW)	✓	✓							✓	✓	✓	✓	✓	1.22
Point Pair List (CCL)	✓	✓							✓	✓	✓	✓	✓	1.18
Validate calibration (CVL)	✓	✓							✓	✓	✓	✓	✓	1.18
Calibration Plate (CCP)	[]	✓		[]					✓	✓	✓	✓	✓	1.19
Set fiducials (CSF)		✓							✓	✓	✓	✓	✓	1.22
Add image (CAI version 1)	✓	✓							✓	✓	✓	✓	✓	2.2
Add image (CAI version 2)	✓	✓							✓	✓	✓	✓	✓	2.8
Multi-Image (CMP)	✓	✓							✓	✓	✓	✓	✓	2.2
Robotics Multi-Image (CRP)		✓								✓	✓	✓	✓	2.2
Copy calibration (CCC version 1)		✓							✓	✓	✓	✓	✓	1.19
Copy calibration (CCC version 2)		✓							✓	✓	✓	✓	✓	2.8
Set parameters (CSP)	[]	✓		[]					✓	✓	✓	✓	✓	1.22
Read parameters (CGP)	[]	✓		[]					✓	✓	✓	✓	✓	1.22

Telegram	ALL		OB		CR			RO		Interfaces				From version
	A	P	S	A	S	A	P	A	P	1	2	3	4	
<b>VISOR® Visualization</b>														
Get image (GIM)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓				1.0
<b>VISOR® Service</b>														
Update visualization data (UVR)	✓	✓		✓		✓	✓	✓	✓				✓	1.22
Read sensor identity (GSI)	✓	✓		✓		✓	✓	✓	✓				✓	1.19
Update firmware (UFW)	✓	✓		✓		✓	✓	✓	✓				✓	1.19
Load jobset (SJS)	✓	✓		✓		✓	✓	✓	✓				✓	1.19
Get job set (GJS)	✓	✓		✓		✓	✓	✓	✓				✓	1.19

See also: [Overview](#)

## 7.3 Sequences

### Important recommendations for PLC programmers

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

**NOTE:**



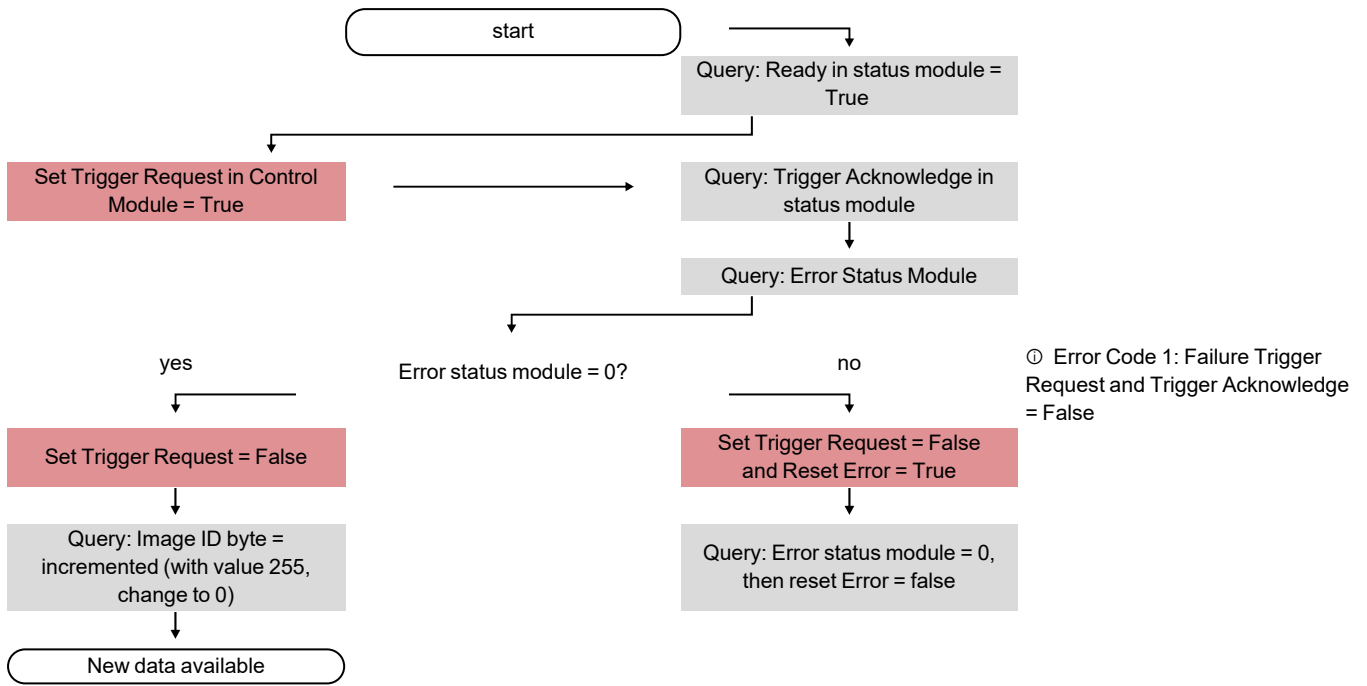
The complete execution of an action cannot be recognized as safe due to the low/high change of READY, since due to possibly long cycle times between PLC and VISOR<sup>®</sup> (e.g. 32ms), READY may never become low.

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

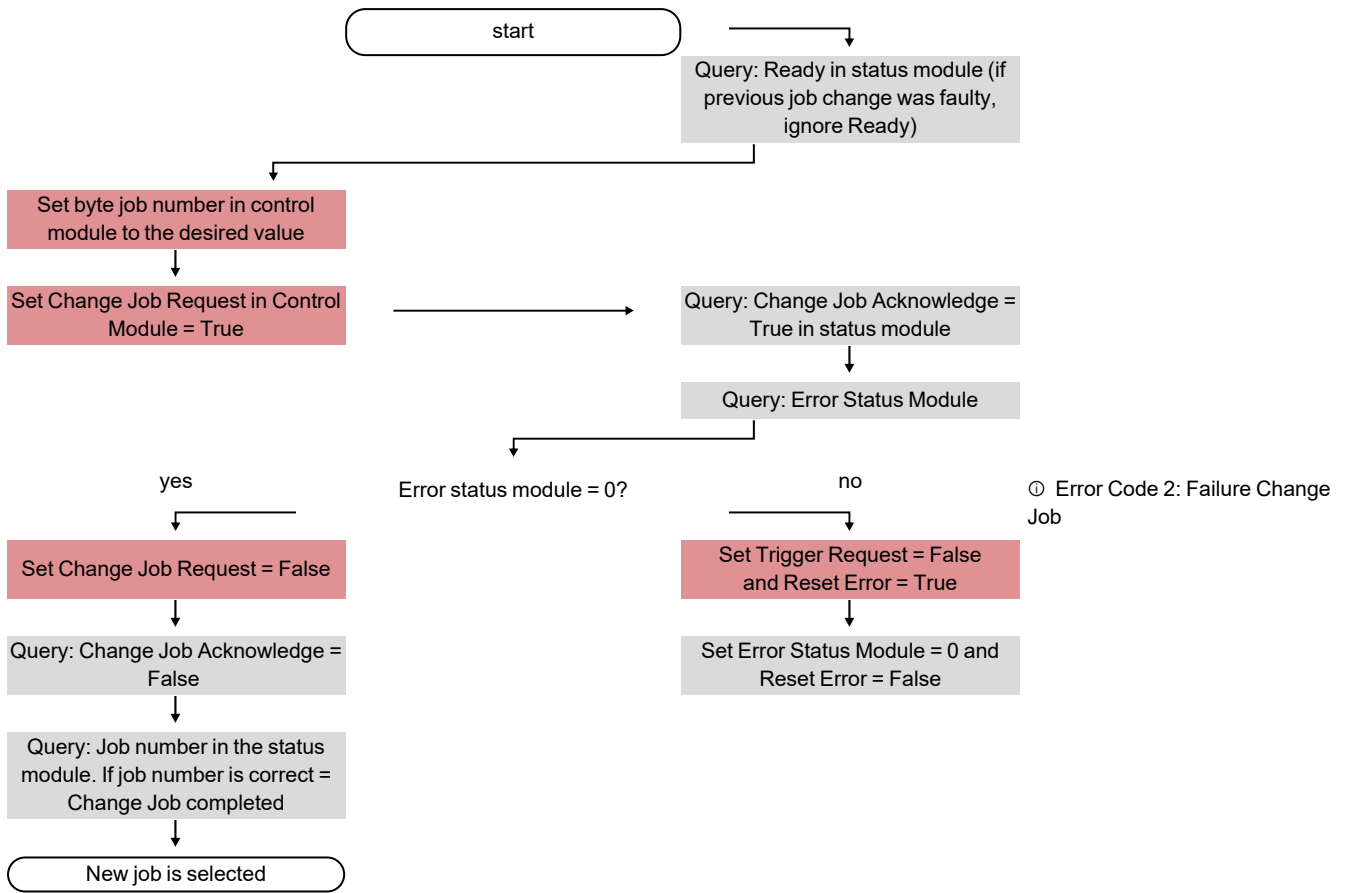
### Accepting / discarding of requests of the control module

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

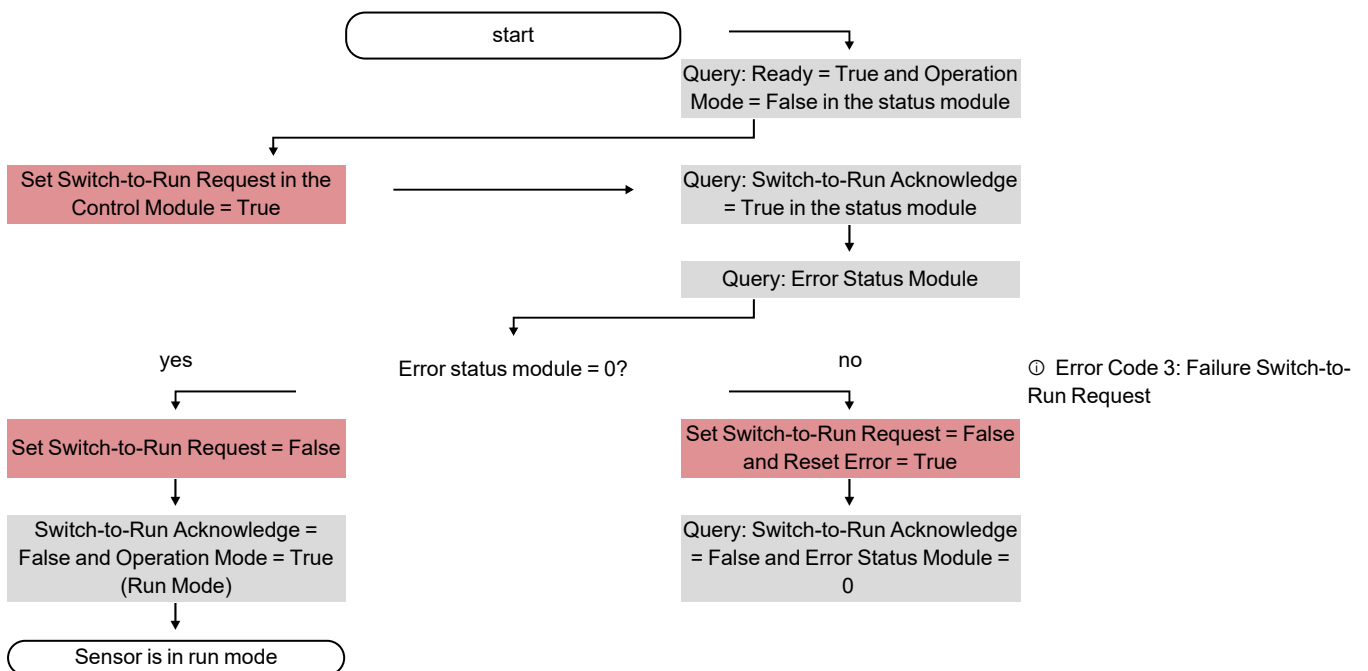
### 7.3.1 Trigger Request Sequence



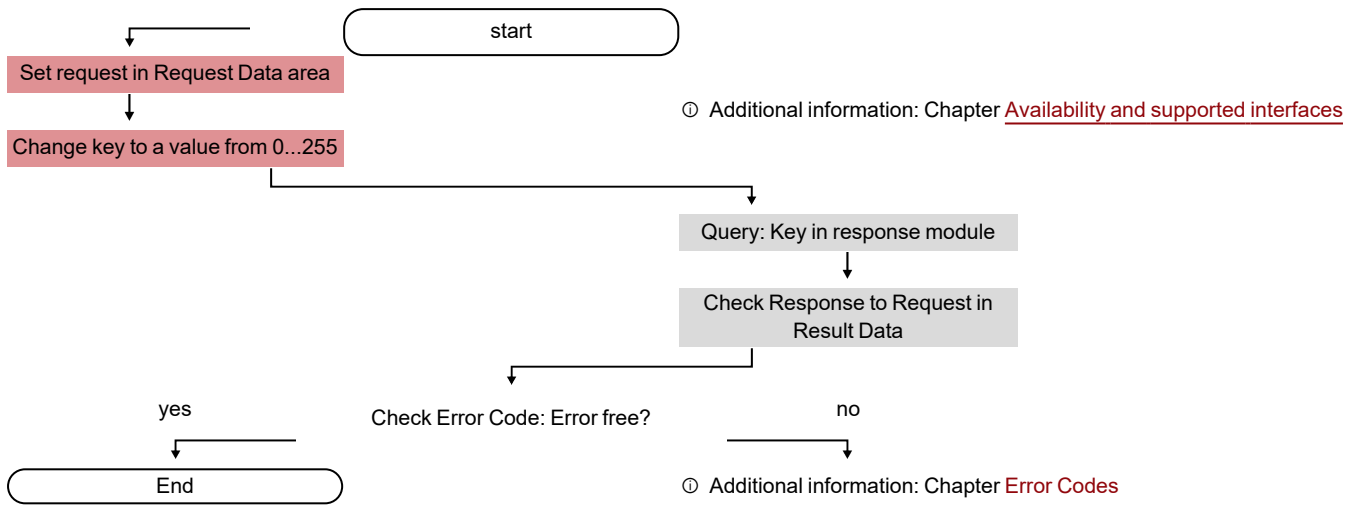
### 7.3.2 Change job request sequence



### 7.3.3 Switch to Run sequence



### 7.3.4 Sequence for requests via request/response module



**Additional information:**

[Availability and supported interfaces](#)

[Error codes](#)

**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.

### 7.3.5 Calibration via telegrams

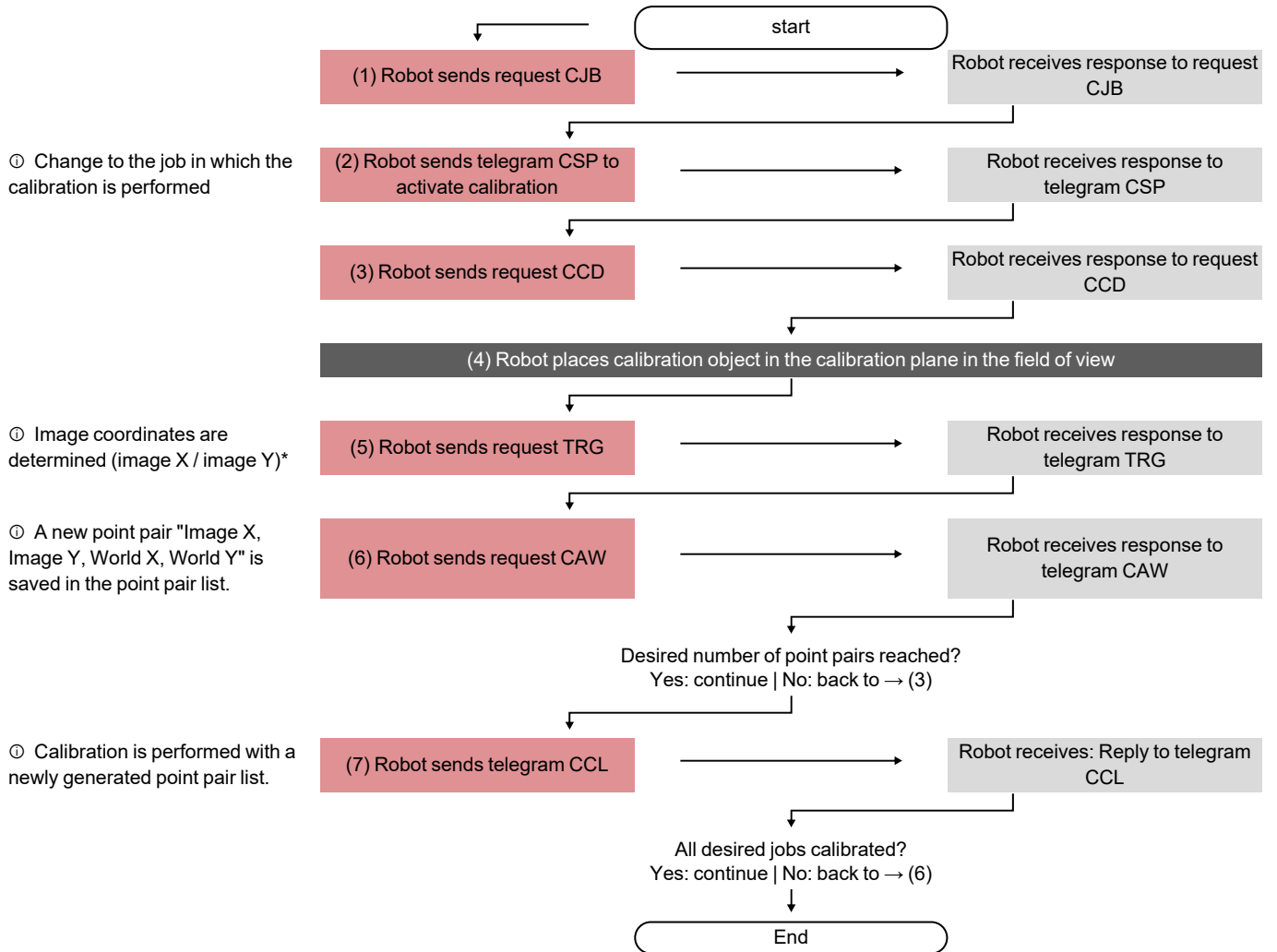
Various interface telegrams are available for the "Robotics" calibration methods (see also [Overview](#)).

The telegrams can be used for recalibration during process drift or with changed mounting situation. For example, they can be executed automatically directly from the robot controller.

**Meaning of the colors**



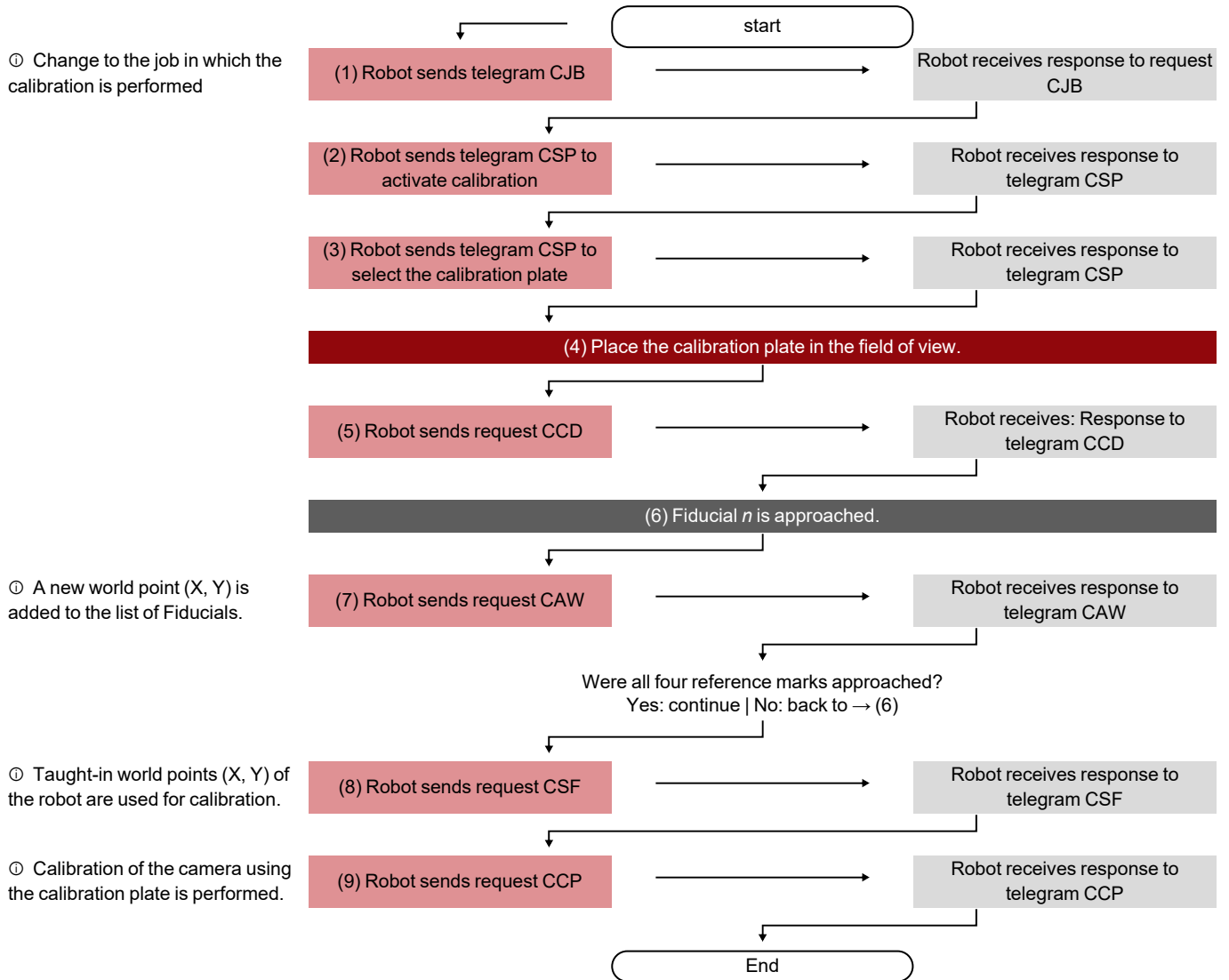
### 7.3.5.1 Automated calibration: Point pair list (Robotics)



\* To add image coordinates, the X value must be entered in line 1 and the Y value in line 2 under Output / Telegram / Payload. Furthermore, the overall job result must be positive.

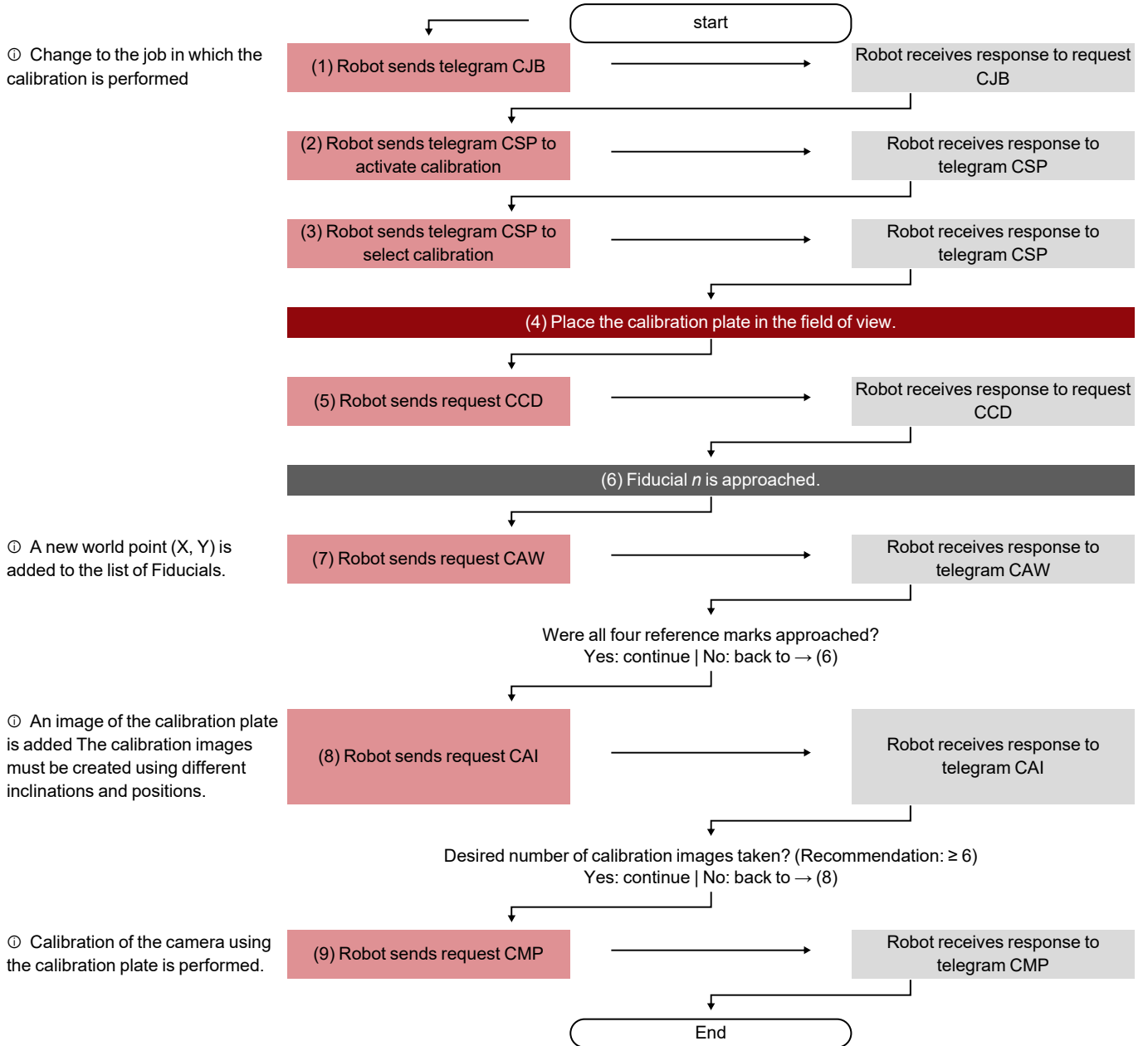
### 7.3.5.2 Automated calibration: Calibration plate (Robotics)

#### Calibration plate (Robotics) - Standard process Single image calibration

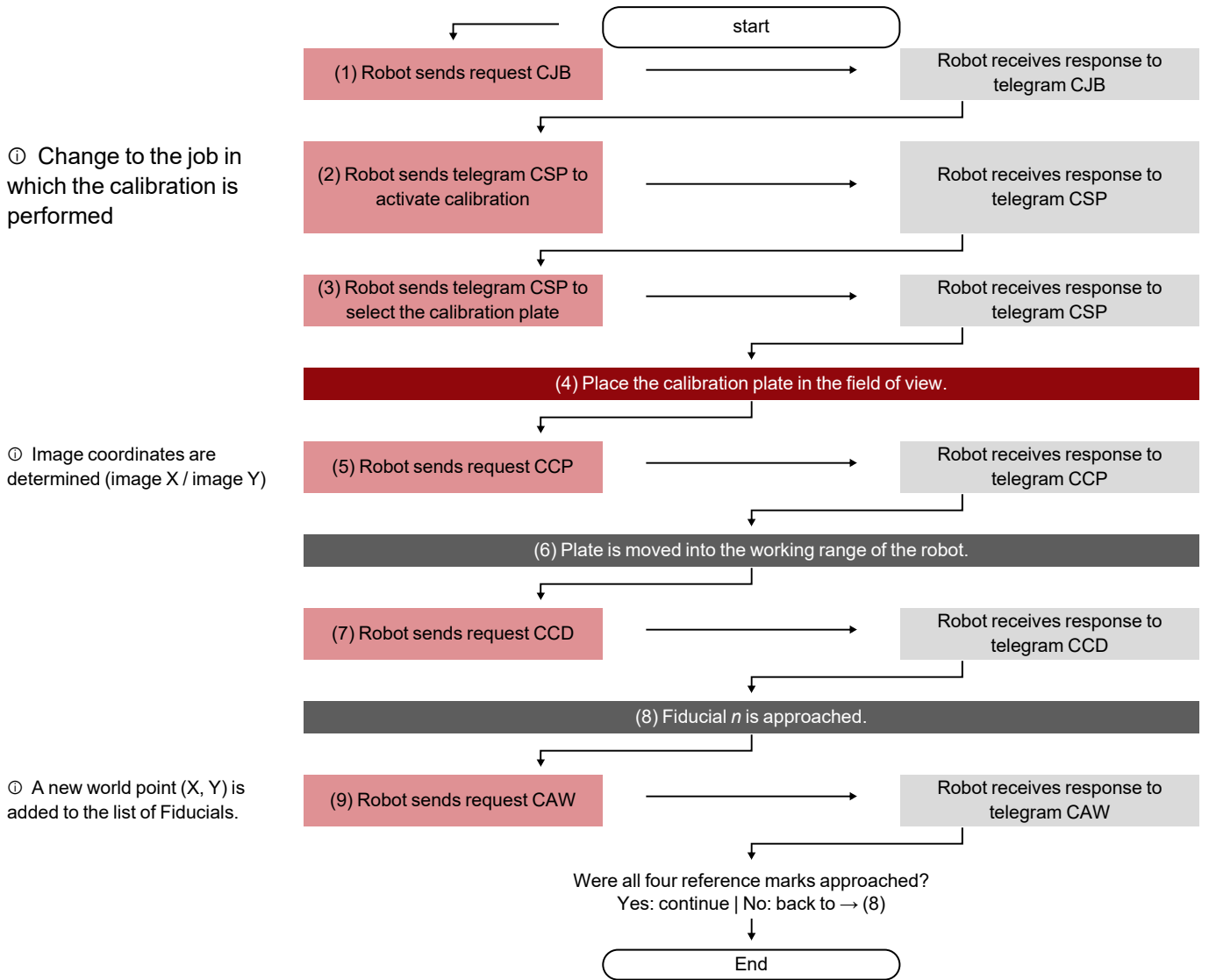


**Calibration plate (Robotics)**

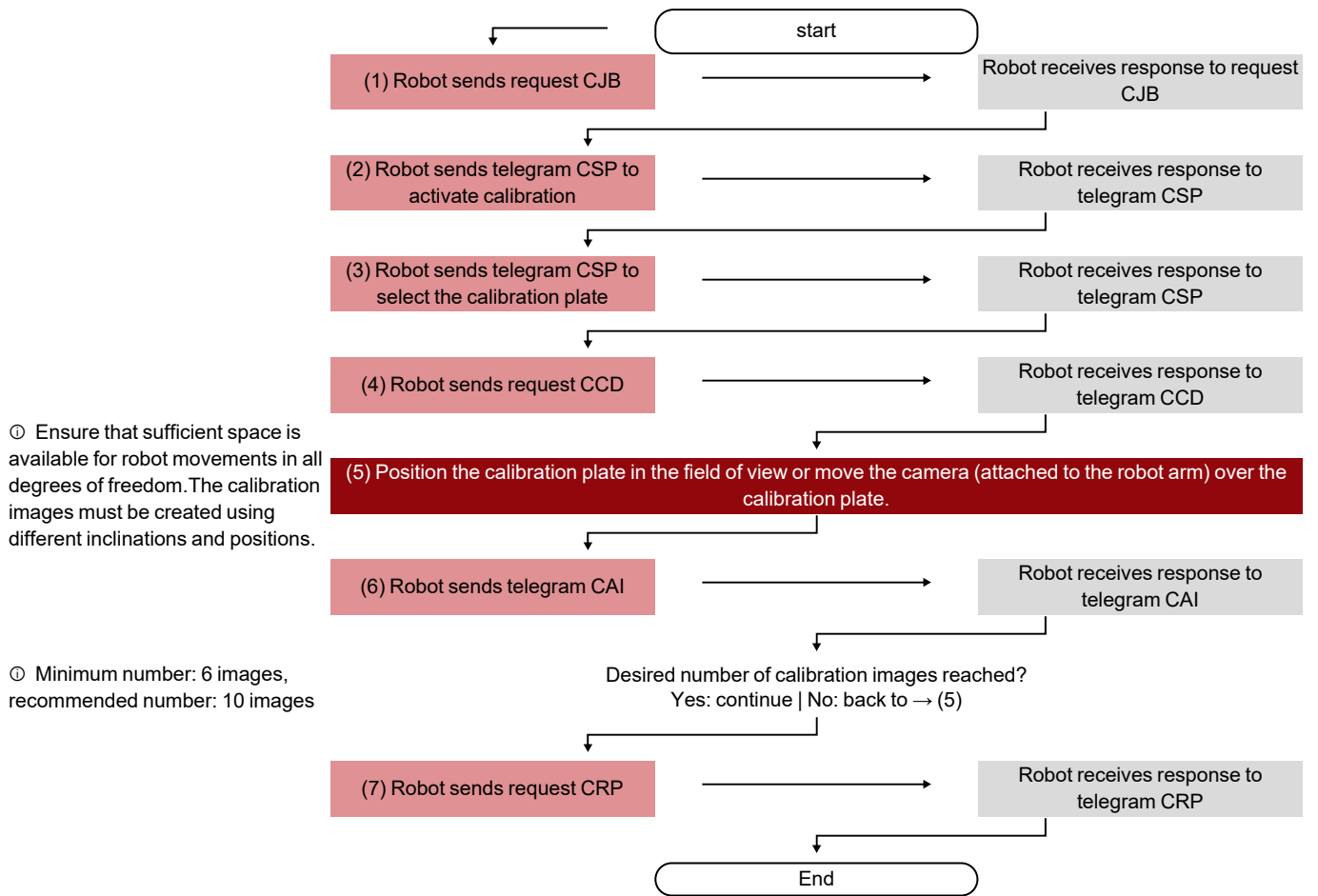
**– Standard process Multi-image calibration**



**Calibration plate (Robotics) AI – special case: Separate working area and field of view (e.g. applications with a conveyor belt)**

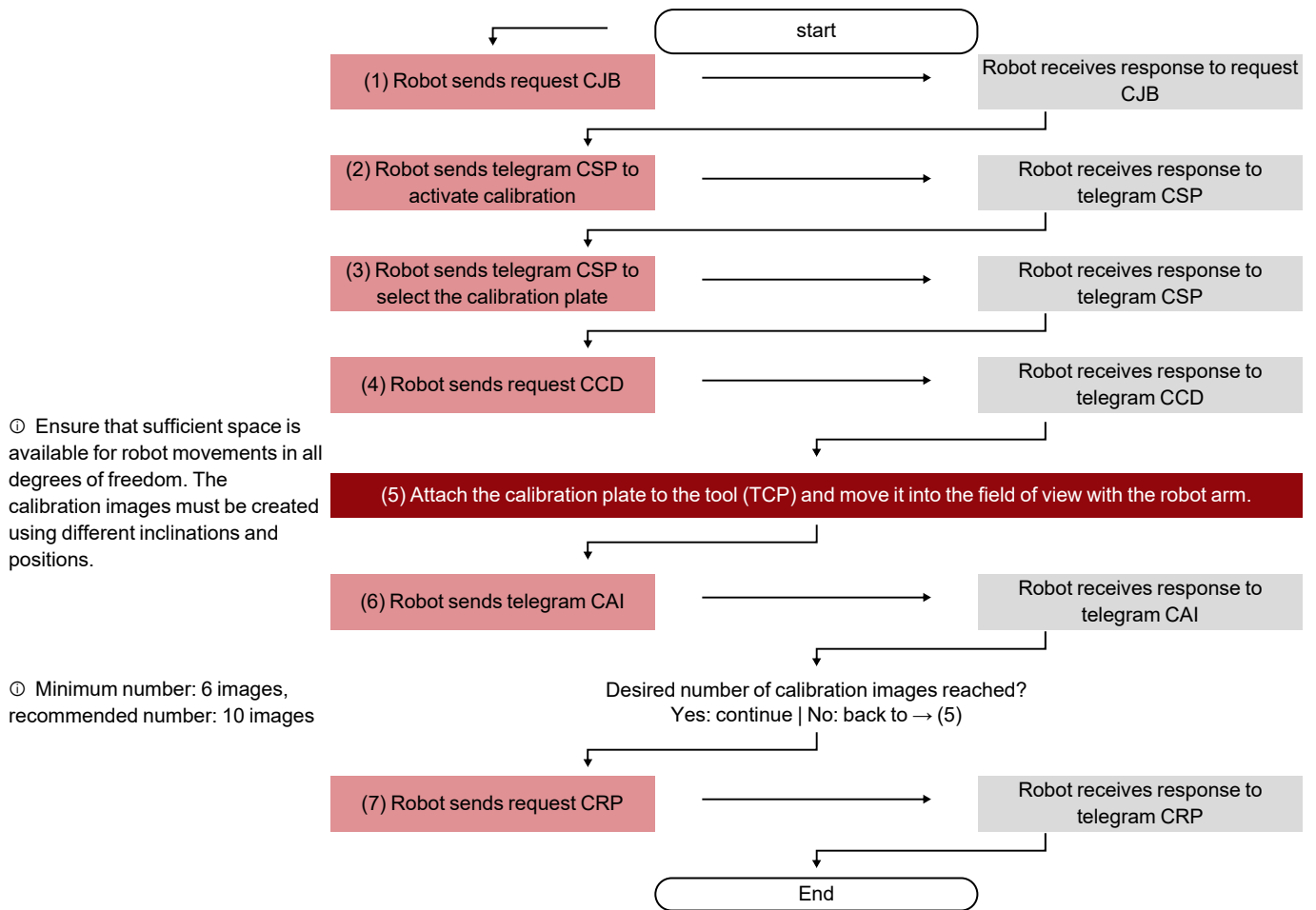


### 7.3.5.3 Automated calibration: Hand-Eye calibration (Robotics)



Additional information: User Manual, chapter "Hand-Eye calibration (Robotics)", Notes on position selection

### 7.3.5.4 Automated calibration: Base-Eye calibration (Robotics)

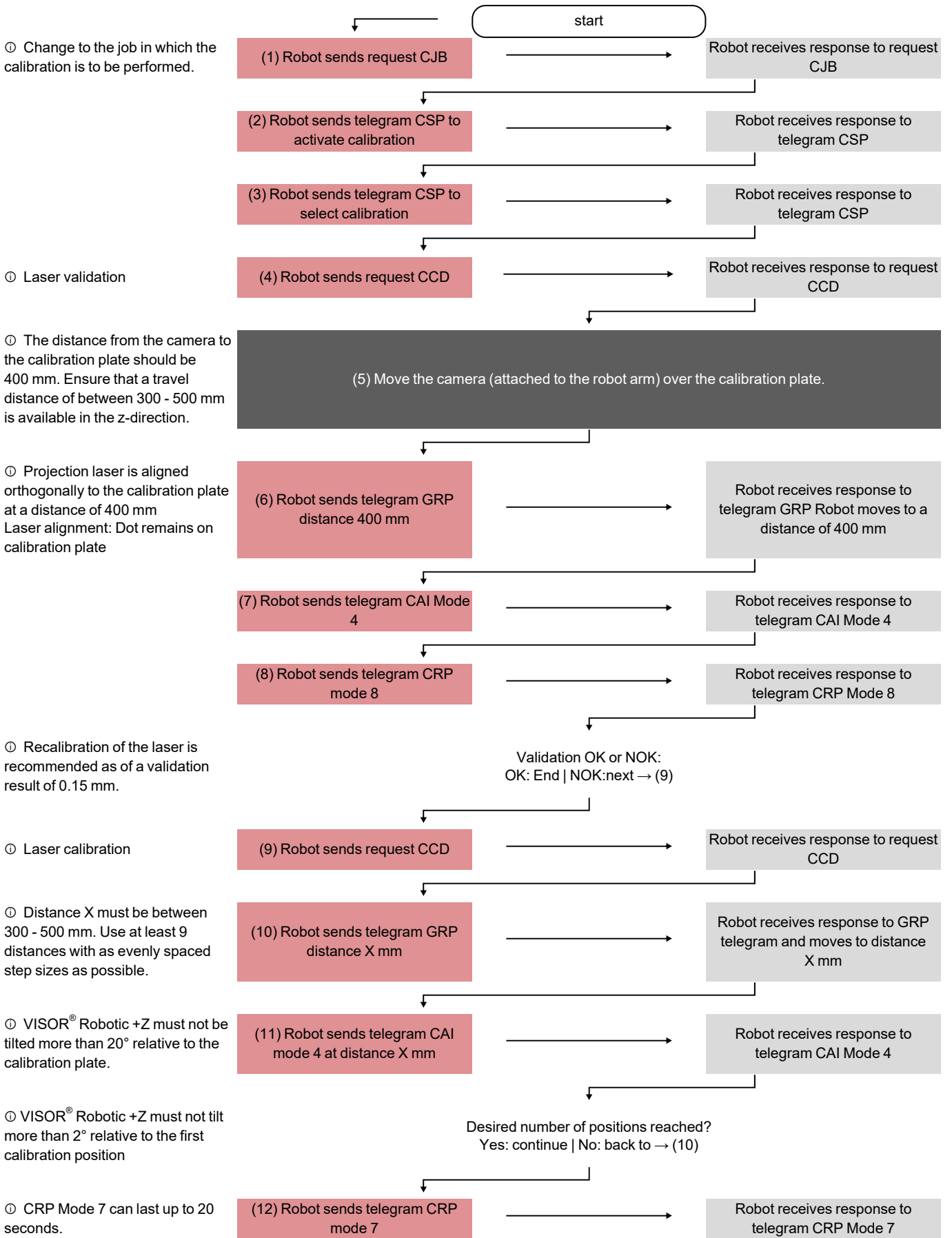


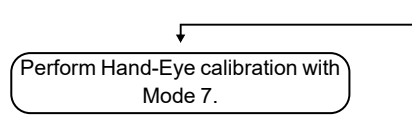
Additional information: User Manual, chapter "Base-Eye calibration (Robotics)", Notes on position selection

### 7.3.6 Calibration routine VISOR® Robotic +Z

#### Preparation of calibration routine VISOR® Robotic +Z

1. Create a standard job with a focus setting of 400 mm in SensoConfig.
2. Add a detector of your choice.
3. Start the sensor.
4. Carry out the Hand-Eye calibration at a distance of 400 mm with CRP Mode 7.





Additional information: User manual 068-14797, chapter "Calibration routine VISOR<sup>®</sup>Robotic +Z"

## 7.4 Description Telegrams ASCII

### 7.4.1 General

#### Reset statistics (ASCII)

[Availability and supported interfaces](#)

##### Overview

Reset Statistics (RST) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>R</b>	Reset statistics
2	<b>S</b>	
3	<b>T</b>	
Example:	RST	
Reset Statistics (RST) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>R</b>	Reset statistics
2	<b>S</b>	
3	<b>T</b>	
4	P F	P: Pass F: Fail
Example:	RSTP	
Additional information:		
Accepted in run mode:		Yes
Accepted in configuration mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		<a href="#">Availability and supported interfaces</a>
End of telegram:		Max. 4 bytes (optional)

**NOTE:**



Statistics values can be output in the operating step Output / tab Telegram / "Payload", selection "GENERAL".

Additional information: see data output ASCII / [General](#)

## Process image from file (ASCII)

### Availability and supported interfaces

#### Overview

Process image from file (PIF) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>P</b>	Process image from file
2	<b>I</b>	
3	<b>F</b>	
4	1	Request version
5	0	Reserved
6	1	Reserved
7-9	X	Length of the following file name of the file (actual number of bytes with file extension, max. 255 bytes)
10 ... n	X	File name (UTF-8) and format available on the device in the directory "/tmp/". Allowed extensions: Monochrome sensor: .pgm Color sensor: .ppm (RGB) or .pgm (Bayer) or .jpg (RGB)
Example:	PIF1 0 1 009 Image.pgm PIF1 0 1 008 test.pgm	
Process image from file (PIF) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>P</b>	Process image from file
2	<b>I</b>	
3	<b>F</b>	
4	P F	P: Pass F: Fail
5-7	X	<u>Error codes</u>
8	0	Reserved
9-16	X	Length of the implicit result output
17 ... n	X	Output of the implicit result
Example:	PIF P 000 00000010 0123456789 PIF P 000 0 00000014 [15;P;1;53371] PIF P 000 1 00000005 [2;7]	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	No change	
Supported interfaces:	<u>Availability and supported interfaces</u>	
End of telegram:	Max. 4 bytes (optional)	

**NOTE:**

Image size of the test image must match the image size of the currently active job on the device.

## 7.4.2 Control

### Trigger (ASCII)

#### Availability and supported interfaces

#### Overview

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

## Extended trigger (ASCII)

### Availability and supported interfaces

#### Overview

Extended Trigger (TRX) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>T</b>	Extended trigger, (trigger with index, for correlation of trigger to corresponding result data, via port 2006)
2	<b>R</b>	
3	<b>X</b>	
4 - 5	X	Length of following data (0-99)
6 ... n	X	Data
Example:	TRX06MyPart	
Extended Trigger (TRX) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>T</b>	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	<b>R</b>	
3	<b>X</b>	
4	P F	P: 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:	TRXP06MyPartR00000000	
Additional information:		
Accepted in run mode:		Yes
Accepted in configuration mode:		Yes
Accepted when Ready is low:		No
Status of Ready signal during processing:		Low
Supported interfaces:		<a href="#">Availability and supported interfaces</a>
End of telegram:		Max. 4 bytes (optional)

## Trigger Robotics (ASCII)

### Availability and supported interfaces

#### Overview

Trigger Robotics (TRR) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>T</b>	Trigger Robotics
2	<b>R</b>	
3	<b>R</b>	
4	1	Request version
5-6	X	Length of trigger identifier
7 ... n	X	Trigger Identifier
n+1...n+8	X	Pose_TCP Pos. X (in user unit * 1000)
n+9...n+16	X	Pose_TCP Pos. Y (in user unit * 1000)
n+17...n+24	X	Pose_TCP Pos. Z (in user unit * 1000)
n+25...n+32	X	Pose_TCP Angle X (in degrees * 1000)
n+33...n+40	X	Pose_TCP Angle Y (in degrees * 1000)
n+41...n+48	X	Pose_TCP Angle Z (in degrees * 1000)
Example:	TRR104Part000040040000500500006006000070070000800800009009	
Trigger Robotics (TRR) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>T</b>	Trigger (response to command trigger without index, via port 2006. If defined: Result data without index via port 2005)
2	<b>R</b>	
3	<b>R</b>	
4	P F	P: Pass F: Fail
5-7	X	<a href="#">Error codes</a>
8-9	X	Length of trigger identifier
10 ... n	X	Trigger Identifier
n+1	X	Operation Mode C = Config R = Run
n+2...n+9	X	Length of result data
n+10...m	X	Result data
Example:	TRRP00004PartR00000000	
Additional information:		
Accepted in run mode:		Yes
Accepted in configuration mode:		Yes

Accepted when Ready is low:	No
Status of Ready signal during processing:	Low
Supported interfaces:	<a href="#">Availability and supported interfaces</a>
End of telegram:	

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

## Set Trigger ID (ASCII)

[Availability and supported interfaces](#)

[Overview](#)

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

## Read robot position (ASCII)

### Availability and supported interfaces

### Overview

Read robot position (GRP) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read robot position
2	<b>R</b>	
3	<b>P</b>	
4	1	Request version
5	x	Mode 0: Calibration plate 1: best detector result
6-13	x	Distance: Distance value (in customer unit *1000)
14	x	Z-axis 0: Perpendicular to the VISOR® 1: Perpendicular to the laser (at VISOR® Robotic +Z)
15	x	Z-rotation 0: not aligned 1: aligned to the object
16-23	x	Pose_TCP Pos. X (in customer unit *1000)
24-31	x	Pose_TCP Pos. Y (in customer unit *1000)
32-39	x	Pose_TCP Pos. Z (in customer unit *1000)
40-47	x	Pose_TCP Angle X (in degrees *1000)
48-55	x	Pose_TCP Angle Y (in degrees *1000)
56-63	x	Pose_TCP Angle Z (in degrees *1000)
64-66	x	Detector number (only for Mode 1)
Example:	GRP11000010012000010010000200200003003000040040000500500006006	
Read robot position (GRP) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read robot position
2	<b>R</b>	
3	<b>P</b>	
4	P F	P: Pass F: Fail
5-7	x	<u>Error codes</u>
8-15	x	Pose_TCP Pos. X (in customer unit *1000)
16-23	x	Pose_TCP Pos. Y (in customer unit *1000)
24-31	x	Pose_TCP Pos. Z (in customer unit *1000)
32-39	x	Pose_TCP Angle X (in degrees *1000)
40-47	x	Pose_TCP Angle Y (in degrees *1000)
48-55	x	Pose_TCP Angle Z (in degrees *1000)

56-59	x	End of the string
Example:	GRPP00000001001000020020000300300004004000500500006006	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	No change	
Supported interfaces:	<a href="#">Availability and supported interfaces</a>	
End of telegram:	Max. 4 bytes (optional)	

## Job change (ASCII)

### Availability and supported interfaces

#### Overview

Job change (CJB) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Job change
2	<b>J</b>	
3	<b>B</b>	
4 - 6	X	Job number
Example:	CJB005	
Job change (CJB) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Job change
2	<b>J</b>	
3	<b>B</b>	
4	P F	P: Pass F: Fail
5	T F	Triggered Freerun
6 - 8	X	Job number
Example 1:	CJBPT005	
Additional information:		
Accepted in run mode:		Yes
Accepted in configuration mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		Low
Supported interfaces:		<a href="#">Availability and supported interfaces</a>
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)

### Availability and supported interfaces

#### Overview

Job Change Permanent (CJP) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Job change permanent (Change Job Permanently)
2	<b>J</b>	
3	<b>P</b>	
4 - 6	X	Job number
Example:	CJP005	
Job Change Permanent (CJP) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Job change permanent (Change Job Permanently)
2	<b>J</b>	
3	<b>P</b>	
4	P F	P: Pass F: Fail
5	T F	Triggered Freerun
6 - 8	X	Job number
Example 1:	CJPPT005	
Additional information:		
Accepted in run mode:		Yes
Accepted in configuration mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		Low
Supported interfaces:		<a href="#">Availability and supported interfaces</a>
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)

### Availability and supported interfaces

#### Overview

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

### 7.4.3 Job settings

#### Make Job Template (ASCII)

[Availability and supported interfaces](#)


##### Overview


Make Job Template (MJT) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>M</b>	Make Job Template
2	<b>J</b>	
3	<b>T</b>	
4	1	Request version
Example:	MJT1	
Make Job Template (MJT) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>M</b>	Make Job Template
2	<b>J</b>	
3	<b>T</b>	
4	P F	P: Pass F: Fail
5 - 7	X	<a href="#">Error codes</a>
Example:	MJTP000	
Additional information:		
Accepted in run mode:		Yes
Accepted in configuration mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		<a href="#">Availability and supported interfaces</a>
End of telegram:		Max. 4 bytes (optional)

## Auto working distance (ASCII)

### Availability and supported interfaces

#### Overview

Auto working distance (AFC) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>A</b>	Auto Working Distance (Auto Focus)
2	<b>F</b>	
3	<b>C</b>	
4	1	Request version
5	X	0: Temporary 1: Permanent
6	X	Step size 1-5
7 - 9	X	Focus selection 0: Highest score 1: Min. working distance 2: Max. working distance 3: Average working distance 4: Median working distance 5: Maximum score and all planes
10	X	Focus unit 0: millimeter 1: Steps
11	X	Working distance selection 0: Default range 1: Specified range
	 <b>NOTE:</b> The following byte sequence is only relevant if "Distance range selection" has been set to 1.	
12 - 19	X	Start of working region (close)
20 - 27	X	End of working region (far)
Example:	Example 1: AFC11100500 Example 2: AFC111005010001000000100000	
Auto working distance (AFC) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>A</b>	Auto Working Distance (Auto Focus)
2	<b>F</b>	
3	<b>C</b>	
4	P F	P: Pass F: Fail
5 - 7	X	<a href="#">Error codes</a>
8 - 10	X	Focus selection = 5 ; Number of Focus selection distances found = 1-4 ; 1

		<b>NOTE:</b> The following fields [Distance value / Score value] are repeated for each number of distances found.
11 - 18	X	Distance value (in mm *1000) or in steps
19 - 26	X	Score value in %*1000
Example:	AFCP00000200000095000000900009300000089000	
Additional information:		
Accepted in run mode:		Yes
Accepted in configuration mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		<a href="#">Availability and supported interfaces</a>
End of telegram:		Max. 4 bytes (optional)

## Set working distance (ASCII)

### Availability and supported interfaces

### Overview

Set working distance (SFC) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>S</b>	Working distance (Set Focus)
2	<b>F</b>	
3	<b>C</b>	
4	1	Request version
5	X	0: Temporary 1: Permanent Mode 5 only possible permanently.
6	X	Movement 0: Absolute 1: Relative 2: Absolute with reinitialization 4: Absolute from the job (byte 7 unit always 0) 5: Copies the working distance from the current job to all jobs that contain the working distance specified in byte 8-15.
7	X	Unit 0: 1/1000 millimeter 4: Steps
8 - 15	X	Working distance (in mm *1000) in steps or job number For Mode 5: Working distance that the target job has before the copy (in mm *1000)
Example:	SFC11140000010	
Set working distance (SFC) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>S</b>	Working distance (Set Focus)
2	<b>F</b>	
3	<b>C</b>	
4	P F	P: Pass F: Fail
5 - 7	X	<u>Error codes</u>
8 - 15	X	Working distance (in mm *1000) or in steps
Example:	SFCP00000000050	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	No change	
Supported interfaces:	<u>Availability and supported interfaces</u>	
End of telegram:	Max. 4 bytes (optional)	

## Read working distance (ASCII)

### Availability and supported interfaces

#### Overview

Read working distance (GFC) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read working distance (Get Focus)
2	<b>F</b>	
3	<b>C</b>	
4	1	Request version
5	X	Unit 0 - 1/1000 millimeters 4 - steps
Example:	GFC10	
Read working distance (GFC) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read working distance (Get Focus)
2	<b>F</b>	
3	<b>C</b>	
4	P F	P: Pass F: Fail
5 - 7	X	<u>Error codes</u>
8 - 15	X	Distance value (in mm *1000) or in steps
Example:	GFPC00000092500	
Additional information:		
Accepted in run mode:		Yes
Accepted in configuration mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		<u>Availability and supported interfaces</u>
End of telegram:		Max. 4 bytes (optional)

## Auto shutter speed (ASCII)

### Availability and supported interfaces

#### Overview

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

## Set shutter speed (ASCII)

### Availability and supported interfaces

#### Overview

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

## Read shutter speed value (ASCII)

[Availability and supported interfaces](#)

[Overview](#)

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

## Set gain (ASCII)

### Availability and supported interfaces

#### Overview

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

## Read gain value (ASCII)

### Availability and supported interfaces

#### Overview

Read gain value (GGA) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read gain value (Get Gain)
2	<b>G</b>	
3	<b>A</b>	
Example:	GGA	
Read gain value (GGA) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read gain value (Get Gain)
2	<b>G</b>	
3	<b>A</b>	
4	P F	P: Pass F: Fail
5 - 9	X	Current gain value (value *1000), e.g. 1.000 = 01000
Example:	GGAP01000	
Additional information:		
Accepted in run mode:		Yes
Accepted in configuration mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		<a href="#">Availability and supported interfaces</a>
End of telegram:		Max. 4 bytes (optional)

## Set parameters (ASCII)

### Availability and supported interfaces

### Overview

Set parameters (SPP/SPT) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>S</b>	Set parameters
2	<b>P</b>	
3	<b>P</b> <b>T</b>	P: Permanent T: Temporary
4 - 6	X	Detector number
7 - 9	X	Command: Parameter number, see <a href="#">Overview set / read detector parameters (ASCII)</a>
10 - 14	X	Length of value (max. 512 bytes)
15 ... n	X	Value
Example:	SPP0010010000560000	
Set parameters (SPP/SPT) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>S</b>	Set parameters
2	<b>P</b>	
3	<b>P</b> <b>T</b>	P: Permanent T: Temporary
4	P F	P: Pass F: Fail
5 - 8	X	SI08 - Signed Integer 08 UI08 - Unsigned Integer 08 SI16 - Signed Integer 16 UI16 - Unsigned Integer 16 SI32 - Signed Integer 32 UI32 - Unsigned Integer 32 SI40 - Signed Integer 40 UI40 - Unsigned Integer 40 FLOT - Float DOBL - Double STRG - String BOOL - Boolean SP08 - Special Signed 8 UDEF - Undefined IARR - Integer Array ZERO - Default Zero Parameter
Example:	SPPPSTRG	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	Low	

Supported interfaces:	<a href="#">Availability and supported interfaces</a>
End of telegram:	Max. 4 bytes (optional)

## Read parameter (ASCII)

### Availability and supported interfaces

### Overview

Read parameter (GPA) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read parameter (Get Parameter)
2	<b>P</b>	
3	<b>A</b>	
4 - 6	X	Detector number e.g. 001
7 - 9	X	Command: Parameter number, see <a href="#">Overview set / read detector parameters (ASCII)</a>
Example:	GPA001001	
Read parameter (GPA) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read parameter (Get Parameter)
2	<b>P</b>	
3	<b>A</b>	
4	P F	P: Pass F: Fail
5 - 8	X	SI08 - Signed Integer 08 UI08 - Unsigned Integer 08 SI16 - Signed Integer 16 UI16 - Unsigned Integer 16 SI32 - Signed Integer 32 UI32 - Unsigned Integer 32 SI40 - Signed Integer 40 UI40 - Unsigned Integer 40 FLOT - Float DOBL - Double STRG - String BOOL - Boolean SP08 - Special Signed 8 UDEF - Undefined IARR - Integer Array ZERO - Default Zero Parameter
9 - 13	X	Length of value (n) e.g. 00005
14 ... n	X	Value
Example:	GPAPSTRG00005Test1	
Additional information:		
Accepted in run mode:		Yes
Accepted in configuration mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change

Supported interfaces:	<a href="#"><u>Availability and supported interfaces</u></a>
End of telegram:	Max. 4 bytes (optional)

**Overview set / read detector parameters (ASCII)**

Detector	Function	Value	Multiplier	Length
<b>Alignment</b>				
Pattern matching	Threshold value Min.	1	1000	n
Contour comparison	Threshold value Max.	2	1000	n
	Result offset 0: "Off" 1: "Image plane (in pixels)" 2: "Align (2D)" 3: "Robot (3D)"	30	1	n
	Result offset Image plane: Pos. X	31	1000	n
	Result offset Image plane: Pos. Y	32	1000	n
	Result offset Image plane: Angle	33	1000	n
	Result offset Align (2D), Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z	34	1000	48 (6 * 8 bytes per value)
	Calculate Result offset* with transmitted position  <ul style="list-style-type: none"> <li>• Align (2D): Pos. X, Pos. Y, 0, 0, 0, Angle Z</li> <li>• Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z</li> </ul> *A valid position for the detector must be available	35	1000	48 (6 * 8 bytes per value)
Edge detection	Probe 1: Transition 0: Any 1: Dark to light 2: Light to dark	101	1	n
	Probe 2: Transition 0: Any 1: Dark to light 2: Light to dark	102	1	n
	Probe 3: Transition 0: Any 1: Dark to light 2: Light to dark	103	1	n
	Probe 1: Threshold value Min.	104	1000	n
	Probe 2: Threshold value Min.	105	1000	n
	Probe 3: Threshold value Min.	106	1000	n

Detector	Function	Value	Multiplier	Length	
Contour check	Calculates the missing contours	101	1	n	
	Calculates the additional contour	102	1	n	
	Calculates the roughness contours	103	1	n	
	Maximum number of invalid contours	104	1	n	
	Pixel tolerance range	105	1	n	
	Minimum length of the defective contours	106	1	n	
	Minimum roughness of the defective contours	107	1	n	
	Maximum roughness of the defective contours	108	1	n	
	Automatic contrast	109	1	n	
	Contrast value	110	1	n	
	Minimum contour length	111	1	n	
	Automatic alignment	112	1	n	
	<b>Detectors</b>				
Pattern matching Contour Contour 3D Shape find	Threshold value Min.	1	1000	n	
	Threshold value Max.	2	1000	n	
	Result offset 0: "Off" 1: "Image plane (in pixels)" 2: "Align (2D)" 3: "Robot (3D)"	30	1	n	
	Result offset Image plane: Pos. X	31	1000	n	
	Result offset Image plane: Pos. Y	32	1000	n	
	Result offset Image plane: Angle	33	1000	n	
	Result offset <ul style="list-style-type: none"> <li>Align (2D): Pos. X, Pos. Y, 0, 0, 0, Angle Z</li> <li>Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z</li> </ul>	34	1000	48 (6 * 8 bytes per value)	
	Calculate Result offset* with transmitted position <ul style="list-style-type: none"> <li>Align (2D): Pos. X, Pos. Y, 0, 0, 0, Angle Z</li> <li>Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z</li> </ul> <p>*A valid position for the detector must be available</p>	35	1000	48 (6 * 8 bytes per value)	
	Result offset <ul style="list-style-type: none"> <li>Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z</li> </ul>	34	1000	48 (6 * 8 bytes per value)	
	Target Mark 3D	Result offset <ul style="list-style-type: none"> <li>Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z</li> </ul>	34	1000	48 (6 * 8 bytes per value)

Detector	Function	Value	Multiplier	Length
	Calculate Result offset* with transmitted position <ul style="list-style-type: none"> <li>Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z</li> </ul> *A valid position for the detector must be available	35	1000	48 (6 * 8 bytes per value)
	ID of the active Target Mark	101	-	n
	Target Mark name	102	-	n
	Add current Target Mark to the list of Target Marks (parameter can only be set!) A Target Mark must have been detected.	103	-	n
Shape find	Geometry 0: "Circle" 1: "Square" 2: "Rectangle"	101	1	n
	Edge transition 0: "Dark - bright" 1: "Bright - dark"	102	1	n
	Diameter Length Height	103	1000	n
	Width	104	1000	n
	Tolerance	105	1	n
	Min. angle "Circle": [0°:0°] "Square": [-45°:45°] "Rectangle": [-90°:90°]	106	1	n
	Max. Angle "Circle": [0°:0°] "Square": [-45°:45°] "Rectangle": [-90°:90°]	107	1	n
	Min. contrast	108	1	n

Detector	Function	Value	Multiplier	Length
Distance	Relationship of calibration plate coordinate system to projector coordinate system***	111	1000, 1000, 1000, 1000, 1000, 1000	48 (6 * 8 bytes per value)
	Relationship of robot coordinate system to projector coordinate system***	112	1000, 1000, 1000, 1000, 1000, 1000	48 (6 * 8 bytes per value)
	Reference Camera- to Projector coordinate system***	113	1000, 1000, 1000, 1000, 1000, 1000	48 (6 * 8 bytes per value)
	Reference TCP- to Projector coordinate system***	114	1000, 1000, 1000, 1000, 1000, 1000	48 (6 * 8 bytes per value)
	Relationship of calibration plate coordinate system to projector coordinate system (pos. X, Y, Z)***	121	1000, 1000, 1000	24 (3 * 8 bytes per value)
	Relationship of calibration plate coordinate system to projector coordinate system (angle X, Y, Z)***	122	1000, 1000, 1000	24 (3 * 8 bytes per value)
	Laser offset Z in mm	123	1000	8
Grayscale	Threshold value Min.	1	1000	n
	Threshold value Max.	2	1000	n
	Grayscale value Min.	101	1000	n
	Grayscale value Max.	102	1000	n
	Invert grayscale value	103	1	n
Contrast Brightness	Threshold value Min.	1	1000	n
	Threshold value Max.	2	1000	n
Caliper	Threshold value Distance Min.	101	1000	n
	Threshold value Distance Max.	102	1000	n
	Invert distance threshold value 0: not inverted 1: inverted	103	1	1
	Distance mode 0: Minimum 1: Maximum 2: Average 3: Median 4: Smallest opposite 5: Largest opposite 6: Minimum/Maximum by search stripe	104	1	n
	Probe 1: Threshold value Min.	105	1000	n
	Probe 2: Threshold value Min.	106	1000	n
	Probe 1: Smoothing	107	1000	n
	Probe 2: Smoothing	108	1000	n
	Probe 1: Transition 0: Any 1: Dark to light 2: Light to dark	109	1	n

Detector	Function	Value	Multiplier	Length
BLOB	Probe 2: Transition 0: Any 1: Dark to light 2: Light to dark	110	1	n
	Probe 1: Number of search stripes	111	1	n
	Probe 2: Number of search stripes	112	1	n
	Grayscale value Min.	101	1000	n
	Grayscale value Max.	102	1000	n
	Invert grayscale value 0: not inverted 1: inverted	103	1	1
	Threshold value Number of BLOBs Min.	120	1	n
	Threshold value Number of BLOBs Max.	121	1	n
	Invert number threshold value 0: not inverted 1: inverted	122	1	1
	Number of set features (read only)	123	1	n
	Selection of a feature from the list	124	1	n
	Feature threshold value Min.*	125	1000	n
	Feature threshold value Max.*	126	1000	n
	Invert feature threshold value*	127	1	1
	*The feature to be changed must first be selected before it can be changed. (124)			
Barcode Data code OCR	Reference string	101	-	n (length of string)
Color value Color area	Color space (read only)	21	1 = RGB 2 = HSV 3 = LAB	3
	Channel selection (read only)	22	Bit field one digit per color channel	4
	Color channel 1: Threshold value Min.	101	1000	n
	Color channel 1: Threshold value Max.	102	1000	n
	Color channel 1: Invert threshold value	103	1	n
	Color channel 2: Threshold value Min.	104	1000	n
	Color channel 2: Threshold value Max.	105	1000	n
	Color channel 2: Invert threshold value	106	1	n
	Color channel 3: Threshold value Min.	107	1000	n
	Color channel 3: Threshold value Max.	108	1000	n
Color list	Color space (read only)	21	1 = RGB 2 = HSV 3 = LAB	3

Detector	Function	Value	Multiplier	Length
	Channel selection (read only)	22	Bit field one digit per color channel	4
	Color distance threshold value	101	1000	n
	Set color distance threshold value active	102	1	n
	Number of colors in list	103	1	n
	Selection of a color from the list	104	1	n
	Color value of the selected color (color channel 1, color channel 2, color channel 3, color channel 4 [constantly 0])*	105	1000	32
	*The feature to be changed must first be selected before it can be changed. (104)			
Busbar	Threshold value Min.	1	1000	n
Wafer	Threshold value Max.	2	1000	n
Result processing	Selection of expression name	122	-	n (length of string)
	Current expression*	124	-	n (length of string)
	Selection of the static variable	142	-	n (length of string)
	Set initial value for static variable**	144	-	n (length of string)
	Set current value of static variable as initial value**	145	-	n (length of string)
	Reset static variable to initial value	146	-	n (length of string)
	*The expression to be changed must first be selected before it can be changed. (122)			
	**Before accessing a static variable, the static variable must first be selected. (142)			
	*** All Parameter are read-only.			

## Set region of interest (ROI) (ASCII)

### Availability and supported interfaces

### Overview

Set ROI (SRP/SRT) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>S</b>	Set region of interest (Set ROI)
2	<b>R</b>	
3	<b>P</b> <b>T</b>	P = Permanent T = Temporary
4 - 11	X	ROI Info Length in bytes, from byte 5 to end 39 byte: Circle 55 byte: rectangle, ellipse, free shape
12 - 14	X	Detector number e.g. 001
15 - 16	X	ROI Index 00: for yellow region of interest 01: for red teach range 02: Position control
17 - 18	X	ROI shape 01: Circle 02: Rectangle 03: Ellipse 04: Free shape
19 - 26	X	Center X (value in pixels * 1000), e.g. 160 pixels = 00160000
27 - 34	X	Center Y (value in pixels * 1000), e.g. 120 pixels = 00120000
35 - 42	X	Half width / X-radius (value in pixels * 1000), e.g. 80 pixels = 00080000
43 - 50	X	Half height (not for circle) (value in pixels * 1000), e.g. 40 pixels = 00040000
51 - 58	X	Angle (not for circle) (value in ° * 1000), e.g. 180° = 00180000
Example:	SRP0000005500100020016000000120000000800000004000000180000 Length=55, detector=1, yellow region of interest, rectangle, center X=160, center Y=120, half width= 80, half height=40, orientation=180	
Set ROI (SRP/SRT) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>S</b>	Set region of interest (Set ROI)
2	<b>R</b>	
3	<b>P</b> <b>T</b>	Permanent Temporary
4	P F	P: Pass F: Fail
Example:	SRPP	

Additional information:	
Accepted in run mode:	Yes
Accepted in configuration mode:	No
Accepted when Ready is low:	Yes
Status of Ready signal during processing:	Low
Supported interfaces:	<a href="#">Availability and supported interfaces</a>
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 region of interest (ASCII)

### Availability and supported interfaces

### Overview

Read region of interest (GRI) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read region of interest (Get ROI)
2	<b>R</b>	
3	<b>I</b>	
4 - 6	X	Detector number e.g. 001
7 - 8	X	ROI Index 00: for yellow region of interest 01: for red teach range 02: Position control
Example:	GRI00100	
Read region of interest (GRI) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read region of interest (Get ROI)
2	<b>R</b>	
3	<b>I</b>	
4	P F	P: Pass F: Fail
5 - 12	X	ROI Info Length in bytes, from byte 5 to end 39 byte: Circle 55 byte: rectangle, ellipse, free shape
13 - 15	X	Detector number e.g. 001
16 - 17	X	ROI Index 00: for yellow region of interest 01: for red teach range 02: Position control
18 - 19	X	ROI shape 01: Circle 02: Rectangle 03: Ellipse 04: Free shape
20 - 27	X	Center X (value in pixels * 1000)
28 - 35	X	Center Y (value in pixels * 1000)
36 - 43	X	Half width / X-radius (value in pixels * 1000)
44 - 51	X	Half height (not for circle) (value in pixels * 1000), e.g. 40 pixels = 00040000
52 - 59	X	Angle (not for circle) (value in ° * 1000), e.g. 180° = 00180000

Example:	GRIP0000005500100020016000000120000000800000004000000090000 (Length= 55, detector 1, region of interest, rectangle, center X= 160, center Y= 120, half width= 80, half height= 40, angle= 90)
Additional information:	
Accepted in run mode:	Yes
Accepted in configuration mode:	No
Accepted when Ready is low:	Yes
Status of Ready signal during processing:	Low
Supported interfaces:	<a href="#">Availability and supported interfaces</a>
End of telegram:	Max. 4 bytes (optional)

## Set ROI content (ASCII)

### Availability and supported interfaces

### Overview

Set ROI content (SRC) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>S</b>	Set ROI content
2	<b>R</b>	
3	<b>C</b>	
4	1	Request version
5	X	0: Temporary 1: Permanent
6-8	X	Detector number 0: Alignment >0: Default detector in job
9-10	00	Reserved
11	X	0: Do not use mask file 1: Use mask file
12-16	00000	Reserved
17-19	X	Length of the following file name of the file (actual number of bytes with file extension, max. 255 bytes) Or for 000: Default name mask.pgm
20 ... n	X	File name (UTF-8) and format available on the device in the directory "/tmp". Default name mask.pgm File format: PGM
Example:	SRC1100100100000008Test.pgm (for file "Test.pgm") SRC1100100100000000 (for file: "mask.pgm")	
Set ROI content (SRC) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>S</b>	Set ROI content
2	<b>R</b>	
3	<b>C</b>	
4	P F	P: Pass F: Fail
5-7	X	<u>Error codes</u>
Example:	SRCP000	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	No change	
Supported interfaces:	<u>Availability and supported interfaces</u>	
End of telegram:	Max. 4 bytes (optional)	

## Read ROI content (ASCII)

### Availability and supported interfaces

### Overview


Read ROI content (GRC) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read ROI content (Get ROI content)
2	<b>R</b>	
3	<b>C</b>	
4	1	Request version
5-7	X	Detector number 0: Alignment >0: Default detector in job
8-9	00	Reserved
10	X	0: Do not create mask file 1: Create mask file (if mask is available in job)
11	X	0: Do not create pattern file 1: Create pattern file
12	X	0: Do not create contour file 1: Create contour file
13-15	000	Reserved
16-18	X	Length of the following file name of the file (actual number of bytes with file extension, max. 255 bytes) Or for 000: Default name mask.pgm / pattern.pgm / contour.pgm
19 ... n	X	File name (UTF-8) and format available on the device in the directory "/tmp". Default name mask.pgm / pattern.pgm / contour.pgm File format: PGM
Example:	GRC10010010000008Test.pgm (file with standard extension: Testmask.pgm) GRC100100111000008Test.pgm (files with standard extension: Testmask.pgm, Testpattern.pgm & Testcontour.pgm) GRC100100111000000 (file only with standard extension: mask.pgm, pattern.pgm & contour.pgm)	
Read ROI content (GRC) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read ROI content (Get ROI content)
2	<b>R</b>	
3	<b>C</b>	
4	P F	P: Pass F: Fail
5-7	X	<u>Error codes</u>
8-12	X	Width of the search region
13-17	X	Height of the search region
Example:	GRCP0000108001440	

Additional information:	
Accepted in run mode:	Yes
Accepted in configuration mode:	No
Accepted when Ready is low:	Yes
Status of Ready signal during processing:	No change
Supported interfaces:	<a href="#">Availability and supported interfaces</a>
End of telegram:	Max. 4 bytes (optional)

## Read job list (ASCII)

### Availability and supported interfaces

### Overview


Read job list (GJL) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read job list (Get Job List)
2	<b>J</b>	
3	<b>L</b>	
Example:	GJL	
Read job list (GJL) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read job list (Get Job List)
2	<b>J</b>	
3	<b>L</b>	
4	P F	P: Pass F: Fail
5 - 7	X	Response version
8 - 10	X	Number of jobs
11 - 13	X	Active job number
	<b>NOTE:</b> The following byte sequence is repeated for each job from 1 to "Number of jobs". The byte numbers shift accordingly.	
14 - 16	X	Number of characters for the job name. This can be used to specify a unique name for job n.
17 ... n	X	From this position, the name for job n follows in the specified length.
n+1 ... n + 3	X	Number of subsequent bytes. A description for job n can be specified.
n + 4 ... m	X	From this position, the description for job n follows in the specified length.
m + 1 ... m + 3	X	Number of subsequent bytes. This can be used to specify a unique name for the author of job n.
m + 4 ... k	X	From this position, the name for the author of job n follows in the specified length.
k + 1 ... k + 8	X	Date of creation of Job n (8 bytes)
k + 20 ... k + 16	X	Date of last modification of job n (8 bytes)
Example:	GJLP001001001007testjob010Default.Job004Test2014112720141128	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	No change	

Supported interfaces:	<a href="#"><u>Availability and supported interfaces</u></a>
End of telegram:	Max. 4 bytes (optional)

## Read detector list (ASCII)

### Availability and supported interfaces

#### Overview

Read detector list (GDL) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Get Detector List
2	<b>D</b>	
3	<b>L</b>	
Example:	GDL	
Read detector list (GDL) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Get Detector List
2	<b>D</b>	
3	<b>L</b>	
4	P F	P: Pass F: Fail
5 - 7	X	Job number of the current job
8 - 10	X	Number of detectors in the current job
	 <b>NOTE:</b> The following byte sequence is repeated for each detector in the job. The byte numbers shift accordingly.	
11 - 13	X	Number of subsequent bytes. This allows a unique name for the detector n to be specified.
14 ... n	X	From this position, the name for detector n follows, in the given length.
n + 1 ... n+ 5	X	001 - Pattern matching 004 - Contour 005 - 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 024 - Contour 3D 027 - Result processing 028 - Target Mark 3D
Example:	GDLP001001012testdetector00005	
Additional information:		
Accepted in run mode:	Yes	

Accepted in configuration mode:	No
Accepted when Ready is low:	Yes
Status of Ready signal during processing:	No change
Supported interfaces:	<a href="#">Availability and supported interfaces</a>
End of telegram:	Max. 4 bytes (optional)

## Teach detector (ASCII)

### Availability and supported interfaces

#### Overview

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

## Set trigger delay (ASCII)

### Availability and supported interfaces

### Overview

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

## Read trigger delay (ASCII)

### Availability and supported interfaces

#### Overview

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

## Save job permanently (ASCII)

### Availability and supported interfaces

#### Overview

Save Job Permanently (SJP) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>S</b>	Save Job Permanently (Store Job Permanently)
2	<b>J</b>	
3	<b>P</b>	
4	1	Request version
Example:	SJP1	
Save Job Permanently (SJP) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>S</b>	Save Job Permanently (Store Job Permanently)
2	<b>J</b>	
3	<b>P</b>	
4	P F	P: Pass F: Fail
5 - 7	X	<u>Error codes</u>
Example:	SJPP000	
Additional information:		
Accepted in run mode:		Yes
Accepted in configuration mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		Low
Supported interfaces:		<u>Availability and supported interfaces</u>
End of telegram:		Max. 4 bytes (optional)

## Set image acquisition parameters (ASCII)

### Availability and supported interfaces

### Overview

Set image acquisition parameters (SAP) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>S</b>	Set image acquisition parameters
2	<b>A</b>	
3	<b>P</b>	
4	1	Request version
5	X	0: Temporary 1: Permanent
6 - 8	X	Command: Number of the image acquisition parameter, see <a href="#">Overview Set / read image acquisition parameters (ASCII)</a>
9 - 16	X	Length of the specification for parameters (max. total length 255 bytes).
17 ... n		Specification for parameter 1 (depending on image acquisition parameter)
Example:	SAP11009000000011 This example selects the triggered mode.  SAP110100000002010900002500000001000 This example enableds the automatic brightness control with a target value of 90 (maximum shutter speed: 25 ms, maximum gain 1).	
Set image acquisition parameters (SAP) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>S</b>	Set image acquisition parameters
2	<b>A</b>	
3	<b>P</b>	
4	P F	P: Pass F: Fail
5 - 8	X	<a href="#">Error codes</a>
Example:	SAPP000	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	No change	
Supported interfaces:	<a href="#">Availability and supported interfaces</a>	
End of telegram:	Max. 4 bytes (optional)	

## Read image acquisition parameters (ASCII)

### Availability and supported interfaces

### Overview

Read image acquisition parameters (GAP) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read image acquisition parameters
2	<b>A</b>	
3	<b>P</b>	
4	1	Request version
5 - 7	X	Number of the image acquisition parameter, see <a href="#">Overview Set / read image acquisition parameters (ASCII)</a>
8 ... n		Specification for parameter 1 (depending on image acquisition parameter)
n ... m		Specification for parameter n
Example:	GAP1009	
Read image acquisition parameters (GAP) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read image acquisition parameters
2	<b>A</b>	
3	<b>P</b>	
4	P F	P: Pass F: Fail
5 - 7	X	<a href="#">Error codes</a>
8 - 10	X	Number of the image acquisition parameter, see <a href="#">Overview Set / read image acquisition parameters (ASCII)</a>
11 - 18	X	Length of the following data
20 ... n	X	Value for parameter 1 Values after this parameter exist only if multiple values are returned (depending on image acquisition parameter).
Example:	GAPP000009000000011	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	No change	
Supported interfaces:	<a href="#">Availability and supported interfaces</a>	
End of telegram:	Max. 4 bytes (optional)	

**Overview Set / read image acquisition parameters (ASCII)**

Parameter description	Parameter number	Data type	Multiplier	Byte count	Description	Value range
Shutter speed	001	Unsigned Int	1000	8 bytes	Shutter speed in [ms] * 1000	0.026 ... 100 ms
Gain	002	Unsigned Int	1000	8 bytes	Gain Value * 1000	*Depending on image chip
Working distance	003	Unsigned Int	1000	8 bytes	Byte 1: Movement 0: Absolute 1: Relative 2: Absolute with reinitialization Byte 2: Unit 0: Millimeter [mm] 4: Steps Bytes 3-10: Value in unit * 1000	0 ... 100000 mm
Internal illumination	004	Unsigned Char	1	1 byte	0: Off 1: On	0, 1
Internal illumination - Selection of Quadrants	005	Unsigned Char	1		Byte 1: Q1 Byte 2: Q2 Byte 3: Q3 Byte 4: Q4 Per byte 0: Off 1: On	0, 1
External illumination	006	Unsigned Char	1		0: Off 1: On 2: Permanent	0, 1, 2
External illumination - pins	007	Unsigned Char	1		Byte 1: Output RD 09 Byte 2: Output YE 06 Byte 3: Output BK 07 Byte 4: Output GY 08 Per byte 0: Off 1: On	0, 1
Target laser	008	Unsigned Char	1		0: Off 1: At image acquisition 2: Between image acquisition	0, 1, 2
Trigger mode	009	Unsigned Char	1		0: Trigger 1: Free run	0, 1

Parameter description	Parameter number	Data type	Multiplier	Byte count	Description	Value range
Automatic brightness control	010	Unsigned Char	1	1 byte	Byte 1: active 0: Off 1: On	0, 1
		Unsigned Char	1	3 bytes	Bytes 2-4: Target brightness	0-255
		Unsigned Int	1000	8 bytes	Byte 5-12: max. shutter speed value * 1000	0-100
		Unsigned Int	1000	8 bytes	Byte 13-20: max. gain value * 1000	0-32 (*depending on image chip)

## Data output - Set parameter (ASCII)

### Availability and supported interfaces

#### Overview

Data output - Set parameter (STP) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>S</b>	Data output: Set parameter
2	<b>T</b>	
3	<b>P</b>	
4	1	Request version
5	X	0: Temporary 1: Permanent
6 - 8	X	<u>Parameter number, see parameter for STP and GTP telegrams (ASCII)</u>
9 - 16	X	Length of value
17 ... n	X	<u>Value for selected parameter, see parameter for STP and GTP telegrams (ASCII)</u>
Example:	STP1100100000005VISOR	
Data output - Set parameter (STP) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>S</b>	Data output: Set parameter
2	<b>T</b>	
3	<b>P</b>	
4	P F	P: Pass F: Fail
5 - 7	X	<u>Error codes</u>
Example:	STPP000	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	No change	
Supported interfaces:	<u>Availability and supported interfaces</u>	

## Data output - Read parameter (ASCII)

### Availability and supported interfaces

### Overview

Data output - Read parameters (GTP) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Data output: Get parameter
2	<b>T</b>	
3	<b>P</b>	
4	1	Request version
5 - 7	X	<u>Parameter number, see parameter for STP and GTP telegrams (ASCII)</u>
8-10	0	all rows
<u>Applies as of parameter 006</u>	>0	specific rows
Example:	GTP1001	
Data output - Read parameters (GTP) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Data output: Get parameter
2	<b>T</b>	
3	<b>P</b>	
4	P F	P: Pass F: Fail
5 - 7	X	<u>Error codes</u>
8 - 10	X	<u>Parameter number, see parameter for STP and GTP telegrams (ASCII)</u>
11 - 18	X	Length of the following data
19 ... n	X	Parameter values, depending on the selected parameter
Example:	GTPP00000100000008	

## Data output for Telegrams (ASCII)

Parameter description	Parameter number	Argument Specifier	Parameter value	Telegram parameters, value range
Start	001		0...8 byte(s)	String
Trailer	002		0...8 byte(s)	String
Separator	003		0... 5 byte(s)	String
End of response	004		0... 4 byte(s)	String
Number of lines, detector-specific payload	005		3 bytes	Integer total values <255
Detector-specific payload column Active	006	3 bytes 0: all rows >0: specific rows	1 byte * Number of rows	[0.1]
Detector-specific payload column Detector*	007	3 bytes 0: all rows >0: specific rows		
Detector-specific payload column Value*	008	3 bytes 0: all rows >0: specific rows		
Detector-specific payload column Min. length	009	3 bytes 0: all rows >0: specific rows	3 bytes * Number of rows	[0-255]
Detector-specific payload column No. of results	010	3 bytes 0: all rows >0: specific rows	4 bytes * Number of rows	[0-1000]
Detector-specific payload column Factor	011	3 bytes 0: all rows >0: specific rows	4 bytes * Number of rows	[1; 10; 100; 1000]

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

Example: There are 4 lines defined as output in the detector-specific payload. Set the minimum length (command 009 from table) in line 1 := 5, line 2 := 50, line 3 := 100, line 4 := 150: STP1100900000015000005050100150

## 7.4.4 Calibration

### Calibration: Initialize (ASCII)

[Availability and supported interfaces](#)

#### Overview

Initialize (CCD) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Initialize (Calibration: Clear Data)
2	<b>C</b>	
3	<b>D</b>	
Example:	CCD	
Initialize (CCD) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Initialize (Calibration: Clear Data)
2	<b>C</b>	
3	<b>D</b>	
4	P F	P: Pass F: Fail
Example:	CCDP	
Additional information:		
Accepted in run mode:		Yes
Accepted in configuration mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		<a href="#">Availability and supported interfaces</a>
End of telegram:		Max. 4 bytes (optional)

## Calibration: Add World Point (ASCII)

### Availability and supported interfaces

### Overview

Calibration: Add world point (CAW) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Calibration: Add World Point
2	<b>A</b>	
3	<b>W</b>	
4	1	Request version
5	X	1: Only fiducials 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 user unit * 1000)
19 - 26	X	World Y (in user unit * 1000)
27 - 34	0	Constant (8 bytes)
Example:	CAW100001001000000020000000000000 (World X = 100 mm; World Y = 200mm)	
Calibration: Add world point (CAW) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Calibration: Add World Point
2	<b>A</b>	
3	<b>W</b>	
4	P F	P: Pass F: Fail
5 - 7	X	<u>Error codes</u>
8 - 12	X	Current number of points
13 - 20	X	Image point X
21 - 28	X	Image point Y
Example:	CAWP000000010028800000566000 (Reference point 1; Image X = 288; Image Y = 566)	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	No change	
Supported interfaces:	<u>Availability and supported interfaces</u>	
End of telegram:	Max. 4 bytes (optional)	

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

## Calibration: Point pair list (ASCII)

### Availability and supported interfaces

#### Overview

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

## Calibration: Validate point pair list (ASCII)

### Availability and supported interfaces

#### Overview

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

## Calibration: Calibration plate (ASCII)

### Availability and supported interfaces

### Overview

Calibration: Calibration Plate (CCP) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Calibration: Calibrate by Plate
2	<b>C</b>	
3	<b>P</b>	
4	1	Request version
5	X	0: Temporary 1: Permanent
6	X	0: No fiducials are used. Origin of Measuring coordinate system identical to origin of Calibration Plate Coordinate System. 1: No fiducials are used. Measuring coordinate system is identical with Camera coordinate system. 2: Use world system, fiducials job 3: Use world system, fiducials request <b>CAW</b> .
7	X	0: Calibration (internal and external parameters) 1: Validate calibration 2: Calibration (internal parameters only) 5: Calibration Transformation Measuring coordinate system 7: Calibration of the external parameters for VISOR® Robotic +Z
Example:	CCP1110	
Calibration: Calibration Plate (CCP) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Calibration: Calibrate by Plate
2	<b>C</b>	
3	<b>P</b>	
4	P F	P: Pass F: Fail
5 - 7	X	<u>Error codes</u>
8 - 12	X	Number of currently detected calibration points
13 - 20	X	Deviation calibration, RMSE
21 - 28	X	Deviation calibration, mean
29 - 36	X	Deviation calibration, max.
37 - 44	X	Deviation calibration, min.
45 - 52	X	CPF_MF X (in user unit * 1000)
53 - 60	X	CPF_MF Y (in user unit * 1000)
61 - 68	0	CPF_MF Z (in user unit * 1000)
69 - 76	0	CPF_MF Angle X (in degrees * 1000)
77 - 84	0	CPF_MF Angle Y (in degrees * 1000)

85 - 92	X	CPF_MF Angle Z (in degrees * 1000)
93 - 100	X	Deviation fiducials, mean
101 - 108	X	Deviation fiducials, max.
109 - 116	X	Deviation fiducials, min.
Example:	CCPP00000012000010010000200200003003000040040 00050050000600600007007000080080000900900001001	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	No change	
Supported interfaces:	<a href="#">Availability and supported interfaces</a>	
End of telegram:	Max. 4 bytes (optional)	

## Calibration: Set Fiducial (ASCII)

### Availability and supported interfaces

#### Overview

Calibration: Set fiducial (CSF) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Calibration: Set Fiducial
2	<b>S</b>	
3	<b>F</b>	
4	1	Request version
5	X	0: Temporary 1: Permanent
Example:	CSF11	
Calibration: Set fiducial (CSF) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Calibration: Set Fiducial
2	<b>S</b>	
3	<b>F</b>	
4	P F	P: Pass F: Fail
5 - 7	X	<u>Error codes</u>
8 - 15	X	X value (in user unit * 1000)
16 - 23	X	Y value (in user unit * 1000)
24 - 31	X	Z value (in user unit * 1000)
32 - 39	X	Angle X value (in degrees * 1000)
40 - 47	X	Angle Y value (in degrees * 1000)
48 - 55	X	Angle Z value (in degrees * 1000)
56 - 63	X	Deviation fiducials, mean
64 - 71	X	Deviation fiducials, max.
72 - 79	X	Deviation fiducials, min.
Example:	CSFP000000010010000200200003003000040040 00050050000600600001001000020200003003	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	No change	
Supported interfaces:	<u>Availability and supported interfaces</u>	
End of telegram:	Max. 4 bytes (optional)	

## Calibration: Add image (ASCII)

### Availability and supported interfaces

### Overview

Calibration: Add image (CAI) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Calibration: Add Image
2	<b>A</b>	
3	<b>I</b>	
4	X	Version 1: no additional image aquisition 2: additional image aquisition with evaluation
5	X	Mode 1: Multi-image calibration 2: Hand-Eye calibration (Robotics) 3: Base-Eye calibration (Robotics) 4: Laser calibration VISOR® Robotic +Z
6-8	0	Append at the end of the list (3 bytes)
9	X	Define Measurement plane 0: Do not use image to define Measurement plane 1: Use image to define Measurement plane
10-11	X	"Robotics: Order of rotation" 00: Use rotation order specified in job 01: Yaw-Pitch-Roll (e.g. Stäubli) 02: Roll-Pitch-Yaw (e.g. Kuka, Fanuc, Hanwha, ABB**, UR**) ** when using the corresponding conversion function
12-19	X	Pose_TCP Pos. X (in user unit * 1000)
20-27	X	Pose_TCP Pos. Y (in user unit * 1000)
28-35	X	Pose_TCP Pos. Z (in user unit * 1000)
36-43	X	Pose_TCP Angle X (in degrees * 1000)
44-51	X	Pose_TCP Angle Y (in degrees * 1000)
52-59	X	Pose_TCP Angle Z (in degrees * 1000)
Example:	CAI11001102000040040000500500006006000070070000800800009009	
Calibration: Add image (CAI) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Calibration: Add Image
2	<b>A</b>	
3	<b>I</b>	

4	P F	P: Pass F: Fail
5-7	X	<a href="#">Error codes</a>
8-10	X	Current number of calibration objects
11-15	X	Total number of detected points
Example:	CAIP00000100021	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	Low	
Supported interfaces:	<a href="#">Availability and supported interfaces</a>	
End of telegram:	Max. 4 bytes (optional)	

## Calibration: Multi-Image (ASCII)

### Availability and supported interfaces

### Overview

Calibration: Multi-Image (CMP) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Calibration Calibrate Multi-Image Plate
2	<b>M</b>	
3	<b>P</b>	
4	1	Request version
5	X	0: Temporary 1: Permanent
6	X	Origin of the world coordinate system: 0: World coordinate system identical with the Calibration Plate Coordinate System (center of the plate). 1: Origin of the world coordinate system so that it is identical with the origin of the image coordinate system (upper left pixel). 2: Use World coordinate system of fiducials, as specified in the job file. 3: Use World coordinate system of fiducials as set in request CAW.
7	X	Mode 0: Calibration (internal and external parameters) 1: Validate (use existing calibration; at least one calibration point is added. Via back projection it can be inferred whether the point fits to the current calibration or is shifted). 2: Calibration (internal parameters only) 3: Set Measurement plane with new internal parameters 4: Set Measurement plane with existing internal parameters 5: Only calibrate Measurement plane (CPF_MF) (set Z-shift measurement plane)
Example:	CMP1105	
Calibration: Multi-Image (CMP) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Calibration Calibrate Multi-image
2	<b>M</b>	
3	<b>P</b>	
4	P F	P: Pass F: Fail
5-7	X	<a href="#">Error codes</a>
8-10	X	Field of view coverage (%)
11-15	X	Total number of detected points

16-18	X	Number of images used
19-21	X	Number of invalid images
22	X	Tilt between calibration plate poses 0: sufficient 1: not sufficient
23-30	X	Deviation calibration, RMSE [px]
31-38	X	Deviation calibration, max. [px]
39-46	X	Deviation fiducials, RMSE (in user unit * 1000)
47-54	X	Deviation fiducials, max. (in user unit * 1000)
Example:	CMPP00008900312011002000001001000020020000300300004004	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	No change	
Supported interfaces:	<a href="#">Availability and supported interfaces</a>	
End of telegram:	Max. 4 bytes (optional)	

## Calibration: Robotics multi-image (ASCII)

### Availability and supported interfaces

### Overview

Calibration: Robotics Multi-Image (CRP) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Calibration: Calibrate Robotics Plate
2	<b>R</b>	
3	<b>P</b>	
4	1	Request version
5	x	0: Temporary 1: Permanent
6	X	Origin of the world coordinate system: 4: Set world frame to User Robot Frame
7	X	Mode 0: Calibration (internal and external parameters) 1: Validate (use existing calibration; at least one calibration point is added. Via back projection it can be inferred whether the point fits to the current calibration or is shifted). 2: Calibration (internal parameters only) 3: Set Measurement plane with new internal parameters 4: Set Measurement plane with existing internal parameters 5: Only calibrate Measurement plane (CPF_MF) (set Z-shift measurement plane) 6: Calibrate Hand-Eye (TCP_CF) / Base-Eye (RF_CF) only 7: Calibrate distance value VISOR® Robotic +Z 8: Validate distance value VISOR® Robotic +Z
Example:	CRP1140	
Calibration: Robotics Multi-Image (CRP) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Calibration: Calibrate Robotics Plate
2	<b>R</b>	
3	<b>P</b>	
4	P F	P: Pass F: Fail
5-7	X	<u>Error codes</u>
8-10	X	Field of view coverage (%)
11-15	X	Total number of detected points
16-18	X	Number of images used
19-21	X	Number of invalid images
22-29	X	Deviation calibration, RMSE [px]
30-37	X	Deviation calibration, max. [px]

38-45	X	Deviations calibration plate pose Translation RMSE (in user unit * 1000)
46-53	X	Deviations calibration plate pose Translation Max. (in user unit * 1000)
54-61	X	Deviations calibration plate pose Rotation RMSE (in degrees * 1000)
62-69	X	Deviations calibration plate pose Rotation Max. (in degrees * 1000)
Example:	CRPP00009201349012004000010010000200200003003000040040000500500006006	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	No change	
Supported interfaces:	<a href="#">Availability and supported interfaces</a>	
End of telegram:	Max. 4 bytes (optional)	

## Calibration: Copy calibration (ASCII)

### Availability and supported interfaces

### Overview

#### Version 1

Calibration: Copy calibration (CCC) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Calibration: Copy calibration version 1
2	<b>C</b>	
3	<b>C</b>	
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. Note: The Working distance (Image acquisition tab) must match in all cases.
Example:	CCC110021	
Calibration: Copy calibration (CCC) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Calibration: Copy calibration
2	<b>C</b>	
3	<b>C</b>	
4	P F	P: Pass F: Fail
5 - 7	X	<u>Error codes</u>
8 - 10	X	Job number of the job where the error occurred 00: Successful >0 - Job number of the job where the error first occurred
Example:	CCCP000000	
Additional information:		
Accepted in run mode:		Yes
Accepted in configuration mode:		No
Accepted when Ready is low:		Yes
Status of Ready signal during processing:		No change
Supported interfaces:		<u>Availability and supported interfaces</u>
End of telegram:		Max. 4 bytes (optional)

**Version 2**

Calibration: Copy calibration (CCC) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Calibration: Copy calibration version 2
2	<b>C</b>	
3	<b>C</b>	
4	2	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. Note: The Working distance (Image acquisition tab) must match in all cases.
10	X	Z-Offset: 0: do not copy 1: copy
11	X	External parameters 0: do not copy 1: copy
12	X	Internal parameters 0: do not copy 1: copy
13	X	Hand eye reference 0: do not copy 1: copy
14	X	Current tool position (RF_TCP) 0: do not copy 1: copy
Example:	CCC21000111111	
Calibration: Copy calibration (CCC) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>C</b>	Calibration: Copy calibration
2	<b>C</b>	
3	<b>C</b>	
4	P F	P: Pass F: Fail
5 - 7	X	<u>Error codes</u>
8 - 10	X	Job number of the job where the error occurred 00: Successful >0 - Job number of the job where the error first occurred
Example:	CCCP000000	
Additional information:		
Accepted in run mode:	Yes	

Accepted in configuration mode:	No
Accepted when Ready is low:	Yes
Status of Ready signal during processing:	No change
Supported interfaces:	<a href="#">Availability and supported interfaces</a>
End of telegram:	Max. 4 bytes (optional)

## Calibration: Set parameters (ASCII)

### Availability and supported interfaces

#### Overview

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

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

## Calibration: Read parameter (ASCII)

### Availability and supported interfaces

#### Overview

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

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

**Calibration parameters for telegrams CSP and CGP (ASCII)**

Parameter description	Parameter number	Parameter value	Length	Calibration status after CSP
Status calibration	001	0: Invalid 1: Valid	1 byte	—*
Calibration method	002	0: None 2: Point pair list (Robotics) 3: Calibration plate (Measurement) 4: Calibration plate (Robotics) 5: Hand-Eye calibration (Robotics) 6: Base-Eye calibration (Robotics)	1 byte	invalid
Unit [user unit]	004	0: Millimeter [mm] 1: Centimeter [cm] 2: Meter [m] 3: Inch ["] 4: Arbitrary unit [au]	1 byte	no change
Internal parameters	010	Focal length (in mm *1000) Kappa (*1000) Pixel pitch X (in $\mu\text{m}$ * 1000) Pixel pitch Y (in $\mu\text{m}$ * 1000) Coordinate origin X (in pixels * 1000) Coordinate origin Y (in pixels * 1000) Image size X (number of pixels) Image size Y (number of pixels)	64 (8 * 8 bytes per value)	—*
Reference Camera- to Measuring coordinate system (CF_MF)	011	Translation X, Y, Z (in user unit * 1000) Angle X, Y, Z (in degrees * 1000)	48 (6 * 8 bytes per value)	—*
Reference Camera- to Calibration Plate Coordinate System (CF_CPF)	012	Translation X, Y, Z (in user unit * 1000) Angle X, Y, Z (in degrees * 1000)	48 (6 * 8 bytes per value)	—*
Reference Robot- to Camera coordinate system (RF_CF)	013	Translation X, Y, Z (in user unit * 1000) Angle X, Y, Z (in degrees * 1000)	48 (6 * 8 bytes per value)	—*
Reference Calibration plate- to Measuring coordinate system (CPF_MF)	014	Translation X, Y, Z (in user unit * 1000) Angle X, Y, Z (in degrees * 1000)	48 (6 * 8 bytes per value)	—*
Reference Robot- to Measuring coordinate system (RF_MF)	015	Translation X, Y, Z (in user unit * 1000) Angle X, Y, Z (in degrees * 1000)	48 (6 * 8 bytes per value)	—*
Reference TCP- to Camera coordinate system (TCP_CF)	016	Translation X, Y, Z (in user unit * 1000) Angle X, Y, Z (in degrees * 1000)	48 (6 * 8 bytes per value)	—*

Parameter description	Parameter number	Parameter value	Length	Calibration status after CSP
Reference robot- to TCP coordinate system (RF_TCP)	017	Translation X, Y, Z (in user unit * 1000) Angle X, Y, Z (in degrees * 1000)	48 (6 * 8 bytes per value)	no change
Z-shift of Measurement plane	021	Value (in user unit * 1000)	8 bytes	no change
Focal length in [mm]	022	[mm * 1000]	8 bytes	invalid (CSP for C-Mount only)
Calibration plate type	023	Character string with name of the description file e.g. "Calibration plate 15x13 50mm.cpd" (see directory \SensoPart\VISOR Vision Sensor\SensoConfig\[Version]\Calibration)	n	invalid
Fiducial 1	024	Translation X, Y, Z (in user unit * 1000)	24 (3 * 8 bytes per value)	invalid
Fiducial 2	025			
Fiducial 3	026			
Fiducial 4	027			
Number of existing calibration plate types	037	Request - Selection of type: 0: All 1: Measurement 2: Robotics Response: Number of plates	Request: 1 Response: 5	—*
Available calibration plate types (file names)	038	Request - Selection of type: 0: All 1: Measurement 2: Robotics Request - Index: 0: All file names >0: Index selection Response: File names of Calibration plates	Request: 1 / 5 Response: n (String)	—*
Robotics: Order of rotation	039	"Robotics: Order of rotation" 00: Use rotation order specified in job 01: Yaw-Pitch-Roll (e.g. Stäubli) 02: Roll-Pitch-Yaw (e.g. Kuka, Fanuc, Hanwha, ABB**, UR**) ** when using the corresponding conversion function	1	invalid
Average sensor resolution	041	Value (in user unit/pixel * 1000)	8 bytes	—*

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

## 7.4.5 Visualization

### Get image (ASCII)

#### Availability and supported interfaces

#### Overview

Get image (GIM) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Get Image
2	<b>I</b>	
3	<b>M</b>	
4	X	0: Last image 1: Last bad image 2: Last good image
Example:	GIM1	
Get image (GIM) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Get Image
2	<b>I</b>	
3	<b>M</b>	
4	P F	P: Pass F: Fail
5	X	<u>Error codes</u>
6	X	Image type 0: Grayscale 3: Bayer-Pattern_BG When converting the color image from Bayer into RGB, the appropriate image type must be considered. Pre-processing filters of the category "Arrangement" have an influence on the Bayer type. Bayer Pattern begins with blue - green.
7	X	Image result 1: Good image 0: Failed image
8 - 11	X	Number of rows e.g. 0480 / 0200
12 - 15	X	Number of columns e.g. 0640 / 0320
16 - 19	X	End of the message string if specified. Otherwise start image data from Byte no. 16.
20 ... n	X	Binary image data (rows * columns)
Example:	GIMP0004800640...	
Additional information:		
Accepted in run mode:		Yes
Accepted in configuration mode:		No

Accepted when Ready is low:	Yes
Status of Ready signal during processing:	Low
Supported interfaces:	<a href="#">Availability and supported interfaces</a>
End of telegram:	Max. 4 bytes (optional)

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

### Update visualization data (ASCII)

#### Availability and supported interfaces

#### Overview

Update visualization data (UVR) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>U</b>	Update visualization data (Update Visualization Results)
2	<b>V</b>	
3	<b>R</b>	
4	1	Request version
5	X	Image: 0: No image is created 1: Grayscale / RGB image without filter, BMP format 2: Grayscale image / Bayer pattern without filter, BMP format 3: Grayscale / RGB image with filter, BMP format 4: Grayscale image / Bayer pattern with filter, BMP format 5: Grayscale / RGB image without filter, JPEG format (low compression) 6: Grayscale / RGB image with filter, JPEG format (low compression) 7: Grayscale / RGB image without filter, JPEG format (compression high) 8: Grayscale / RGB image with filter, JPEG format (compression high)
6	X	Result XML: 0: Result file is not created 1: Result file is created
7	X	Statistic XML: 0: Statistics file is not created 1: Statistics file is created
8	X	Image type: 0: Last image (Any) 1: Last fail image (Fail) 2: Last pass image (Pass) 3: Next image (Any) 4: Next fail image (Fail) 5: Next pass image (Pass)
9 - 11	X	Directory number (constant) 001: visu001
Example:	UVR11110001	
Update visualization data (UVR) Response string from sensor (ASCII)		
Byte no.	Content	Meaning

1	<b>U</b>	Update visualization data (Update Visualization Results)
2	<b>V</b>	
3	<b>R</b>	
4	P F	P: Pass F: Fail
5 - 7	X	<a href="#">Error codes</a>
8	X	Reserved
9 - 11	X	Directory number (constant) 001: visu001
Example:	UVRP0000001	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Status of Ready signal during processing:	No change	
Supported interfaces:	<a href="#">Availability and supported interfaces</a>	
End of telegram:	Max. 4 bytes (optional)	

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

- image.bmp
- overlay.xml

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

Name		Value	Description
detector	type	pattern_matching contour contrast brightness gray caliper blob ocr data code barcode	Detector type
	number	Integer	Position in detector list
	name	String	Name of the detector defined in the configuration
roi	purpose	search teach position_control result	Type of overlay element. The different types have different colors.
	shape	rectangle rectangle_mask ellipse	Shape of the overlay element
center	x	Float	Center position in X (pixels)
	y	Float	Center position in Y (pixels)

Name		Value	Description
size	half_width	Float	Half width of overlay element
	half_height	Float	Half height of overlay element
angle	angle	Float	Angle of overlay element (degrees)
number	value	Float	Number of element types in this detector
line	x1	Float	Start point X line 1 (pixels)
	y1	Float	Start point Y line 1 (pixels)
	x2	Float	Start point X line 2 (pixels)
	y2	Float	Start point Y line 2 (pixels)

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

Detector	Search	teach	position_control	result
Pattern matching	Yes	Yes	Yes	1
Contour	Yes	Yes	Yes	200
Contour 3D	Yes	Yes	No	20
Target Mark 3D	Yes	Yes	No	1
Contrast	Yes	No	No	No
Brightness	Yes	No	No	No
Grayscale	Yes	No	No	No
Caliper	Yes	No	No	No
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)

### Availability and supported interfaces

#### Overview

Read sensor identity (GSI) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read sensor identity (Get Sensor Identity)
2	<b>S</b>	
3	<b>I</b>	
4	1	Request version
Example:	GS11	
Read sensor identity (GSI) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>G</b>	Read sensor identity (Get Sensor Identity)
2	<b>S</b>	
3	<b>I</b>	
4	P F	P: Pass F: Fail
5 - 7	X	<u>Error codes</u>
8 - 10	X	Length of the following data
11 ... n	X	Version of the firmware as well as information about the hardware. Regions are clearly separated by a semicolon.
Example:	GSIP0000262.0.0.3; V20-RO-P3-R-M-M2-L	
Additional information:		
Accepted in run mode:	Yes	
Accepted in configuration mode:	No	
Accepted when Ready is low:	Yes	
Supported interfaces:	<u>Availability and supported interfaces</u>	
End of telegram:	Max. 4 bytes (optional)	

## Update firmware (ASCII)

### Availability and supported interfaces

#### Overview

Update firmware (UFW) Request string to sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>U</b>	Update firmware
2	<b>F</b>	
3	<b>W</b>	
4	1	Request version
Example:	UFW1	
Update firmware (UFW) Response string from sensor (ASCII)		
Byte no.	Content	Meaning
1	<b>U</b>	Update firmware
2	<b>F</b>	
3	<b>W</b>	
4	P F	P: Pass F: Fail
5 - 7	X	<u>Error codes</u>
Example:	UFWP000	
Additional information:		
Accepted in run mode:		Yes
Accepted in configuration mode:		No
Accepted when Ready is low:		Yes
Supported interfaces:		<u>Availability and supported interfaces</u>
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.

## Load jobset (ASCII)

### Availability and supported interfaces

#### Overview

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

### Availability and supported interfaces

#### Overview

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

The jobset is stored with the specified name in the /tmp/ directory on the VISOR® vision sensor.

## 7.5 Description Telegrams BINARY

### 7.5.1 General

#### Reset statistics (BINARY)

[Availability and supported interfaces](#)

##### Overview

##### Reset Statistics (RST) Request string to sensor (BINARY)

Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x05	Telegram length
5	Unsigned Char	0x04	Reset statistics

##### Reset Statistics (RST) Response string from sensor (BINARY)

Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x07	Telegram length
5	Unsigned Char	0x04	Reset statistics
6 - 7	Unsigned Short	0xXX	<a href="#">Error codes</a>

Additional information:

Accepted in run mode:	Yes
Accepted in configuration mode:	No
Accepted when Ready is low:	Yes
Status of Ready signal during processing:	Low
Supported interfaces:	<a href="#">Availability and supported interfaces</a>

#### NOTE:



Statistics values can be output in the operating step Output / tab Telegram / "Payload", selection "GENERAL".

Additional information: see data output BINARY / [General](#)

## Process image from file (BINARY)

### Availability and supported interfaces

#### Overview

Process image from file (PIF) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length 9 bytes + length of subsequent data (n)
5	Unsigned Char	0x2D	Process image from file
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0x00	Reserved
8	Unsigned Char	0x01	Reserved
9	Unsigned Char	0xXX	Length of the following file name of the file (actual number of bytes with file extension, max. 255 bytes)
10 ... n	Unsigned Char	0xXX	File name (UTF-8) and format available on the device in the directory "/tmp". Allowed extensions: Monochrome sensor: .pgm Color sensor: .ppm (RGB) or .pgm (Bayer) or .jpg (RGB)
Process image from file (PIF) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length
5	Unsigned Char	0x2D	Process image from file
6-7	Unsigned Short	0xXX	<a href="#">Error codes</a>
8	Unsigned Char	0x00	Reserved
9-12	Unsigned Int	0xXX	Length of the implicit result output
13 ... n	Unsigned Char	0xXX	Output of the implicit result
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			<a href="#">Availability and supported interfaces</a>



#### **NOTE:**

Image size of the test image must match the image size of the currently active job on the device.

## 7.5.2 Control

### Trigger (BINARY)

#### Availability and supported interfaces

#### Overview

<b>Trigger (TRG) Request string to sensor (BINARY)</b>			
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)
<b>Trigger (TRG) Response string from sensor (BINARY)</b>			
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	<u>Error codes</u>
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			Yes
Accepted when Ready is low:			No
Status of Ready signal during processing:			Low
Supported interfaces:			<u>Availability and supported interfaces</u>

## Extended trigger (BINARY)

### Availability and supported interfaces

#### Overview

Extended Trigger (TRX) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length 6 bytes + length of subsequent data (n)
5	Unsigned Char	0x13	Extended trigger (trigger with index, for correlation of trigger to corresponding result data, via port 2006)
6	Unsigned Char	0xXX	Length of following data (0-99)
7 ... n	Unsigned Char	0xXX	Data
Extended Trigger (TRX) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length
5	Unsigned Char	0x13	Extended Trigger command, (response to trigger with index and result data, via port 2006, for correlation of trigger to corresponding result, Result data without index, via port 2005 also)
6 - 7	Unsigned Short	0xXX	<u>Error codes</u>
8	Unsigned Char	0xXX	Length of following data (n)
9 ... n	Unsigned Char	0xXX	Data of sending command
n+1	Unsigned Char	0xXX	Operating mode 0 = Config mode 1 = Run mode
n + 2 ... n + 5	Unsigned Int	0xXX	Length of result data
n + 6 ... m	Unsigned Char	0xXX	Result data
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			Yes
Accepted when Ready is low:			No
Status of Ready signal during processing:			Low
Supported interfaces:			<u>Availability and supported interfaces</u>

## Trigger Robotics (BINARY)

### Availability and supported interfaces

### Overview

Trigger Robotics (TRR) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length 31 (0x1F) + Length of trigger identifier in Bytes
5	Unsigned Char	0x37	Trigger Robotics
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	Length of trigger identifier in bytes
8 ... n	Unsigned Char	0xXX	Trigger Identifier
n+1...n+4	Unsigned Int	0xXX	Pose_TCP Pos. X (in user unit * 1000)
n+5...n+8	Unsigned Int	0xXX	Pose_TCP Pos. Y (in user unit * 1000)
n+9...n+12	Unsigned Int	0xXX	Pose_TCP Pos. Z (in user unit * 1000)
n+13...n+16	Unsigned Int	0xXX	Pose_TCP Angle X (in degrees * 1000)
n+17...n+20	Unsigned Int	0xXX	Pose_TCP Angle Y (in degrees * 1000)
n+20...n+24	Unsigned Int	0xXX	Pose_TCP Angle Z (in degrees * 1000)
Trigger Robotics (TRR) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x07	Telegram length 8 (0x08) + Length of trigger identifier in Bytes
5	Unsigned Char	0x37	Trigger Robotics, (Response to command Trigger without index, via port 2006. If defined: Result data without index via port 2005)
6 - 7	Unsigned Short	0xXX	<u>Error codes</u>
8	Unsigned Char	0xXX	Length of trigger identifier
9 ... n	Unsigned Char	0xXX	Trigger Identifier
n+1	Unsigned Char	0xXX	Operation Mode 0x00 = Config 0x01 = Run
n+2...n+5	Unsigned Int	0xXX	Length of the result data in bytes
n+6...m	Unsigned Int	0xXX	Result data
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			Yes
Accepted when Ready is low:			No

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

## Set Trigger ID (BINARY)

Availability and supported interfaces

Overview

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

## Read robot position (BINARY)

### Availability and supported interfaces

### Overview

Read robot position (GRP) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length Mode 0: 37 (0x25) Mode 1: 38 (0x26)
5	Unsigned Char	0x42	Read robot position
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	Mode 0: Calibration plate 1: best detector result
8-11	Int	0xXX	Distance: Distance value (in customer unit *1000)
12	Unsigned Char	0xXX	Z-axis 0: Perpendicular to the VISOR® 1: Perpendicular to the laser (at VISOR® Robotic +Z)
13	Unsigned Char	0xXX	Z-rotation 0: not aligned 1: aligned to the object
14-17	Int	0xXX	Pose_TCP Pos. X (in customer unit *1000)
18-21	Int	0xXX	Pose_TCP Pos. Y (in customer unit *1000)
22-25	Int	0xXX	Pose_TCP Pos. Z (in customer unit *1000)
26-29	Int	0xXX	Pose_TCP Angle X (in degrees *1000)
30-33	Int	0xXX	Pose_TCP Angle Y (in degrees *1000)
34-37	Int	0xXX	Pose_TCP Angle Z (in degrees *1000)
38	Unsigned Char	0xXX	Detector number (only for Mode 1)
Read robot position (GRP) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x1F	Telegram length
5	Unsigned Char	0x42	Read robot position
6 - 7	Unsigned Short	0xXX	<u>Error codes</u>
8-11	Int	0xXX	Pose_TCP Pos. X (in customer unit *1000)
12-15	Int	0xXX	Pose_TCP Pos. Y (in customer unit *1000)
16-19	Int	0xXX	Pose_TCP Pos. Z (in customer unit *1000)
20-23	Int	0xXX	Pose_TCP Angle X (in degrees *1000)
24-27	Int	0xXX	Pose_TCP Angle Y (in degrees *1000)
28-31	Int	0xXX	Pose_TCP Angle Z (in degrees *1000)
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes

Status of Ready signal during processing:	No change
Supported interfaces:	<u><a href="#">Availability and supported interfaces</a></u>

## Job change (BINARY)

### Availability and supported interfaces

#### Overview

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



#### **NOTE:**

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

## Job Change Permanent (BINARY)

### Availability and supported interfaces

#### Overview

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



#### **NOTE:**

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

## Job change by job name (BINARY)

### Availability and supported interfaces

#### Overview

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

### 7.5.3 Job settings

#### Make Job Template (BINARY)

[Availability and supported interfaces](#)

[Overview](#)

Make Job Template (MJT) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x06	Telegram length 6 bytes
5	Unsigned Char	0x41	Make Job Template
6	Unsigned Char	0x01	Request version
Make Job Template (MJT) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x07	Telegram length 7 bytes
5	Unsigned Char	0x41	Make Job Template
6 - 7	Unsigned Char	0xXX	<a href="#">Error codes</a>
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			User App

#### Auto working distance (BINARY)

[Availability and supported interfaces](#)

[Overview](#)

Auto working distance (AFC) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length 11 Bytes (0x0B) + selected options 8 Bytes (0x08)
5	Unsigned Char	0x32	Auto working distance
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	0x00: Temporary 0x01: Permanent
8	Unsigned Char	0xXX	Step size of search (0x01 - 0x05)
9	Unsigned Char	0xXX	Selection of distance value 0x00: Highest score 0x01: Min. working distance 0x02: Max. working distance 0x03: Average working distance 0x04: Median working distance 0x05: Highest score - output of all working distances found
10	Unsigned Char	0xXX	Unit 0x00: 1/1000 millimeters (µm) 0x01: Motor steps

11	Unsigned Char	0xXX	Selection of region of interest 0x00: Entire range 0x01: Selected range
12...15	Unsigned Int	X	Start of region of interest (only if region of interest selection == 0x01)
16...19	Unsigned Int	X	End of region of interest (only if selection region of interest == 0x01)
<b>Auto working distance (AFC) Response string from sensor (BINARY)</b>			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length 11 Bytes (0x0B) + working distances + score values
5	Unsigned Char	0x32	Auto working distance
6 - 7	Unsigned Short	0xXX	<a href="#">Error codes</a>
8 - 11	Unsigned Int	X	Number of output working distances
12 ... n	Unsigned Int	X	Distance value in 1/1000 mm or motor steps (4 bytes per output working distance)
n-m	Unsigned Int	X	Score value to distance value multiplied by 1000 (4 bytes per output working distance)
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			<a href="#">Availability and supported interfaces</a>

## Set working distance (BINARY)

### Availability and supported interfaces

#### Overview

<b>Set working distance (SFC) Request string to sensor (BINARY)</b>			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x0D	Telegram length
5	Unsigned Char	0x31	Set working distance
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	0: Temporary 1: Permanent Mode 5 only possible permanently.
8	Unsigned Char	0xXX	Movement 0: Absolute 1: Relative 2: Absolute with reinitialization 4: Absolute from the job (byte 9 unit always 0) 5: Copies the working distance from the current job to all jobs that contain the working distance specified in byte 10-13.
9	Unsigned Char	0xXX	Unit 0: 1/1000 millimeter 4: Steps
10 - 13	Signed integer	0xXX	Working distance (in mm *1000), in steps or job number For Mode 5: Working distance that the target job has before the copy (in mm *1000)
<b>Set working distance (SFC) Response string from sensor (BINARY)</b>			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x0B	Telegram length
5	Unsigned Char	0x31	Set working distance
6 - 7	Unsigned Short	0xXX	<u>Error codes</u>
8 - 11	INT	0xXX	Current working distance
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			<u>Availability and supported interfaces</u>

## Read working distance (BINARY)

### Availability and supported interfaces

#### Overview

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

## Auto shutter speed (BINARY)

### Availability and supported interfaces

#### Overview

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

## Set shutter speed value (BINARY)

[Availability and supported interfaces](#)

[Overview](#)

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

## Read shutter speed value (BINARY)

### Availability and supported interfaces

#### Overview

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

## Set gain value (BINARY)

### Availability and supported interfaces

#### Overview

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

## Read gain value (BINARY)

### Availability and supported interfaces

#### Overview

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

## Set parameter (BINARY)

### Availability and supported interfaces

### Overview

Set parameters (SPP/SPT) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length = 9 bytes + length of the selected parameter
5	Unsigned Char	0xXX	0x05: Set parameter permanent 0x06: Set parameter temporarily
6	Unsigned Char	0xXX	Detector number, XX = 1 ... n
7	Unsigned Char	0xXX	Set command, see <a href="#">Overview set / read detector parameters (BINARY)</a>
8 - 9	Unsigned Short	0xXX	Length of the string / value (n), see <a href="#">Overview set / read detector parameters (BINARY)</a>
10 ... n	Unsigned Char	0xXX	String / Value
Set parameters (SPP/SPT) Response string from sensor (BINARY)			
(may be delayed up to 4-5 seconds)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x08	Telegram length
5	Unsigned Char	0xXX	0x05: Set parameter permanent 0x06: Set parameter temporarily
6 - 7	Unsigned Short	0xXX	<a href="#">Error codes</a>
8	Unsigned Char	0xXX	Parameter type 0x00: I8 0x01: U8 0x02: I16 0x03: U16 0x04: I32 0x05: U32 0x06: I40 0x07: U40 0x08: Float 0x09: Double 0x0A: String 0x0B: Boolean 0x0C: Special signed8 0x0D: Undefined 0x0E: Integer array
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			Low
Supported interfaces:			<a href="#">Availability and supported interfaces</a>

## Read parameter (BINARY)

### Availability and supported interfaces

### Overview

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

**Overview set / read detector parameters (BINARY)**

Detector	Function	Value	Multiplier	Length
<b>Alignment</b>				
Pattern matching	Threshold value Min.	0x01	1000	4
Contour comparison	Threshold value Max.	0x02	1000	4
	Result offset 0: "Off" 1: "Image plane (in pixels)" 2: "Align (2D)" 3: "Robot (3D)"	0x1E	1	1
	Result offset Image plane: Pos. X	0x1F	1000	4
	Result offset Image plane: Pos. Y	0x20	1000	4
	Result offset Image plane: Angle	0x21	1000	4
	Result offset Align (2D), Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z	0x22	1000	24 (6 * 4 bytes per value)
	Calculate Result offset* with transmitted position  <ul style="list-style-type: none"> <li>Align (2D): Pos. X, Pos. Y, 0, 0, 0, Angle Z</li> <li>Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z</li> </ul> *A valid position for the detector must be available	0x23	1000	24 (6 * 4 bytes per value)
Edge detection	Probe 1: Transition 0: Any 1: Dark to light 2: Light to dark	0x65	1	1
	Probe 2: Transition 0: Any 1: Dark to light 2: Light to dark	0x66	1	1
	Probe 3: Transition 0: Any 1: Dark to light 2: Light to dark	0x67	1	1
	Probe 1: Threshold value Min.	0x68	1000	4
	Probe 2: Threshold value Min.	0x69	1000	4
	Probe 3: Threshold value Min.	0x6A	1000	4

Detector	Function	Value	Multiplier	Length	
Contour check	Calculates the missing contours	65	1	1	
	Calculates the additional contours	66	1	1	
	Calculates the roughness contours	67	1	1	
	Maximum number of invalid contours	68	1	2	
	Pixel tolerance range	69	1	2	
	Minimum length of the defective contours	6A	1	2	
	Minimum roughness of the defective contours	6B	1	2	
	Maximum roughness of the defective contours	6C	1	2	
	Automatic contrast	6D	1	1	
	Contrast value	6E	1	2	
	Minimum contour length	6F	1	2	
	Automatic alignment	70	1	1	
	<b>Detectors</b>				
Pattern matching Contour Contour 3D Shape find	Threshold value Min.	0x01	1000	4	
	Threshold value Max.	0x02	1000	4	
	Result offset 0: "Off" 1: "Image plane (in pixels)" 2: "Align (2D)" 3: "Robot (3D)"	0x1E	1	1	
	Result offset Image plane: Pos. X	0x1F	1000	4	
	Result offset Image plane: Pos. Y	0x20	1000	4	
	Result offset Image plane: Angle	0x21	1000	4	
	Result offset Align (2D), Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z	0x22	1000	24 (6 * 4 bytes per value)	
	Calculate Result offset* with transmitted position  <ul style="list-style-type: none"> <li>Align (2D): Pos. X, Pos. Y, 0, 0, 0, Angle Z</li> <li>Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z</li> </ul> *A valid position for the detector must be available	0x23	1000	24 (6 * 4 bytes per value)	
	Target Mark 3D	Result offset  <ul style="list-style-type: none"> <li>Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z</li> </ul>	0x22	1000	24 (6 * 4 bytes per value)

Detector	Function	Value	Multiplier	Length
	Calculate Result offset* with transmitted position <ul style="list-style-type: none"> <li>Robot (3D): Pos. X, Pos. Y, Pos. Z, Angle X, Angle Y, Angle Z</li> </ul> *A valid position for the detector must be available	0x23	1000	24 (6 * 4 bytes per value)
	ID of the active Target Mark	0x65	-	n
	Target Mark name	0x66	-	n
	Add current Target Mark to the list of Target Marks (parameter can only be set!) A Target Mark must have been detected.	0x67	-	n
Shape find	Geometry 0: "Circle" 1: "Square" 2: "Rectangle"	0x65	1	1
	Edge transition 0: "Dark - bright" 1: "Bright - dark"	0x66	1	1
	Diameter Length Height	0x67	1000	4
	Width	0x68	1000	4
	Tolerance	0x69	1	1
	Min. angle "Circle": [0°:0°] "Square": [-45°:45°] "Rectangle": [-90°:90°]	0x6a	1000	4
	Max. Angle "Circle": [0°:0°] "Square": [-45°:45°] "Rectangle": [-90°:90°]	0x6b	1000	4
	Min. contrast	0x6c	1	1

Detector	Function	Value	Multiplier	Length
Distance	Relationship of calibration plate coordinate system to projector coordinate system***	0x6F	1000, 1000, 1000, 1000, 1000, 1000	48 (6 * 8 bytes per value)
	Relationship of robot coordinate system to projector coordinate system***	0x70	1000, 1000, 1000, 1000, 1000, 1000	48 (6 * 8 bytes per value)
	Reference Camera- to Projector coordinate system***	0x71	1000, 1000, 1000, 1000, 1000, 1000	48 (6 * 8 bytes per value)
	Reference TCP- to Projector coordinate system***	0x72	1000, 1000, 1000, 1000, 1000, 1000	48 (6 * 8 bytes per value)
	Relationship of calibration plate coordinate system to projector coordinate system (pos. X, Y, Z)***	0x79	1000, 1000, 1000	24 (8 * 8 bytes per value)
	Relationship of calibration plate coordinate system to projector coordinate system (angle X, Y, Z)***	0x7A	1000, 1000, 1000	24 (8 * 8 bytes per value)
	Laser offset Z in mm	0x7B	1000	8
Grayscale	Threshold value Min.	0x01	1000	4
	Threshold value Max.	0x02	1000	4
	Grayscale value Min.	0x65	1000	4
	Grayscale value Max.	0x66	1000	4
	Invert grayscale value	0x67	1	4
Contrast Brightness	Threshold value Min.	0x01	1000	4
	Threshold value Max.	0x02	1000	4
Caliper	Threshold value Distance Min.	0x65	1000	4
	Threshold value Distance Max.	0x66	1000	4
	Invert distance threshold value	0x67	1	1
	Distance mode 0: Minimum 1: Maximum 2: Average 3: Median 4: Smallest opposite 5: Largest opposite	0x68	1	1
	Probe 1: Threshold value Min.	0x69	1000	4
	Probe 2: Threshold value Min.	0x6A	1000	4
	Probe 1: Smoothing	0x6B	1000	4
	Probe 2: Smoothing	0x6C	1000	4
	Probe 1: Transition 0: Any 1: Dark to light 2: Light to dark	0x6D	1	1

Detector	Function	Value	Multiplier	Length
BLOB	Probe 2: Transition 0: Any 1: Dark to light 2: Light to dark	0x6E	1	1
	Probe 1: Number of search stripes	0x6F	1	1
	Probe 2: Number of search stripes	0x70	1	4
	Grayscale value Min.	0x65	1000	4
	Grayscale value Max.	0x66	1000	4
	Invert grayscale value 0: not inverted 1: inverted	0x67	1	1
	Threshold value Number of BLOBs Min.	0x78	1	1
	Threshold value Number of BLOBs Max.	0x79	1	1
	Invert number threshold value 0: not inverted 1: inverted	0x7A	1	1
	Number of set features (read only)	0x7B	1	1
	Selection of a feature from the list	0x7C	1	1
	Feature threshold value Min.*	0x7D	1000	4
	Feature threshold value Max.*	0x7E	1000	4
	Invert feature threshold value*	0x7F	1	1
	*The feature to be changed must first be selected before it can be changed. (0x7C)			
Barcode Data code OCR	Reference string	0x65	-	n (length of string)
Color value Color area	Color space (read only)	0x15	0x01 = RGB 0x02 = HSV 0x03 = LAB	1
	Channel selection (read only)	0x16	Bit field one digit per color channel	1
	Color channel 1: Threshold value Min.	0x65	1000	4
	Color channel 1: Threshold value Max.	0x66	1000	4
	Color channel 1: Invert threshold value	0x67	1	1
	Color channel 2: Threshold value Min.	0x68	1000	4
	Color channel 2: Threshold value Max.	0x69	1000	4
	Color channel 2: Invert threshold value	0x6A	1	1
	Color channel 3: Threshold value Min.	0x6B	1000	4
	Color channel 3: Threshold value Max.	0x6C	1000	4
Color channel 3: Invert threshold value	0x6D	1	1	
Color list	Color space (read only)	0x15	1 = RGB 2 = HSV 3 = LAB	3

Detector	Function	Value	Multiplier	Length
	Channel selection (read only)	0x16	Bit field one digit per color channel	4
	Color distance threshold value	0x65	1000	n
	Set color distance threshold value active	0x66	1	n
	Number of colors in list	0x67	1	n
	Selection of a color from the list	0x68	1	n
	Color value of the selected color (color channel 1, color channel 2, color channel 3, color channel 4 [constantly 0])*	0x69	1000	32
	*The feature to be changed must first be selected before it can be changed. (0x68)			
Busbar Wafer	Threshold value Min.	0x01	1000	n
	Threshold value Max.	0x02	1000	n
Result processing	Selection of expression name	0x7A	-	n (length of string)
	Current expression*	0x7C	-	n (length of string)
	Selection of the static variable	0x8E	-	n (length of string)
	Set initial value for static variable**	0x90	-	n (length of string)
	Set current value of static variable as initial value**	0x91	-	n (length of string)
	Reset static variable to initial value	0x92	-	n (length of string)
	*The expression to be changed must first be selected before it can be changed. (0x7A)			
	**Before accessing a static variable, the static variable must first be selected. (0x8E)			

## Set region of interest (ROI) (BINARY)

### Availability and supported interfaces

### Overview

Set ROI (SRP/SRT) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length in bytes 24 byte: Circle 32 byte: rectangle, ellipse, free shape
5	Unsigned Char	0xXX	0x10: Set parameter temporarily 0x11: Set parameter permanent
6 - 9	Unsigned Int	0xXX	19 byte: Circle 27 bytes: Rectangle, ellipse, free shape
10	Unsigned Char	0xXX	Detector number
11	Unsigned Char	0xXX	Region of interest (ROI) Type 0x00: Region of interest (yellow) 0x01: Teach-in region (red) 0x02: Position control (blue)
12	Unsigned Char	0xXX	Region of interest (ROI) Shape 0x01: Circle 0x02: Rectangle 0x03: Ellipse 0x04: Free shape
13 - 16	Unsigned Int	0xXX	ROI parameter: Center X (value in [px] * 1000)
17 - 20	Unsigned Int	0xXX	ROI parameter: Center Y (value in [px] * 1000)
21 - 24	Unsigned Int	0xXX	ROI parameter: half width or radius X (value in [px] * 1000)
			Only for ellipse / rectangle / free form:
25 - 28	Unsigned Int	0xXX	ROI parameter: half height or radius Y (value in pixels * 1000)
29 - 32	Unsigned Int	0xXX	ROI parameter: Angle in ° degrees (value in ° [degrees] * 1000)
Set ROI (SRP/SRT) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1	Unsigned Int	0x07	Telegram length
5	Unsigned Char	0xXX	0x10: Set parameter temporarily 0x11: Set parameter permanent
6 - 7	Unsigned Short	0xXX	<u>Error codes</u>
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			Low
Supported interfaces:			<u>Availability and supported interfaces</u>
Parameter:			The parameters are given in the coordinate system of the Alignment and not in the coordinate system of the image.

## Read region of interest (BINARY)

### Availability and supported interfaces

#### Overview

Read region of interest (GRI) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x07	Telegram length
5	Unsigned Char	0x12	Get ROI
6	Unsigned Char	0xXX	Detector number
7	Unsigned Char	0xXX	Region of interest (ROI) Type 0x00: Region of interest (yellow) 0x01: Teach-in region (red) 0x02: Position control (blue)
Read region of interest (GRI) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length
5	Unsigned Char	0x12	Get ROI
6 - 7	Unsigned Short	0xXX	<a href="#">Error codes</a>
8 - 11	Unsigned Int	0xXX	Region of interest (ROI) Info Length in bytes from Byte 8
12	Unsigned Char	0xXX	Detector number
13	Unsigned Char	0xXX	Region of interest (ROI) Type 0x00: Region of interest (yellow) 0x01: Teach-in region (red) 0x02: Position control (blue)
14	Unsigned Char	0xXX	Region of interest (ROI) Shape 0x01: Circle 0x02: Rectangle 0x03: Ellipse 0x04: 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 / free form:			
27 - 30	Unsigned Int	0xXX	ROI parameter: half height or 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 configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			Low
Supported interfaces:			<a href="#">Availability and supported interfaces</a>

## Set ROI content (BINARY)

### Availability and supported interfaces

### Overview

Set ROI content (SRC) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length 11 bytes + length of subsequent data (n)
5	Unsigned Char	0x39	Set ROI content
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	0x00: Set parameter temporarily 0x01: Set parameter permanent
8	Unsigned Char	0xXX	Detector number 0: Alignment >0: Default detector in job
9	Unsigned Char	0x00	Reserved
10	Unsigned Char	0xXX	Bit 0: Use mask file Bit 1 - 7: reserved
11	Unsigned Char	0xXX	Length of the following file name of the file (actual number of bytes with file extension, max. 255 bytes) Or for 000: Default name mask.pgm
12 ... n	Char	0xXX	File name (UTF-8) and format available on the device in the directory "/tmp". Default name mask.pgm File format: PGM
Set ROI content (SRC) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x07	Telegram length
5	Unsigned Char	0x39	Set ROI content
6-7	Unsigned Short	0xXX	<u>Error codes</u>
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			<u>Availability and supported interfaces</u>

## Read ROI content (BINARY)

### Availability and supported interfaces


#### Overview

Read ROI content (GRC) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length 10 bytes + length of subsequent data (n)
5	Unsigned Char	0x3A	Read ROI content (Get ROI content)
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	Detector number 0: Alignment >0: Default detector in job
8	Unsigned Char	0x00	Reserved
9	Unsigned Char	0xXX	Bit 0: Use mask file (if available) Bit 1: Use pattern file Bit 2: Use contour file Bit 3-7: reserved
10	Unsigned Char	0xXX	Length of the following file name of the file (actual number of bytes with file extension, max. 255 bytes) Or for 000: Default name mask.pgm / pattern.pgm / contour.pgm
11 ... n	Char		File name (UTF-8) and format available on the device in the directory "/tmp". Default name mask.pgm / pattern.pgm / contour.pgm File format: PGM
Read ROI content (GRC) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x0B	Telegram length
5	Unsigned Char	0x3A	Set ROI content
6-7	Unsigned Short	0xXX	<u>Error codes</u>
8-9	Unsigned Short	0xXX	Width of the search region
10-11	Unsigned Short	0xXX	Height of the search region
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			<u>Availability and supported interfaces</u>

## Read job list (BINARY)

### Availability and supported interfaces


#### Overview

Read job list (GJL) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x05	Telegram length
5	Unsigned Char	0x14	Read job list
Read job list (GJL) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length
5	Unsigned Char	0x14	Read job list
6-7	Unsigned Short	0xXX	<u>Error codes</u>
8	Unsigned Char	0x01	Constant
9	Unsigned Char	0xXX	Number of jobs
10	Unsigned Char	0xXX	Active job number
	<b>NOTE:</b> The following byte sequence is repeated for each job from 1 to "Number of jobs". The byte numbers shift accordingly.		
11	Unsigned Char	0xXX	Number of subsequent bytes. This can be used to specify a unique name for job n.
11 ... n	Char	0xXX	From this position, the name for job n follows in the specified length.
n + 1 ... n + 3	Unsigned Char	0xXX	Number of subsequent bytes. A description for job n can be specified.
n + 4 ... m	Char	0xXX	From this position, the description for Job 1 follows in the specified length.
m + 1 ... m + 3	Unsigned Char	0xXX	Number of subsequent bytes. This can be used to specify a unique name for the author of job n.
m + 4 ... k	Char	0xXX	From this position, the name for the author of job n follows in the specified length.
k + 1 ... k + 4	Unsigned Int	0xXX	Date of creation of Job n (4 bytes)
k + 5 ... k + 8	Unsigned Int	0xXX	Date of last modification of job n (4 bytes)
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			<u>Availability and supported interfaces</u>

## Read detector list (BINARY)

### Availability and supported interfaces

#### Overview

Read detector list (GDL) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x05	Telegram length
5	Unsigned Char	0x15	Read detector list
Read detector list (GDL) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length
5	Unsigned Char	0x18	Read detector list
6	Unsigned Short	0xXX	<u>Error codes</u>
8	Unsigned Char	0xXX	Job number of current job
9	Unsigned Char	0xXX	Number of detectors in the current job
		<b>NOTE:</b> The following byte sequence is repeated for each detector in the job. The byte numbers shift accordingly.	
10	Unsigned Char	0xXX	Number of subsequent bytes. This allows a unique name for the detector n to be specified.
11 ... n	Unsigned Char	0xXX	From this position, the name for detector n follows, in the given length.
n + 1 ... n + 2	Unsigned Char	0xXX	Detector 0x01: Pattern matching 0x04: Contour 0x05: Grayscale 0x06: Contrast 0x07: Brightness 0x0A: Wafer 0x0B: OCR 0x0D: Data code 0x0E: Barcode 0x11: Busbar 0x12: Color value 0x13: Color area 0x14: Color list 0x15: Caliper 0x16: BLOB 0x18: Contour 3D 0x1B: Result processing 0x1C: Target Mark 3D
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			<u>Availability and supported interfaces</u>

## Teach detector (BINARY)

### Availability and supported interfaces

#### Overview

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

## Set trigger delay (BINARY)

[Availability and supported interfaces](#)

### Overview

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

## Read trigger delay (BINARY)

### Availability and supported interfaces

#### Overview

#### Get trigger delay (GTD) Request string to sensor (BINARY)

Byte no..	Data type	Content	Meaning
1 - 4	Unsigned Int	0x06	Telegram length
5	Unsigned Char	0x28	Read trigger delay
6	Unsigned Char	0xX1	Request version

#### Get trigger delay (GTD) Response string from sensor (BINARY)

Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x0B	Telegram length
5	Unsigned Char	0x28	Read trigger delay
6 - 7	Unsigned Short	0xXX	<a href="#">Error codes</a>
8 - 11	Unsigned Int	0xXX	Trigger delay in msec (max. 3000 msec) in encoder steps (max. 65535 steps)

Additional information:

Accepted in run mode:	Yes
Accepted in configuration mode:	No
Accepted when Ready is low:	Yes
Status of Ready signal during processing:	No change
Supported interfaces:	<a href="#">Availability and supported interfaces</a>

## Save job permanently (BINARY)

### Availability and supported interfaces

#### Overview

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

## Set image acquisition parameters (BINARY)

### Availability and supported interfaces

### Overview

Set image acquisition parameters (SAP) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length in bytes 12 bytes (0x0C) + length of selected parameter
5	Unsigned Char	0x3D	Set image acquisition parameters
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	0x00: Temporary 0x01: Permanent
8	Unsigned Char	0xXX	Command: Number of the image acquisition parameter, see <a href="#">Overview: Set / read detector parameters (BINARY)</a>
9 - 12	Unsigned Int	0xXX	Length of the specification for parameters (max. total length 255 bytes).
13 ... n	Unsigned Int	0xXX	Specification for parameter 1 (depending on image acquisition parameter)
Set image acquisition parameters (SAP) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x07	Telegram length
5	Unsigned Char	0x3D	Set image acquisition parameters
6 - 7	Unsigned Short	0xXX	<a href="#">Error codes</a>
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			<a href="#">Availability and supported interfaces</a>

## Read image acquisition parameters (BINARY)

### Availability and supported interfaces

#### Overview

Read image acquisition parameters (GAP) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x07	Telegram length in byte + length of selected parameter
5	Unsigned Char	0x3E	Read image acquisition parameters
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	Number of the image acquisition parameter, see <a href="#">Overview: Set / read detector parameters (BINARY)</a>
			Specification for parameter 1 (depending on image acquisition parameter)
			Specification for parameter n
Read image acquisition parameters (GAP) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length in bytes, 12 bytes (0x0C) + length of selected parameter
5	Unsigned Char	0x3E	Read image acquisition parameters
6 - 7	Unsigned Short	0xXX	<a href="#">Error codes</a>
8	Unsigned Char	0xXX	Parameter number, see <a href="#">Overview: Set / read detector parameters (BINARY)</a>
9 - 12	Unsigned Int	0xXX	Length of the following data
13 ... n	Unsigned Char	0xXX	Value for parameter 1, see <a href="#">Overview: Set / read detector parameters (BINARY)</a>
			Value for parameter n
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			<a href="#">Availability and supported interfaces</a>

**Overview: Set / read detector parameters (BINARY)**

Parameter description	Parameter number	Data type	Multiplier	Byte count	Description	Value range
Shutter speed	001	Unsigned Int	1000	8 bytes	Shutter speed in [ms] * 1000	0.026 ... 100 ms
Gain	002	Unsigned Int	1000	8 bytes	Gain Value * 1000	*Depending on image chip
Working distance	003	Unsigned Int	1000	8 bytes	Byte 1: Movement 0x00: Absolute 0x01: Relative 0x02: Absolute with reinitialization Byte 2: Unit 0x00: Millimeter [mm] 0x04: Steps Bytes 3-10: Value in unit * 1000	0 ... 100000 mm
Internal illumination	004	Unsigned Char	1	1 byte	0x00: Off 0x01: On	0, 1
Internal illumination - Selection of Quadrants	005	Unsigned Char	1		Byte 1: Q1 Byte 2: Q2 Byte 3: Q3 Byte 4: Q4 Per byte 0x00: Off 0x01: On	0, 1
External illumination	006	Unsigned Char	1		0x00: Off 0x01: On 0x02: Permanent	0, 1, 2
External illumination - pins	007	Unsigned Char	1		Byte 1: Output RD 09 Byte 2: Output YE 06 Byte 3: Output BK 07 Byte 4: Output GY 08 Per byte 0x00: Off 0x01: On	0, 1
Target laser	008	Unsigned Char	1		0x00: Off 0x01: At image acquisition 0x02: Between image acquisition	0, 1, 2
Trigger mode	009	Unsigned Char	1		0x00: Trigger 0x01: Free run	0, 1

Parameter description	Parameter number	Data type	Multiplier	Byte count	Description	Value range
Automatic brightness control	010	Unsigned Char	1	1 byte	Byte 1: active 0x00: Off 0x01: On	0, 1
		Unsigned Char	1	1 byte	Bytes 2-4: Target brightness	0-255
		Unsigned Int	1000	4 bytes	Byte 5-12: max. shutter speed value *1000	0-100
		Unsigned Int	1000	4 bytes	Byte 13-20: max. gain value * 1000	0-32 (* depending on image chip)

## Data output - Set parameter (BINARY)

### Availability and supported interfaces

### Overview

Data output - Set parameter (STP) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length in bytes, 12 bytes (0x0c) + length of selected parameter
5	Unsigned Char	0x3F	Data output: Set parameter
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	0x00: Temporary 0x01: Permanent
8	Unsigned Char	0xXX	<u>Parameter number, see parameter for STP and GTP telegrams (BINARY)</u>
9 - 12	Unsigned Int	0xXX	Length of the following data
13 ... n	Unsigned Char	0xXX	<u>Value for selected parameter, see parameter for STP and GTP telegrams (BINARY)</u>
Data output - Set parameter (STP) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x07	Telegram length
5	Unsigned Char	0x3F	Data output: Set parameter
6 - 7	Unsigned Short	0xXX	<u>Error codes</u>

## Data output - Read parameter (BINARY)

### Availability and supported interfaces

### Overview

Data output - Read parameters (GTP) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length in byte 7 (0x07) applies for parameter 0x01 - 0x05 From parameter 0x06 + length of selected parameters
5	Unsigned Char	0x40	Data output: Get parameter
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	<a href="#">Data output for Telegrams (BINARY)</a>
8 <a href="#">Applies as of parameter 0x06</a>	Unsigned Char	0x00	all rows
	Unsigned Char	>0x00	specific rows
Data output - Read parameters (GTP) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length in bytes, 12 bytes (0x0C) + length of selected parameter
5	Unsigned Char	0x40	Data output: Get parameter
6 - 7	Unsigned Short	0xXX	<a href="#">Error codes</a>
8	Unsigned Char	0xXX	<a href="#">Parameter number, see parameter for STP and GTP telegrams (BINARY)</a>
9 - 12	Unsigned Int	0xXX	Length of the following data
13 ... n	Unsigned Char	0xXX	<a href="#">Data output for Telegrams (BINARY)</a>

## Data output for Telegrams (BINARY)

Parameter description	Parameter number	Argument Specifier	Parameter value	Telegram parameters, value range
Start	0x01		0...8 byte(s)	String
Trailer	applies only to ASCII telegram		applies only to ASCII telegram	applies only to ASCII telegram
Separator	applies only to ASCII telegram		applies only to ASCII telegram	applies only to ASCII telegram
End of response	0x04		0... 4 byte(s)	String
Number of lines, detector-specific payload	0x05		1 byte	Integer total values <255
Detector-specific payload column Active	0x06	1 byte 0x00: all rows >0x00: specific rows	Unsigned char* Number of rows	[0..1]
Detector-specific payload column Detector*	0x07			
Detector-specific payload column Value*	0x08			
Detector-specific payload column Min. length	0x09	1 byte 0x00: all rows >0x00: specific rows	Unsigned char* Number of rows	[0-255]
Detector-specific payload column No. of results	0x0A	1 byte 0x00: all rows >0x00: specific rows	Unsigned short* number of lines	[0-1000]
Detector-specific payload column Factor	0x0b	1 byte 0x00: all rows >0x00: specific rows	Unsigned short* number of lines	[1; 10; 100; 1000]
Detector-specific payload column Bit depth	0x0c	1 byte 0x00: all rows >0x00: specific rows	Unsigned char* Number of rows	{0, 8, 16, 32} (Binary Requests Only)
Detector-specific payload column Sign*	0x0d			

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

## 7.5.4 Calibration

### Calibration: Initialize (BINARY)

[Availability and supported interfaces](#)

#### Overview

#### Calibration: Initialize (CCD) Request string to sensor (BINARY)

Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x05	Telegram length
5	Unsigned Char	0x1F	Initialize (Calibration: Clear Data)

#### Calibration: Initialize (CCD) Response string from sensor (BINARY)

Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x07	Telegram length
5	Unsigned Char	0x1F	Initialize (Calibration: Clear Data)
6 - 7	Unsigned Short	0xXX	<a href="#">Error codes</a>

#### Additional information:

Accepted in run mode:	Yes
Accepted in configuration mode:	No
Accepted when Ready is low:	Yes
Status of Ready signal during processing:	No change
Supported interfaces:	<a href="#">Availability and supported interfaces</a>

## Calibration: Add World Point (BINARY)

### Availability and supported interfaces

#### Overview

<b>Calibration: Add world point (CAW) Request string to sensor (BINARY)</b>			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x15	Telegram length
5	Unsigned Char	0x26	Calibration: Add world point
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	0x01: Only fiducials Calibration plate (Robotics) 0x04: World point and image point Point pair list (Robotics)
8 - 9	Unsigned Short	0x00	Constant (2 bytes)
10 - 13	Unsigned Int	0xXX	World X (in mm *1000)
14 - 17	Unsigned Int	0xXX	World Y (in mm *1000)
18 - 21	Unsigned Char	0x00	Constant (4 bytes)
<b>Calibration: Add world point (CAW) Response string from sensor (BINARY)</b>			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x11	Telegram length
5	Unsigned Char	0x26	Calibration: Add world point
6 - 7	Unsigned Short	0xXX	<u>Error codes</u>
8 - 9	Unsigned Short	0xXX	Current number of points
10 -13	Unsigned Int	0xXX	Image point X
14 - 17	Unsigned Int	0xXX	Image point Y
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			<u>Availability and supported interfaces</u>

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

## Calibration: Point pair list (BINARY)

### Availability and supported interfaces

#### Overview

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

## Calibration: Validate point pair list (BINARY)

### Availability and supported interfaces

#### Overview

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

## Calibration: Calibration plate (BINARY)

### Availability and supported interfaces

#### Overview

Calibration: Calibration Plate (CCP) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x09	Telegram length
5	Unsigned Char	0x24	Calibration: Calibration plate
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	0x00: Temporary 0x01: Permanent
8	Unsigned Char	0xXX	0x00: No fiducials are used. Origin of Measuring coordinate system identical to origin of Calibration Plate Coordinate System. 0x01: No fiducials are used. Measuring coordinate system is identical with Camera coordinate system. 0x02: Use world system, fiducials job 0x03: Use world system, fiducials request CAW.
9	Unsigned Char	0xXX	0x00: Calibration (internal and external parameters) 0x01: Validate calibration 0x02: Calibration (internal parameters only) 0x05: Calibration Transformation Measuring coordinate system 0x07: Calibration of the external parameters for VISOR® Robotic +Z
Calibration: Calibration Plate (CCP) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x3D	Telegram length
5	Unsigned Char	0x24	Calibration: Calibration plate
6 - 7	Unsigned Short	0xXX	<u>Error codes</u>
8 - 9	Unsigned Short	0xXX	Number of currently detected calibration points
10 - 13	Unsigned Int	0xXX	Deviation calibration, RMSE
14 - 17	Unsigned Int	0xXX	Deviation calibration, mean
18 - 21	Unsigned Int	0xXX	Deviation calibration, max.
22 - 25	Unsigned Int	0xXX	Deviation calibration, min.
26 - 29	Unsigned Int	0xXX	CPF_MF X (in user unit * 1000)
30 - 33	Unsigned Int	0xXX	CPF_MF Y (in user unit * 1000)
34 - 37	Unsigned Int	0x00	CPF_MF Z (in user unit * 1000)
38 - 41	Unsigned Int	0x00	CPF_MF Angle X (in degrees * 1000)
42 - 45	Unsigned Int	0x00	CPF_MF Angle Y (in degrees * 1000)
46 - 49	Unsigned Int	0xXX	CPF_MF Angle Z (in degrees * 1000)
50 - 53	Unsigned Int	0xXX	Deviation fiducials, mean
54 - 57	Unsigned Int	0xXX	Deviation fiducials, max.
58 - 61	Unsigned Int	0xXX	Deviation fiducials, min.
Additional information:			
Accepted in run mode:			Yes

Accepted in configuration mode:	No
Accepted when Ready is low:	Yes
Status of Ready signal during processing:	No change
Supported interfaces:	<a href="#"><u>Availability and supported interfaces</u></a>

## Calibration: Set Fiducial (BINARY)

### Availability and supported interfaces

#### Overview

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

## Calibration: Add image (BINARY)

### Availability and supported interfaces

### Overview

Calibration: Add image (CAI) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x22	Telegram length 34 (0x22) Bytes
5	Unsigned Char	0x34	Calibration: Add image
6	Unsigned Char	0xXX	Version 0x01: no additional image acquisition 0x02: additional image acquisition with evaluation
7	Unsigned Char	0xXX	Mode 0x01: Multi-image calibration 0x02: Hand-Eye calibration (Robotics) 0x03: Base-Eye calibration (Robotics) 0x04: Laser calibration VISOR® Robotic +Z
8	Unsigned Short	0x00	Constant
9	Unsigned Char	0xXX	Define Measurement plane 0x00: Do not use image to define Measurement plane 0x01: Use image to define Measurement plane
10	Unsigned Char	0xXX	"Robotics: Order of rotation" 0x00: Use rotation order specified in job 0x01: Yaw-Pitch-Roll (e.g. Stäubli) 0x02: Roll-Pitch-Yaw (e.g. Kuka, Fanuc, Hanwha, ABB**, UR**) ** when using the corresponding conversion function
11-14	Unsigned Char		Pose_TCP Pos. X (in user unit * 1000)
15-18	Unsigned Char		Pose_TCP Pos. Y (in user unit * 1000)
19-22	Unsigned Char		Pose_TCP Pos. Z (in user unit * 1000)
23-26	Unsigned Char		Pose_TCP Angle X (in degrees * 1000)
27-30	Unsigned Char		Pose_TCP Angle Y (in degrees * 1000)
31-34	Unsigned Char		Pose_TCP Angle Z (in degrees * 1000)
Calibration: Add image (CAI) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1-4	Unsigned Int	0x0A	Telegram length
5	Unsigned Char	0x34	Calibration: Add image
6-7	Unsigned Short	0xXX	<u>Error codes</u>
8	Unsigned Short	0xXX	Current number of calibration objects
9-10	Unsigned Char	0xXX	Total number of detected points
Additional information:			

Accepted in run mode:	Yes
Accepted in configuration mode:	Yes
Accepted when Ready is low:	No
Status of Ready signal during processing:	Low
Supported interfaces:	<a href="#">Availability and supported interfaces</a>

## Calibration: Multi-Image (BINARY)

### Availability and supported interfaces

### Overview

Calibration: Multi-Image (CMP) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x09	Telegram length 9 (0x09) Bytes
5	Unsigned Char	0x35	Calibration: Multi-Image
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	0x00: Temporary 0x01: Permanent
8	Unsigned Char	0xXX	Origin of the world coordinate system: 0x00: World coordinate system identical with the Calibration Plate Coordinate System (center of the plate). 0x01: Origin of the world coordinate system so that it is identical with the origin of the image coordinate system (upper left pixel).. 0x02: (only for Calibration plate (Robotics)) Use World coordinate system of fiducials, as specified in the job file. 0x03: (only for Calibration plate (Robotics)) Use World coordinate system of fiducials as set in request CAW.
9	Unsigned Char	0xXX	Mode 0x00: Calibration (internal and external parameters) 0x01: Validate (use existing calibration; at least one calibration point is added. Via back projection it can be inferred whether the point fits to the current calibration or is shifted). 0x02: Calibration (internal parameters only) 0x03: Set Measurement plane with new internal parameters 0x04: Set Measurement plane with existing internal parameters 0x05: Only calibrate Measurement plane (CPF_MF) (set Z-shift measurement plane)
Calibration: Multi-Image (CMP) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1-4	Unsigned Int	0x1D	Telegram length 29 (0x1D) Bytes
5	Unsigned Char	0x35	Calibration: Multi-Image
6-7	Unsigned Short	0xXX	<u>Error codes</u>
8	Unsigned Char	0xXX	Field of view coverage (%) 0x00: no coverage 0x64: Coverage 100%
9-10	Unsigned Short	0xXX	Total number of detected points
11	Unsigned Char	0xXX	Number of images used
12	Unsigned Char	0xXX	Number of invalid images
13	Unsigned Char	0xXX	Tilt between calibration plate poses 0x00: sufficient 0x01: not sufficient
14-17	Unsigned Int	0xXX	Deviation calibration plate RMSE [px]
18-21	Unsigned Int	0xXX	Deviation calibration plate Max. [px]

22-25	Unsigned Int	0xXX	Deviation fiducials, RMSE (in user unit * 1000)
26-29	Unsigned Int	0xXX	Deviation fiducials, max. [px]
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			<a href="#">Availability and supported interfaces</a>

## Calibration: Robotics multi-image (BINARY)

### Availability and supported interfaces

### Overview

Calibration: Robotics Multi-Image (CRP) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x09	Telegram length in bytes 9 Byte
5	Unsigned Char	0x36	Calibration: Calibration plate Robotics
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	0x00: Temporary 0x01: Permanent
8	Unsigned Char	0xXX	Origin of the world coordinate system: 0x04: Set origin of coordinate system equal to Robot Coordinate System
9	Unsigned Char	X	Mode 0x00: Calibration (internal and external parameters) 0x01: Validate (use existing calibration; at least one calibration point is added. Via back projection it can be inferred whether the point fits to the current calibration or is shifted). 0x02: Calibration (internal parameters only) 0x03: Set Measurement plane with new internal parameters 0x04: Set Measurement plane with existing internal parameters 0x05: Only calibrate Measurement plane (CPF_MF) (set Z-shift measurement plane) 0x06: Calibrate Hand-Eye (TCP_CF) / Base-Eye (RF_CF) only 0x07: Calibrate distance value VISOR® Robotic +Z 0x08: Validate distance value VISOR® Robotic +Z
Calibration: Robotics Multi-Image (CRP) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1-4	Unsigned Int	0x2C	Telegram length 44 (0x2C) Bytes
5	Unsigned Char	0x36	Calibration: Calibration plate Robotics
6-7	Unsigned Short	0xXX	<u>Error codes</u>
8	Unsigned Char	0xXX	Field of view coverage 0x00: not sufficient 0x01: sufficient
9-10	Unsigned Short	0xXX	Total number of detected points
11	Unsigned Char	0xXX	Number of images used
12	Unsigned Char	0xXX	Number of invalid images
13-16	Unsigned Int	0xXX	Deviation calibration plate RMSE [px]
17-20	Unsigned Int	0xXX	Deviation calibration plate Max. [px]
21-24	Unsigned Int	0xXX	Deviations calibration plate pose Translation RMSE (in user unit * 1000)
25-28	Unsigned Int	0xXX	Deviations calibration plate pose Translation Max. (in user unit * 1000)
29-32	Unsigned Int	0xXX	Deviations calibration plate pose Rotation RMSE (in degrees * 1000)

33-36	Unsigned Int	0xXX	Deviations calibration plate pose Rotation Max. (in degrees * 1000)
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes

## Calibration: Copy calibration (BINARY)

### Availability and supported interfaces

### Overview

#### Version 1

Calibration: Copy calibration (CCC) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x09	Telegram length
5	Unsigned Char	0x25	Calibration: Copy calibration
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0x01	Constant
8	Unsigned Char	0xXX	Destination 0 : Copy to all jobs >0: Copy to specified job
9	Unsigned Char	0xXX	0: Always copy when the calibration is active. 1: Only copy if the calibration method is the same. Note: The Working distance (Image acquisition tab) must match in all cases.
Calibration: Copy calibration (CCC version 1 and 2) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x08	Telegram length
5	Unsigned Char	0x25	Calibration: Copy calibration
6 - 7	Unsigned Short	0xXX	<u>Error codes</u>
8	Unsigned Char	0xXX	00: Successful >0 : Job number at which the error occurs.
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			<u>Availability and supported interfaces</u>

#### Version 2

Calibration: Copy calibration (CCC) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x0E	Telegram length 0x0E 14 bytes
5	Unsigned Char	0x25	Calibration: Copy calibration
6	Unsigned Char	0x02	Request version
7	Unsigned Char	0x01	Constant

8	Unsigned Char	0xXX	Destination 0x00 : Copy to all jobs >0x00: Copy to specified job
9	Unsigned Char	0xXX	0x00: Always copy when the calibration is active. 0x01: Only copy if the calibration method is the same. Note: The Working distance (Image acquisition tab) must match in all cases.
10	Unsigned Char	0xXX	Z-Offset: 0x00: do not copy 0x01: copy
11	Unsigned Char	0xXX	External parameters 0x00: do not copy 0x01: copy
12	Unsigned Char	0xXX	Internal parameters 0x00: do not copy 0x01: copy
13	Unsigned Char	0xXX	Hand eye reference 0x00: do not copy 0x01: copy
14	Unsigned Char	0xXX	Current tool position (RF_TCP) 0x00: do not copy 0x01: copy

**Calibration: Copy calibration (CCC) Response string from sensor (BINARY)**

Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x08	Telegram length
5	Unsigned Char	0x25	Calibration: Copy calibration
6 - 7	Unsigned Short	0xXX	<a href="#">Error codes</a>
8	Unsigned Char	0xXX	00: Successful >0 : Job number at which the error occurs.

Additional information:

Accepted in run mode:	Yes
Accepted in configuration mode:	No
Accepted when Ready is low:	Yes
Status of Ready signal during processing:	No change
Supported interfaces:	<a href="#">Availability and supported interfaces</a>

## Calibration: Set parameters (BINARY)

### Availability and supported interfaces

#### Overview

Calibration: Set parameters (CSP) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length in bytes, 12 bytes (0x0C) + length of selected parameter
5	Unsigned Char	0x29	Calibration: Set parameter
6	Unsigned Char	0x01	Request version
7	Unsigned Char	0xXX	0x00: Temporary 0x01: Permanent
8	Unsigned Char	0xXX	Parameter number, see <a href="#">Calibration parameters for telegrams CSP and CGP (BINARY)</a>
9 - 12	Unsigned Int	0xXX	Length of the following data
13 ... n	Unsigned Char	0xXX	Parameter value, see <a href="#">Calibration parameters for telegrams CSP and CGP (BINARY)</a>
Calibration: Set parameters (CSP) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x07	Telegram length
5	Unsigned Char	0x29	Calibration: Set parameter
6 - 7	Unsigned Short	0xXX	<a href="#">Error codes</a>
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			<a href="#">Availability and supported interfaces</a>

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

## Calibration: Read parameter (BINARY)

### Availability and supported interfaces

#### Overview

Calibration: Read parameters (CGP) Request string to sensor (BINARY)			
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, see <a href="#">Calibration parameters for telegrams CSP and CGP (BINARY)</a>
Calibration: Read parameters (CGP) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length in bytes, 12 bytes (0x0C) + length of selected parameter
5	Unsigned Char	0x2A	Calibration: Get parameter
6 - 7	Unsigned Short	0xXX	<a href="#">Error codes</a>
8	Unsigned Char	0xXX	Parameter number, see <a href="#">Calibration parameters for telegrams CSP and CGP (BINARY)</a>
9 - 12	Unsigned Int	0xXX	Length of the following data
13 ... n	Unsigned Char	0xXX	Parameter value, see <a href="#">Calibration parameters for telegrams CSP and CGP (BINARY)</a>
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			No change
Supported interfaces:			<a href="#">Availability and supported interfaces</a>

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

**Calibration parameters for telegrams CSP and CGP (BINARY)**

Parameter description	Parameter number	Parameter value	Length	Calibration status after CSP
Status calibration	0x01	0x00: Invalid 0x01: Valid	1 byte	—*
Selection of calibration method	0x02	0x00: None 0x02: Point pair list (Robotics) 0x03: Calibration plate (Measurement) 0x04: Calibration plate (Robotics) 0x05: Hand-Eye calibration (Robotics) 0x06: Base-Eye calibration (Robotics)	1 byte	invalid
User unit	0x04	0x00: Millimeter [mm] 0x01: Centimeter [cm] 0x02: Meter [m] 0x03: Inch ["] 0x04: Arbitrary unit [au]	1 byte	no change
Internal parameters	0x0A	Focal length (in mm *1000) Kappa (*1000) Pixel pitch X (in $\mu\text{m}$ * 1000) Pixel pitch Y (in $\mu\text{m}$ * 1000) Coordinate origin X (in pixels * 1000) Coordinate origin Y (in pixels * 1000) Image size X (number of pixels) Image size Y (number of pixels)	0x20 (8 * 4 bytes per value)	—*
Reference Camera- to Measuring coordinate system (CF_MF)	0x0B	Translation X, Y, Z (in user unit * 1000) Angle X, Y, Z (in degrees * 1000)	0x18 (6 * 4 bytes per value)	—*
Reference Camera- to Calibration Plate Coordinate System (CF_CPF)	0x0C	Translation X, Y, Z (in user unit * 1000) Angle X, Y, Z (in degrees * 1000)	0x18 (6 * 4 bytes per value)	—*
Reference Robot- to Camera coordinate system (RF_CF)	0x0D	Translation X, Y, Z (in user unit * 1000) Angle X, Y, Z (in degrees * 1000)	0x18 (6 * 4 bytes per value)	—*
Reference Calibration plate- to Measuring coordinate system (CPF_MF)	0x0E	Translation X, Y, Z (in user unit * 1000) Angle X, Y, Z (in degrees * 1000)	0x18 (6 * 4 bytes per value)	—*
Reference Robot- to Measuring coordinate system (RF_MF)	0x0F	Translation X, Y, Z (in user unit * 1000) Angle X, Y, Z (in degrees * 1000)	0x18 (6 * 4 bytes per value)	—*
Reference TCP- to Camera coordinate system (TCP_CF)	0x10	Translation X, Y, Z (in user unit * 1000) Angle X, Y, Z (in degrees * 1000)	0x18 (6 * 4 bytes per value)	—*

Parameter description	Parameter number	Parameter value	Length	Calibration status after CSP
Reference robot- to TCP coordinate system (RF_TCP)	0x11	Translation X, Y, Z (in user unit * 1000) Angle X, Y, Z (in degrees * 1000)	0x18 (6 * 4 bytes per value)	no change
Z-shift of Measurement plane	0x15	(in user unit * 1000)	4 bytes	no change
Focal length in [mm]	0x16	[mm * 1000]	4 bytes	invalid (CSP for C-Mount only)
Calibration plate type	0x17	Character string with name of the description file e.g. "Calibration plate 15x13 50mm.cpd" (see directory \SensoPart\VISOR Vision Sensor\SensoConfig\[Version]\Calibration)	n	invalid
Fiducial 1	0x18	Translation X, Y, Z (in user unit * 1000)	0x0C (3* 4 bytes per value)	invalid
Fiducial 2	0x19			
Fiducial 3	0x1A			
Fiducial 4	0x1B			
Number of existing calibration plate types	0x25	Request - Selection of type: 0x00: All 0x01: Measurement 0x02: Robotics Response: Number of plates	Request: 1 Response: 2	—*
Available calibration plate types (file names)	0x26	Request - Selection of type: 0x00: All 0x01: Measurement 0x02: Robotics Request - Index: 0: All file names >0: Index selection Response: File names of Calibration plates	Request: 1 Response: 5 (String)	—*
Robotics: Order of rotation	0x27	"Robotics: Order of rotation" 0x00: Use rotation order specified in job 0x01: Yaw-Pitch-Roll (e.g. Stäubli) 0x02: Roll-Pitch-Yaw (e.g. Kuka, Fanuc, Hanwha, ABB**, UR**) ** when using the corresponding conversion function	1 byte	invalid
Average sensor resolution	0x29	Value (in user unit/pixel * 1000)	4 bytes	—*

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

## 7.5.5 Visualization

### Get image (BINARY)

[Availability and supported interfaces](#)

[Overview](#)

Get image (GIM) Request string to sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0x06	Telegram length
5	Unsigned Char	0x03	Get image
6	Unsigned Char	0xXX	0x00: Last image 0x01: Last failed image 0x02: Last good image
Get image (GIM) Response string from sensor (BINARY)			
Byte no.	Data type	Content	Meaning
1 - 4	Unsigned Int	0xXX	Telegram length in bytes, 13 bytes (0x0D) + number of bytes depending on the image format e.g. 00 04 B0 0D (Dez. 307213)
5	Unsigned Char	0x03	Get image
6 - 7	Unsigned Short	0xXX	<a href="#">Error codes</a>
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	Resulting image 00: Failed image 01: Good image
10 - 11	Unsigned Short	0xXX	Number of rows e.g. 01 E0 = 480
12 - 13	Unsigned Short	0xXX	Number of columns e.g. 02 80 = 640
14 ... n	Unsigned Char	0xXX	Binary image data (rows * columns)
Additional information:			
Accepted in run mode:			Yes
Accepted in configuration mode:			No
Accepted when Ready is low:			Yes
Status of Ready signal during processing:			Low
Supported interfaces:			<a href="#">Availability and supported interfaces</a>

## 7.6 Error codes

Error code	Error code HEX	Description
000	0x00	Successful
001	0x01	Error

Error code	Error code HEX	Description
003	0x03	Invalid parameter data
005	0x05	Invalid telegram
004	0x04	Telegram rejected, simultaneous module requests received
006	0x06	Input parameters with invalid size or invalid value
007	0x07	File does not exist
008	0x08	Recorder off
009	0x09	Matching image of requested type not found
011	0x0B	Invalid data length
012	0x0C	Not allowed due to jobset mismatch
013	0x0D	Failed to start new job from job set
014	0x0E	File already exists
015	0x0F	Insufficient memory in flash
016	0x10	Firmware version mismatch
018	0x12	Calibration plate data not available
019	0x13	Request not supported for the specified interface
020	0x14	More than one vis file present
021	0x15	Sensor type not suitable for vis-file
022	0x16	No part found
023	0x17	Invalid calibration type
024	0x18	Job change rejected. It is not possible to change the job during an active mode.
025	0x19	Change of mode rejected during an active mode. First send "Mode End" or "Break Sequence".
029	0x1D	Temporary job change rejected because job checksum is active.
030	0x1E	Calibration not activated / Calibration not supported
031	0x1F	Error while copying
032	0x20	Mismatched input conditions for destination job
033	0x21	Calibration / validation error
034	0x22	Invalid number of points
035	0x23	Calibration error: Add point (e.g. last job result failed)
036	0x24	Invalid fiducial
037	0x25	Jobset protected: permanent changes to job not allowed
038	0x26	Parameter values are not available to write / read
039	0x27	Sensor is in configuration mode, telegram was rejected
040	0x28	Error while writing / reading parameter value
041	0x29	No matching job found
042	0x2A	Format error
043	0x2B	Jobset / job saving error
044	0x2C	Focus lock time exceeded
045	0x2D	Error with multiple files

Error code	Error code HEX	Description
046	0x2E	Working distance could not be determined
047	0x2F	"Min. processing time per image" was not observed
048	0x30	Region of interest size (ROI) does not match
049	0x31	Region of interest (ROI) Freeform not selected
050	0x32	Calibration method does not match
051	0x33	No calibration plate found
052	0x34	Number of images too low
053	0x35	No calibration possible: distance between tool positions not plausible
054	0x36	Rotation between images not sufficient
055	0x37	Tilt between the images not sufficient
056	0x38	Invalid pose type
057	0x39	Function not activated
058	0x40	Function is not supported by current device
059	0x3B	Secondary VISOR® timeout
060	0x3C	Secondary participant TCP/IP connection error
061	0x3D	Secondary participant TCP/IP invalid response
062	0x3E	Violation of a calibration rule

## 8 Telegram

### 8.1 Description of data output ASCII

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

Basic string structure:

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

Output data (ASCII):

<OPTIONAL FIELDS>				
Parameter	Description	Length ASCII [byte]	Data type	Available for
Selected fields	With this checkbox all selected fields are displayed. The checkbox "Selected fields" itself is not displayed.	16	The output sequence is from left to right and from top to bottom, i.e. one byte is set per active checkbox, starting with the LSB.	All types
Telegram length	Number of characters including the characters for the telegram length itself.	1 ... 10	E.g. output string with 10 characters; telegram length 10 + 2 characters (one byte per decimal place) = 12	All types
Status byte	Returns the Trigger mode.	3	PPF = Trigger PFP = Free run	All types
Detector results	Output of overall result for each detector.	5 ... 259	Byte 1 = AND link of all detectors Byte 2 = total result of the Alignment Byte 3 = total result of the current job  Followed by the number of detectors; one byte per decimal place. Followed by one byte "P = Detector Pass" or one byte "F = Detector Fail" per detector. The detector results are displayed in descending order.	All types

<b>&lt;OPTIONAL FIELDS&gt;</b>				
<b>Parameter</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Digital outputs	Returns the logic gate result for each digital output.	2 ... 7	Byte 1 Number of active outputs (logic gate result assigned) Followed by bytes 2 – 7; one byte per output P = Detector pass F = Detector fail 0 = Inactive output (gap between two active outputs)	All types
log. Outputs	Returns the logic gate result for each logic output.	1 ... 259	Starting from byte 1 Number of active outputs (logic gate result assigned); 1 byte per decimal place Following bytes: One byte per logic output P = Detector pass F = Detector fail 0 = Inactive output (gap between two active outputs)	All types
Execution time	Returns the execution time for the last evaluation.	1 ... 3	Signed integer	All types
Active job	Returns the job for the last evaluation.	1 ... 3	Unsigned Int U8	All types

## **<PAYLOAD>**

### **Overview of Payload - Values**

#### **General**

<b>&lt;PAYLOAD&gt; General</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
"All evaluations" counter	Total number of checks	1 ... 11	Signed integer	GENERAL
Pass parts counter	Number of inspections with result "OK"	1 ... 11	Signed integer	GENERAL
Fail parts counter	Number of inspections with result "Error"	1 ... 11	Signed integer	GENERAL

<b>&lt;PAYLOAD&gt; General</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Timeout	Indicates that the maximum cycle time has been exceeded.	1	BOOL	GENERAL
Recording	Indicates the number of image acquisition repetitions for the last evaluation Only in combination with repeat mode.	1 ... 3	INT	GENERAL
String	This field can be used to enter a constant string into the data output.	1 ... 50	STRING	GENERAL
Job checksum	Calculates a checksum over the active job. This takes into account all job-specific settings except the "Changed" date. Changing settings that are global for the jobset will change the checksum in all jobs. If the checksum is determined for a job, no temporary changes can be made for this job in run mode.	8	STRING	GENERAL
TriggerId	The trigger identifier is sent to the VISOR® with the requests TRX and TRR, together with the trigger signal. The trigger identifier can be set with the request STI.	n	STRING	GENERAL

## Base values

Note: [ . . . ] serves as a placeholder for different elements for which the corresponding output value is available (e.g. P1\_Score).

<b>&lt;PAYLOAD&gt; Base values</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Execution time ExecTimeDetector	Execution time of individual detector in [msec].	1 ... 11	Signed integer	All detectors and Alignment
Score ScoreDetector	Score value of the detector [%]	1 ... 6	Signed integer	All detectors and Alignment
Score vector Score [... ]_Score	Score values of all candidates **	n	Signed integer	All detectors and Alignment
Detector result ResultDetector	Boolean detector result	1	BOOL	All detectors and Alignment
Result vector Result [... ]_Result	Boolean detector results of all candidates **	n	BOOL	All detectors and Alignment
Result vector ValidVector	Vector containing the result (1/0) of the instances found		BOOL	
Number of objects NumObjects	Number of objects found [units]	1 ... 5	Signed integer	   
Number of valid objects NumObjectsValid	Number of valid objects found [units]	1 ... 5	Signed integer	  
Too many BLOBs TooManyObjects		1	BOOL	

**Position / location**

Note: [ . . . ] serves as a placeholder for different elements for which the corresponding output value is available (e.g. P1\_PosX).

<b>&lt;PAYLOAD&gt; Position / location</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Pos. X PosX [... ]_PosX	X coordinate for the found position, 1/1000 [user unit] **	1 ... 11	Signed integer	       
Pos. Y PosY [... ]_PosY	Y coordinate for the found position, 1/1000 [user unit] **	1 ... 11	Signed integer	       

<b>&lt;PAYLOAD&gt; Position / location</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Pos. Z PosZ [... ]_PosZ	Z coordinate of the found position, 1/1000 [user unit]	1 ... 11	Signed integer	With Result offset: +, +, 0 +, 0, 0
Delta Pos. X DeltaPosX	X position delta between the taught object and the found object, 1/1000 [user unit]	1 ... 11	Signed integer	+, +, 0 +, 0, 0, 3D, 3D
Delta Pos. Y DeltaPosY	Y position delta between the taught object and the found object, 1/1000 [user unit]	1 ... 11	Signed integer	+, +, 0 +, 0, 0, 3D, 3D
Delta Pos. Z DeltaPosZ	Z position delta between the taught object and the found object, 1/1000 [user unit]	1 ... 11	Signed integer	3D With Result offset: +, +, 0 +, 0
Angle X AngleX	Orientation of the found object, relative to the X-axis, 1/1000 [°]	1 ... 11	Signed integer	3D, 3D With Result offset: +, +, 0 +, 0, 0
Angle Y AngleY	Orientation of the found object, relative to the Y-axis, 1/1000 [°]	1 ... 11	Signed integer	3D, 3D With Result offset: +, +, 0 +, 0, 0
Angle Z AngleZ	Orientation of the found object, relative to the Z-axis, 1/1000 [°]	1 ... 11	Signed integer	+, +, 0, +, 0, 0 +, 0, 0, 3D, 3D, 3D, 3D, 3D, 3D, 3D, 3D 3D, 3D, 3D, 3D, 3D, 3D, 3D, 3D
Angle Z (180) [... ]_AngleZ180	Orientation of object width (long axis) [°], Value range: -90° ... 90° 0° = East, counterclockwise	1 ... 7	Signed integer	0
Angle Z (360) [... ]_AngleZ360	Orientation of object width (long axis) [°], Value range: -180° ... 180° 0° = East, counterclockwise	1 ... 7	Signed integer	0





<b>&lt;PAYLOAD&gt; Position / location</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Delta Angle X DeltaAngleX	Angle between taught-in and found object, referred to the X-axis, 1/1000 [°]	1 ... 7	Signed integer	With Result offset:
Delta Angle Y DeltaAngleY	Angle between taught-in and found object, referred to the Y-axis, 1/1000 [°]	1 ... 7	Signed integer	With Result offset:
Delta Angle Z DeltaAngleZ	Angle between taught-in and found object, referred to the Z-axis, 1/1000 [°]	1 ... 7	Signed integer	
Uncertainty Translation UncertaintyPos	Uncertainty of translation 1/1000 [user unit]	1 ... 7	Signed integer	
Uncertainty Rotation UncertaintyAngle	Uncertainty of rotation 1/1000 [°]	1 ... 7	Signed integer	
Pose 3D (X, Y, Z, Angle X, Angle Y, Angle Z) Pose3D	Coordinates of the found object, 1/1000 [user unit] Angle: 1/1000 degrees	1...7 bytes per value; separated by specified separator	Signed integer	With Result offset:
Pose 3D (X, Y, Z, Angle X, Angle Y, Angle Z 360) [... ]_Pose3D360	Coordinates of the found object, 1/1000 [user unit] Angle: 1/1000 degrees			
Pose 3D (X, Y, Z, Angle X, Angle Y, Angle Z 180) [... ]_Pose3D180	Coordinates of the found object, 1/1000 [user unit] Angle: 1/1000 degrees			
Delta Pose 3D (X, Y, Z, Angle X, Angle Y, Angle Z) DeltaPose3D	Delta coordinates of the found object, 1/1000 [user unit] Angle: 1/1000 degrees	1...7 bytes per value; separated by specified separator	Signed integer	With Result offset:
Position control PosValid	Vector indicating results of position control for individual candidates		BOOL	

### Measurement - edge based

Note: [...] serves as a placeholder for different elements for which the corresponding output value is available (e.g. P1\_NumRays).

<b>&lt;PAYLOAD&gt; Measurement</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Distance Distance	Calculated distance [user unit]**	1 ... 11	Signed integer	☑
Winning search stripe [...]_Winner				
Number of search stripes [...]_NumRays	Number of parallel search stripes into which the width of the region of interest is divided. [units]	1 ... 5	Signed integer	☑ (only Edge detection) ☑
Number of valid search stripes [...]_RayValid	Number of search stripes used to generate results [units]	1 ... 3	Signed integer	☑ (only Edge detection) ☑
Search stripe Pos. X [...]_RayPosX	Vector of the detection points of the individual search stripes (Position X)	n x 1...11 bytes (with n = number of search stripes)	Signed integer	☑ (only Edge detection) ☑
Search stripe Pos. Y [...]_RayPosY	Vector of the detection points of the individual search stripes (Position Y)		Signed integer	☑ (only Edge detection) ☑
Search stripe distance [...]_RayDistance	Calculated distance [user unit] / 1000 per search stripe pair	1 ... 11	Signed integer	☑

#### Caliper Detector: Definition of P1, P2, P12







	P1	P2	P12
One probe, both sides 	P1 = left scanning line	P2 = right scanning line	P12 results from the center between the scanning lines
One probe, one side 	P1 = left edge of the ROI	P2 = determined scanning line	
Two probes, antiparallel (opposite direction) 	P1 = left scanning line	P2 = right scanning line	
Two probes, same direction 	P1 = left scanning line	P2 = right scanning line	

**Alignment (Edge detection): Definition of P1, P2, P3**

- P1 = Scanning line from probe 1
- P2 = Scanning line from probe 2
- P3 = Scanning line from probe 3

**Measurement - BLOB**

<b>&lt;PAYLOAD&gt; Measurement</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Min. threshold GrayMin	Lower threshold for the binarization of the objects. 0...255	1 ... 3	Unsigned Int	
Max. threshold GrayMax	Upper threshold for the binarization of the objects. 0...255	1 ... 3	Unsigned Int	
Inverted threshold GrayInvert	Specifies whether the range Min <-> Max is inverted. P: inverted F: not inverted	1	BOOL	
Area Area	Area of BLOB without holes, 1/1000 [pixels]	1 ... 11	Signed integer	
Area (incl. holes) AreaIncludingHoles	Area of BLOB including holes, 1/1000 [pixels]	1 ... 11	Signed integer	
Contour length ContourLength	Number of pixels of outer contour, 1/1000 [pixels]	n	Signed integer	
Compactness 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	
Gray value, average GrayMean	Average gray scale value of all the pixels that belong to the BLOB.	n	Signed integer	
Gray value, standard deviation GrayStdDev	Standard deviation of the gray values of all pixels belonging to the BLOB.	n	Signed integer	






<b>&lt;PAYLOAD&gt; Measurement</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Gray value, minimum GrayMin	Smallest value of the gray values of all pixels belonging to the BLOB.	n	Signed integer	
Gray value, maximum GrayMax	Largest value of the gray values of all pixels belonging to the BLOB.	n	Signed integer	
Number of holes NumHoles	Number of holes within the BLOB.	1 ... 11	Signed integer	
Height Height	Height of the geometric element [user unit]*, Height $\geq 0$ , height $\leq$ width	1 ... 11	Signed integer	
Width Width	Width of the geometric element [user unit]* Width $\geq 0$ , width $\geq$ height	1 ... 11	Signed integer	
Radius C1_Radius	Radius of the fitted circle [user unit]	1 ... 11	Signed integer	
Deviation, inside C1_DeviationMin	Returns the largest deviation between the BLOB contour and the contour of the geometric element (deviation inside the fitted circle). [User unit * 1000]	1 ... 7	Signed integer	
Deviation, outside C1_DeviationMax	Returns the largest deviation between the BLOB contour and the contour of the geometric element (deviation outside the fitted circle). [user unit]	1 ... 7	Signed integer	

<b>&lt;PAYLOAD&gt; Measurement</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Deviation, mean C1_DeviationAbsMean	Returns the mean of the absolute "inside" and "outside" deviation values between the BLOB contour and the contour of the geometric element (circle).	1 ... 7	Signed integer	
Axial ratio E1_AxialRatio	Ratio of the long to the short axis (a / b)	1 ... 7	Signed integer	
Eccentricity E1_Eccentricity	Numerical eccentricity Value range of 0.0 to 1.0	n	Signed integer	
Face up / down, area E1_FaceUpArea	Face up / down position, based on: area, position indicated by sign, 1/1000	n	Signed integer	




## Result processing

<b>&lt;PAYLOAD&gt; Result processing</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Whole numbers	All data values of data type INT	1 ... 11	Signed integer	
Real numbers	All data values of the data type REAL	1 ... 11	Signed integer	
String	All data values of the data type STR	n	STRING	
Boolean value	All data values of the data type BOOL	1	BOOL	



Identification

<b>&lt;PAYLOAD&gt; Identification</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
String String Target Mark ID Id Target Mark name Name	Content of the read code or content of the Target Mark or assigned Target Mark name. If a fixed string length is desired, the minimum string length (Payload) and the maximum string length (detector settings) must be set to the same value (e.g. 127).	0 ... 255	STRING	
String length StringLengthSymbols Length of Target Mark name (characters) NameNumSymbols	Length of read code [characters]	1 ... 6	Signed integer	
String length (bytes) StringLengthBytes Length of Target Mark name (bytes) NameNumBytes Length of Target Mark ID (bytes) IdNumBytes	Length of read code [bytes]	1 ... 6	Signed integer	
Reference string met CompareResult	Content check for the read information. The content of the read information is checked on the basis of regular expressions (see detector Data code, Reference string tab)	1	BOOL	
Truncated StringTruncated	Code complete or truncated F: Code complete P: Code truncated	1	BOOL	















## Identification - quality

<b>&lt;PAYLOAD&gt; Identification - Quality</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Quality - overall Q	Output of all Q parameters. Depending on the selected code type and standard.	1 byte per value; separated by specified separator For 2D code parameter Q9 (mean light): 1...3	Unsigned Char; for 2D Code Q9 (Meanlight) Unsigned Short	
Quality - individual Q1 Q2 ...	Output of the individual quality values: Selection of Q1-Q24 in dependence on the selected code type and standard. Numbers: 1-4 Letters: A-F	1 For 2D code parameter Q9 (mean light): 1...3	Unsigned Char; for 2D Code Q9 (Meanlight) Unsigned Short	
Min. Quality Quality	Used to check whether the minimum required quality is being met	1 ... 7	Unsigned Int	

## Color

<b>&lt;PAYLOAD&gt; Color</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Color value:  <ul style="list-style-type: none"> <li>• Red, green, blue</li> <li>• Hue, saturation, lightness</li> <li>• Luminance, a, b</li> </ul> ValueChannel1	Value for color parameter	0 ... 7	Signed integer	
Color distance DistanceColor	Distance of the current color versus the taught-in color	0 – 7	Signed integer	



















## Solar

<PAYLOAD> Solar				
Value	Description	Length ASCII [byte]	Data type	Available for
Height Height	Height of the geometric element [user unit]*, Height $\geq 0$ , height $\leq$ width	1 ... 11	Signed integer	 
Width Width	Width of the geometric element [user unit]* Width $\geq 0$ , width $\geq$ height	1 ... 11	Signed integer	 
Number of wafers NumWafers	Wafer found [units]	1	Unsigned Char	
Number of busbars NumBusbars	Number of busbars found [units]	1	Unsigned Char	
Result bytes ResultDetail	Individual results of the detector bit by bit	2	Unsigned Char	
Result bytes ResultByte	Individual results of the detector bit by bit	1	Unsigned Char	
Busbar position X BusbarPosX	Position X of the center of gravity of all busbars	4	Signed integer	
Busbar position Y BusbarPosY	Position Y of the center of gravity of all busbars	4	Signed integer	
Area busbars BusbarArea	Total area of all busbars	4	Signed integer	
Wafer / busbar found	Wafer / busbar found	1	BOOL	 
Wafer brightness threshold Histogram	Fixed input of the brightness threshold.	1	Unsigned Char	
Min. threshold GrayMin	Lower threshold for binarization	1	Unsigned Char	
Max. threshold GrayMax	Upper threshold for binarization	1	Unsigned Char	
Wafer area AreaCheck	The wafer's area 1/1000 [pixels]	4	Signed integer	
Total contour points NumContourPoints	Number of contour points of the outer contour of the wafer	4	Signed integer	

<b>&lt;PAYLOAD&gt; Solar</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Contour points with deviations NumContourPointsInvalid	Number of contour points of the outer contour of the wafer with deviations	4	Signed integer	
Number of chipping spots NumDefects	Number of chipping spots	4	Signed integer	
Pads NumPads	Number of pads in the selected busbar	4	Signed integer	
Area check result (good/bad) BusbarAreaCheck	Area check result (good/bad)	1	BOOL	
Position control PosControlActive	Checking whether the center of gravity of the busbars / wafer is within the specified position tolerance	1	BOOL	

**Extended**

<b>&lt;PAYLOAD&gt; Extended</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Scaling Scale	Current scaling factor to the taught-in reference. 1/1000 (factor). Value range of 0.5 to 2	3 ... 4	Unsigned Int	(only Contour comparison) 
Security Confidence	Output of the security values of the individual characters. The reliability value specifies how reliably the reader was able to interpret a character. Value range of 0 to 100 [%]	n	Unsigned Int	ABC
Reference string met CompareResult	The output string matches the reference string.	1	BOOL	ABC
Contrast Contrast	Code contrast Value range of 0 to 100 [%]	n	Unsigned Int	

<b>&lt;PAYLOAD&gt; Extended</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Correction DecodingError	Number of modules corrected by error corrections [units]	n	Unsigned Int	
Result index Index	List index	n	Signed integer	
Module height ModuleHeight	Height of a module in [px]	4	Signed integer	
Module width ModuleWidth	Width of a module in [px]	4	Signed integer	
Length, Reference Contour LengthModel	Length of all reference contours	4	Signed integer	
Length, good contours LengthOk	Length of all good contours	4	Signed integer	
Length, missing contours LengthMissing	Length of all missing contours	4	Signed integer	
Length, additional contours LengthOutbreak	Length of all additional contours	4	Signed integer	
Length, roughness defects LengthRoughness	Length of all roughness defects	4	Signed integer	
Length, defects LengthDefects	Length of all defects	4	Signed integer	
Number, reference contours NumModel	Number of all reference contours	4	Signed integer	
Number, good contours NumOk	Number of all good contours	4	Signed integer	
Number, missing contours NumMissing	Number of all missing contours	4	Signed integer	
Number, additional contours NumOutbreak	Number of all additional contours	4	Signed integer	
Number, roughness contours NumRoughness	Number of all roughness contours	4	Signed integer	
Number, defects NumDefects	Number of all defects	4	Signed integer	
Defect type DefectType	Defect types Missing contour=1 Additional contours=2 Roughness=3	4	Signed integer	
Length DefectLength	Defect length	4	Signed integer	

<b>&lt;PAYLOAD&gt; Extended</b>				
<b>Value</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Distance DefectDistance	Maximum distance of missing contour to reference contour	4	Signed integer	↻
Roughness, min. DefectRoughnessMin	Minimum deviation	4	Signed integer	↻
Roughness, median DefectRoughnessMedian	Median deviation	4	Signed integer	↻
Roughness, max. DefectRoughnessMax	Maximum deviation	4	Signed integer	↻

<b>&lt;CHKSUM&gt;</b>				
<b>Parameter</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Check sum	Decimal value of the XOR check sum of all bytes in the telegram. Transmitted as the last 3 bytes, e.g. Hex.{71} = Dec. 113 = Hex{31}{31}{33}	3	Unsigned Int	All types

<b>&lt;TRAILER&gt;</b>				
<b>Parameter</b>	<b>Description</b>	<b>Length ASCII [byte]</b>	<b>Data type</b>	<b>Available for</b>
Trailer	Characters appended at the end of the string User-defined, up to a max. of 8 characters	0 ... 8	Unsigned Int	All types


**\*NOTE:**

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

\*\*Detector Caliper: Depending on the selected Distance mode. For "Minimum/maximum by search stripe" = vector with two elements [min; max].

## 8.2 Description of data output BINARY

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

Basic string structure:

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



**NOTE:**

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

### Output data (BINARY):

<OPTIONAL FIELDS>				
Parameter	Description	Length BINARY [byte]	Data type	Available for
Selected fields	With this checkbox all selected fields are displayed. The checkbox "Selected fields" itself is not displayed.	2	The output sequence is from left to right and from top to bottom, i.e. one bit is set per active checkbox, starting with the lowest-value one.	All types
Telegram length	Number of characters including the characters for the telegram length itself.	2	Unsigned Short	All types
Status byte	Returns the Trigger mode.	2	0x06 0x00 = Trigger; 0x05 0x00 = Free run	All types
Detector results	Output of overall result for each detector. Byte 1 Bit 1 (LSB) = Global job result (1 = Pass, 0 = Fail) Bit 2 = Boolean result, only Alignment, Alignment inactive = True	3 ... 35		All types
Digital outputs	Returns the logic gate result for each digital output.	n	Bytes 1 and 2: Number of active Outputs Byte 3 ... n: Outputs, bit-coded	All types
log. Outputs	Returns the logic gate result for each logic output.	n	Byte 1 and byte 2: Number of active! logic outputs Byte 3 – n All active logic outputs,	All types

<b>&lt;OPTIONAL FIELDS&gt;</b>				
<b>Parameter</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Execution time	Returns the execution time for the last evaluation.	4	Signed integer	All types
Active job	Returns the job for the last evaluation.	1	Unsigned Int U8	All types

## **<PAYLOAD>**

### **Overview of Payload - Values**

#### **General**

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

<b>&lt;PAYLOAD&gt; General</b>				
<b>Value</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Job checksum	Calculates a checksum over the active job. This takes into account all job-specific settings except the "Changed" date. Changing settings that are global for the jobset will change the checksum in all jobs. If the checksum is determined for a job, no temporary changes can be made for this job in run mode.	8	STRING	GENERAL
TriggerId	The trigger identifier is sent to the VISOR® with the requests TRX and TRR, together with the trigger signal. The trigger identifier can be set with the request STI.	n	STRING	GENERAL

## Base values

Note: [...] serves as a placeholder for different elements for which the corresponding output value is available (e.g. P1\_Score).

<b>&lt;PAYLOAD&gt; Base values</b>				
<b>Value</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Execution time ExecTimeDetector	Execution time of individual detector in [msec].	4	Signed integer	All detectors and Alignment
Score ScoreDetector	Score value of the detector [%]	4	Signed integer	All detectors and Alignment
Score vector Score [...]_Score	Score values of all candidates **	n	Signed integer	All detectors and Alignment
Detector result ResultDetector	Boolean detector result	1	BOOL	All detectors and Alignment



































<b>&lt;PAYLOAD&gt; Base values</b>				
<b>Value</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Result vector Result [... ]_Result	Boolean detector results of all candidates **	n	BOOL	All detectors and Alignment
Result vector ValidVector	Vector containing the result (1/0) of the instances found	n	BOOL	
Number of objects NumObjects	Number of objects found [units]	4	Signed integer	
Number of valid objects NumObjectsValid	Number of valid objects found [units]	4	Signed integer	
Too many BLOBs TooManyObjects		1	BOOL	

### Position / location

Note: [ . . . ] serves as a placeholder for different elements for which the corresponding output value is available (e.g. P1\_PosX).

<b>&lt;PAYLOAD&gt; Position / location</b>				
<b>Value</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Pos. X PosX [... ]_PosX	X coordinate for the found position, 1/1000 [user unit] **	4	Signed integer	
Pos. Y PosY [... ]_PosY	Y coordinate for the found position, 1/1000 [user unit] **	4	Signed integer	
Pos. Z PosZ [... ]_PosZ	Z coordinate of the found position, 1/1000 [user unit]	4	Signed integer	With Result offset: 
Delta Pos. X DeltaPosX	X position delta between the taught object and the found object, 1/1000 [user unit]	4	Signed integer	



<PAYLOAD> Position / location				
Value	Description	Length BINARY [byte]	Data type	Available for
Delta Angle Y DeltaAngleY	Angle between taught-in and found object, referred to the Y-axis, 1/1000 [°]	4	Signed integer	  With Result offset:     
Delta Angle Z DeltaAngleZ	Angle between taught-in and found object, referred to the Z-axis, 1/1000 [°]	4	Signed integer	      
Uncertainty Translation UncertaintyPos	Uncertainty of translation 1/1000 [user unit]	4	Signed integer	
Uncertainty Rotation UncertaintyAngle	Uncertainty of rotation 1/1000 [°]	4	Signed integer	
Pose 3D (X, Y, Z, Angle X, Angle Y, Angle Z) Pose3D	Coordinates of the found object, 1/1000 [user unit] Angle: 1/1000 degrees	6 x 4 bytes	Signed integer	  With Result offset:     
Pose 3D (X, Y, Z, Angle X, Angle Y, Angle Z 360) [... ]_Pose3D360	Coordinates of the found object, 1/1000 [user unit] Angle: 1/1000 degrees	6 x 4 bytes	Signed integer	
Pose 3D (X, Y, Z, Angle X, Angle Y, Angle Z 180) [... ]_Pose3D180	Coordinates of the found object, 1/1000 [user unit] Angle: 1/1000 degrees	6 x 4 bytes	Signed integer	
Delta Pose 3D (X, Y, Z, Angle X, Angle Y, Angle Z) DeltaPose3D	Delta coordinates of the found object, 1/1000 [user unit] Angle: 1/1000 degrees	6 x 4 bytes	Signed integer	  With Result offset:     
Position control PosValid	Vector indicating results of position control for individual candidates	1	BOOL	 

## Measurement - edge based

Note: [...] serves as a placeholder for different elements for which the corresponding output value is available (e.g. P1\_NumRays).

<b>&lt;PAYLOAD&gt; Measurement</b>				
<b>Value</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Distance Distance	Calculated distance [user unit] **	4	Signed integer	
Winning search stripe [...]_Winner	Index of the winning search stripe	4	Signed integer	
Number of search stripes [...]_NumRays	Number of parallel search stripes into which the width of the region of interest is divided. [units]	4	Signed integer	(only Edge detection)
Number of valid search stripes [...]_RayValid	Number of search stripes used to generate results [units]	4	Signed integer	(only Edge detection)
Search stripe Pos. X [...]_RayPosX	Vector of the detection points of the individual search stripes (Position X)	n x 4 bytes (with n = number of search stripes)	Signed integer	(only Edge detection)
[...]_RayPosY	Vector of the detection points of the individual search stripes (Position Y)		Signed integer	(only Edge detection)
Search stripe distance [...]_RayDistance	Calculated distance [user unit] / 1000 per search stripe pair		Signed integer	

**Caliper Detector: Definition of P1, P2, P12**










	P1	P2	P12
One probe, both sides 	P1 = left scanning line	P2 = right scanning line	P12 results from the center between the scanning lines
One probe, one side 	P1 = left edge of the ROI	P2 = determined scanning line	
Two probes, antiparallel (opposite direction) 	P1 = left scanning line	P2 = right scanning line	
Two probes, same direction 	P1 = left scanning line	P2 = right scanning line	



**Alignment (Edge detection): Definition of P1, P2, P3**

- P1 = Scanning line from probe 1
- P2 = Scanning line from probe 2
- P3 = Scanning line from probe 3





## Measurement - BLOB

<PAYLOAD> Measurement				
Value	Description	Length BINARY [byte]	Data type	Available for
Min. threshold GrayMin	Lower threshold for the binarization of the objects. 0...255	4	Unsigned Int	
Max. threshold GrayMax	Upper threshold for the binarization of the objects. 0...255	4	Unsigned Int	
Inverted threshold GrayInvert	Specifies whether the range Min <-> Max is inverted. P: inverted F: not inverted	4	Unsigned Char	
Area Area	Area of BLOB without holes, 1/1000 [pixels]	4	Signed integer	
Area (incl. holes) AreaIncludingHoles	Area of BLOB including holes, 1/1000 [pixels]	4	Signed integer	
Contour length ContourLength	Number of pixels of outer contour, 1/1000 [pixels]	4	Signed integer	
Compactness Compactness	BLOB compactness (circle =1; other > 1). The more the shape of the BLOB deviates from a circle, the greater the compactness value will be.	4	Signed integer	
Gray value, average GrayMean	Average gray scale value of all the pixels that belong to the BLOB.	4	Signed integer	
Gray value, standard deviation GrayStdDev	Standard deviation of the gray values of all pixels belonging to the BLOB.	4	Signed integer	
Gray value, minimum GrayMin	Smallest value of the gray values of all pixels belonging to the BLOB.	4	Signed integer	






<b>&lt;PAYLOAD&gt; Measurement</b>				
<b>Value</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Gray value, maximum GrayMax	Largest value of the gray values of all pixels belonging to the BLOB.	4	Signed integer	
Number of holes NumHoles	Number of holes within the BLOB.	4	Signed integer	
Height Height	Height of the geometric element [user unit]*, Height $\geq 0$ , height $\leq$ width	4	Signed integer	
Width Width	Width of the geometric element [user unit]* Width $\geq 0$ , width $\geq$ height	4	Signed integer	
Radius C1_Radius	Radius of the fitted circle [user unit]	4	Signed integer	
Deviation, inside C1_DeviationMin	Returns the largest deviation between the BLOB contour and the contour of the geometric element (deviation inside the fitted circle). [User unit * 1000]	4	Signed integer	
Deviation, outside C1_DeviationMax	Returns the largest deviation between the BLOB contour and the contour of the geometric element (deviation outside the fitted circle). [user unit]	4	Signed integer	
Deviation, mean C1_DeviationAbsMean	Returns the mean of the absolute "inside" and "outside" deviation values between the BLOB contour and the contour of the geometric element (circle).	4	Signed integer	
Axial ratio E1_AxialRatio	Ratio of the long to the short axis (a / b)	4	Signed integer	

<b>&lt;PAYLOAD&gt; Measurement</b>				
<b>Value</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Eccentricity E1_Eccentricity	Numerical eccentricity Value range of 0.0 to 1.0	4	Signed integer	
Face up / down, area E1_FaceUpArea	Face up / down position, based on: area, position indicated by sign, 1/1000	4	Signed integer	



### Result processing

<b>&lt;PAYLOAD&gt; Result processing</b>				
<b>Value</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Whole numbers	All data values of data type INT	4	Signed integer	
Real numbers	All data values of the data type REAL	4	Signed integer	
String	All data values of the data type STR	n	STRING	
Boolean value	All data values of the data type BOOL	1	BOOL	



## Identification

<b>&lt;PAYLOAD&gt; Identification</b>				
<b>Value</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
String String Target Mark ID Id Target Mark name Name	Content of the read code or content of the Target Mark or assigned Target Mark name. If a fixed string length is desired, the minimum string length (Payload) and the maximum string length (detector settings) must be set to the same value (e.g. 127).	n	STRING	
String length StringLengthSymbols Length of Target Mark name (characters) NameNumSymbols	Length of read code [characters]	n	Signed integer	
String length (bytes) StringLengthBytes Length of Target Mark name (bytes) NameNumBytes Length of Target Mark ID (bytes) IdNumBytes	Length of read code [bytes]	n	Signed integer	
Reference string met CompareResult	Content check for the read information. The content of the read information is checked on the basis of regular expressions (see detector Data code, Reference string tab)	1	BOOL	
Truncated StringTruncated	Code complete or truncated F: Code complete P: Code truncated	1	BOOL	

### Identification - quality

<b>&lt;PAYLOAD&gt; Identification - Quality</b>				
<b>Value</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Quality - overall Q	Output of all Q parameters. Depending on the selected code type and standard.	n	Unsigned Char; for 2D Code Q9 (Meanlight) Unsigned Short	
Quality - individual Q1 Q2 ...	Output of the individual quality values: Selection of Q1-Q24 in dependence on the selected code type and standard. Numbers: 1-4 Letters: A-F	n	Unsigned Char; for 2D Code Q9 (Meanlight) Unsigned Short	
Min. Quality Quality	Used to check whether the minimum required quality is being met	n	Unsigned Int	ABC

### Color

<b>&lt;PAYLOAD&gt; Color</b>				
<b>Value</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Color value: <ul style="list-style-type: none"><li>• Red, green, blue</li><li>• Hue, saturation, lightness</li><li>• Luminance, a, b</li></ul> ValueChannel1	Value for color parameter	4	Signed integer	
Color distance DistanceColor	Distance of the current color versus the taught-in color	4	Signed integer	

## Solar

<PAYLOAD> Solar				
Value	Description	Length BINARY [byte]	Data type	Available for
Height Height	Height of the geometric element [user unit]*, Height $\geq 0$ , height $\leq$ width	4	Signed integer	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Width Width	Width of the geometric element [user unit]* Width $\geq 0$ , width $\geq$ height	4	Signed integer	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Number of wafers NumWafers	Wafer found [units]	1	Unsigned Char	<input checked="" type="checkbox"/>
Number of busbars NumBusbars	Number of busbars found [units]	1	Unsigned Char	<input checked="" type="checkbox"/>
Result bytes ResultDetail	Individual results of the detector bit by bit	1	Unsigned Char	<input checked="" type="checkbox"/>
Result bytes ResultByte	Individual results of the detector bit by bit	1	Unsigned Char	<input checked="" type="checkbox"/>
Busbar position X BusbarPosX	Position X of the center of gravity of all busbars	4	Signed integer	<input checked="" type="checkbox"/>
Busbar position Y BusbarPosY	Position Y of the center of gravity of all busbars	4	Signed integer	<input checked="" type="checkbox"/>
Area busbars BusbarArea	Total area of all busbars	4	Signed integer	<input checked="" type="checkbox"/>
Wafer / busbar found	Wafer / busbar found	1	BOOL	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>
Wafer brightness threshold Histogram	Fixed input of the brightness threshold.	1	Unsigned Char	<input checked="" type="checkbox"/>
Min. threshold GrayMin	Lower threshold for binarization	1	Unsigned Char	<input checked="" type="checkbox"/>
Max. threshold GrayMax	Upper threshold for binarization	1	Unsigned Char	<input checked="" type="checkbox"/>
Wafer area AreaCheck	The wafer's area 1/1000 [pixels]	4	Signed integer	<input checked="" type="checkbox"/>
Total contour points NumContourPoints	Number of contour points of the outer contour of the wafer	4	Signed integer	<input checked="" type="checkbox"/>

<b>&lt;PAYLOAD&gt; Solar</b>				
<b>Value</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Contour points with deviations NumContourPointsInvalid	Number of contour points of the outer contour of the wafer with deviations	4	Signed integer	
Number of chipping spots NumDefects	Number of chipping spots	4	Signed integer	
Pads NumPads	Number of pads in the selected busbar	4	Signed integer	
Area check result (good/bad) BusbarAreaCheck	Area check result (good/bad)	1	BOOL	
Position control PosControlActive	Checking whether the center of gravity of the busbars / wafer is within the specified position tolerance	1	BOOL	

**Extended**

<b>&lt;PAYLOAD&gt; Extended</b>				
<b>Value</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Scaling Scale	Current scaling factor to the taught-in reference. 1/1000 (factor). Value range of 0.5 to 2	4	Unsigned Int	(only Contour comparison) 
Security Confidence	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	Unsigned Int	ABC
Reference string met CompareResult	The output string matches the reference string.	1	BOOL	ABC
Contrast Contrast	Code contrast Value range of 0 to 100 [%]	4	Unsigned Int	

<b>&lt;PAYLOAD&gt; Extended</b>				
<b>Value</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Correction DecodingError	Number of modules corrected by error corrections [units]	4	Unsigned Int	
Result index Index	List index	4	Signed integer	
Module height ModuleHeight	Height of a module in [px]	4	Signed integer	
Module width ModuleWidth	Width of a module in [px]	4	Signed integer	
Length, Reference Contour LengthModel	Length of all reference contours [px]	4	Signed integer	
Length, good contours LengthOk	Length of all good contours [px]	4	Signed integer	
Length, missing contours LengthMissing	Length of the missing contours [px]	4	Signed integer	
Length, additional contours LengthOutbreak	Length of all additional contours [px]	4	Signed integer	
Length, roughness defects LengthRoughness	Length of all roughness defects	4	Signed integer	
Length, defects LengthDefects	Length of all defects	4	Signed integer	
Number, reference contours NumModel	Number of all reference contours	4	Signed integer	
Number, good contours NumOk	Number of all good contours	4	Signed integer	
Number, missing contours NumMissing	Number of all missing contours	4	Signed integer	
Number, additional contours NumOutbreak	Number of all additional contours	4	Signed integer	
Number, roughness contours NumRoughness	Number of all roughness contours	4	Signed integer	
Number, defects NumDefects	Number of all defects	4	Signed integer	
Defect type DefectType	Defect types Missing contour=1 Additional contours=2 Roughness=3	4	Signed integer	
Length DefectLength	Defect length	4	Signed integer	

<b>&lt;PAYLOAD&gt; Extended</b>				
<b>Value</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Distance DefectDistance	Maximum distance of missing contour to reference contour	4	Signed integer	↻
Roughness, min. DefectRoughnessMin	Minimum deviation	4	Signed integer	↻
Roughness, median DefectRoughnessMedian	Median deviation	4	Signed integer	↻
Roughness, max. DefectRoughnessMax	Maximum deviation	4	Signed integer	↻

<b>&lt;CHKSUM&gt;</b>				
<b>Parameter</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Check sum	XOR check sum of all bytes in the telegram. Is transmitted as the last byte.	1	Unsigned Int	All types

<b>&lt;TRAILER&gt;</b>				
<b>Parameter</b>	<b>Description</b>	<b>Length BINARY [byte]</b>	<b>Data type</b>	<b>Available for</b>
Trailer	Characters appended at the end of the string	0 ... 8	Unsigned Int	All types



**NOTE:**

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

\*\*Detector Caliper: Depending on the selected Distance mode. For "Minimum/maximum by search stripe" = vector with two elements [min; max].

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" values (BINARY protocol)

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

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

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

Because Big-endian (from Sensor) is sent, calculation goes as following  
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.

**We look ahead**  
Yesterday, today and in the future



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

[www.sensopart.com](http://www.sensopart.com)