BALLUFF

BVS Cockpit Manual
3.2

english

www.balluff.com

1 Introduction

In automated industrial processes in particular, items are identified by means of optical codes (barcodes, 2D codes or plaintext), the items' attributes, such as their presence, contours, dimensions, colors and positions are detected and verified. The findings of these checks can be prepared a variety of ways and forwarded to the primary control system. The BVS Cockpit operating system was developed to provide the user with easy-to-learn and user-friendly machine vision solutions that can span all products. The BVS Cockpit requires a valid HALCON license, which is supplied with all products.

Outstanding features are:

- Extremely simple, user-friendly operation
- Clear structure:
  Configuration, Monitoring, Statistics mode
- Intuitive creation of inspection programs using managed configuration of clear individual tools
- Structured exchange of data to primary control systems
- Automatic, configurable generation and filing of test results
- Integrated, flexible display of process statistics

BVS Cockpit is available on the following product platforms:

- BVS SC **SMART***CAMERA* und BVS **SMART***CAMERA* Lite
- BAE PD SmartVision Controller (in combination with industrial cameras BVS CA)
- BAI BVS-CA BVS Cockpit Windows (in combination with industrial cameras BVS CA)

1.1 Installation of and access to the BVS Cockpit

The BVS Cockpit can be used and installed in various ways, depending on the product category. To use BVS Cockpit one of the following web browsers is required:

- Google Chrome version 32.0 and above
- Mozilla Firefox version 24.0 and above
- Microsoft Internet Explorer version 11 and above
- Microsoft Edge version 40 and above

NOTE

With **Release version 3.2**

- BVS **SMART***CAMERA* and
- BVS **SMART***CAMERA* Lite

are shipped with the **fixed IP address** 192.168.10.2 (**Subnet mask**: 255.255.255.0).

1.1.1 SMARTCAMERA

NOTE

JavaScript must be enabled in the web browser.

The BVS Cockpit is already integrated into all **SMART***CAMERA* variants and activated via an internal license.

1. Open the web browser on a PC that is connected to the network (Google Chrome version 32.0 or above, Mozilla Firefox version 24.0 or above, Microsoft Internet Explorer version 10 or above) and
2. enter the following URL: **http://192.168.10.2**.

In order that the call works, your local network must be set up correctly. For this case, consult your network administrator.

With the corresponding network configuration (keywords "Auto IP" and "DNS"), you can access the **SMART***CAMERA* via its serial number: **http://sc-serialnumber** bzw. **http://sl-serialnumber**. The serial number can be found on the name plate of the **SMART***CAMERA* (e.g. *SC-170700005DE*).

1.1.2 SmartVision Controller

NOTE

JavaScript must be enabled in the web browser.

On the SmartVision Controller, the BVS Cockpit is already integrated and activated via an external USB license dongle.

1. Open the web browser on a PC that is connected to the network (Google Chrome version 32.0 or above, Mozilla Firefox version 24.0 or above, Microsoft Internet Explorer version 10 or above)
2. and enter the following URL: http://svc-serialnumber:9000. (first instance only; continue sequentially with 9001, etc.)

The serial number can be found on the name plate of the SmartVision Controller (e.g. SVC-123456).

1.1.3 BVS Cockpit Windows

This product variant enables the BVS Cockpit to be installed on any PC running a Windows operating system and to then be operated with one or several BVS CA industrial cameras.

Your computer must meet the following system requirements:

- 64 bit Windows 7 / Windows 10 (recommended)
- Dual-core 2 GHz CPU
- 8 GB RAM

NOTE

JavaScript must be enabled in the web browser.

NOTE

Independently of the used hardware the number of monitored properties highly affects the rendering performance and reactivity of the browser. Reducing the number of displayed overlays (lines, circles, etc.) on the monitor page, for example, also accelerates the rendering speed.

The BVS Cockpit system must initially be installed on the relevant PC using the USB storage medium supplied and is automatically activated via the USB license dongle supplied.

1. From your Windows PC, run the setup program **BVS_Cockpit_Windows_Vx.y.z.msi** (e.g. *BVS_Cockpit_Windows_V2.3.0.msi*) on the Balluff USB storage medium supplied.
2. Select the program directory you want and assign a working directory.
   In the working directory, four workspace folders ("workspace0" ... "workspace3") are created and four BVS Cockpit shortcuts to the BVS Cockpit are automatically placed on the desktop.
3. Plug the USB license dongle into an available USB port.
4. Run one or several the BVS Cockpit shortcuts.
   The appropriate BVS Cockpit session is launched in the workspace specified earlier.
   Your standard web browser (Google Chrome version 32.0 or above, Mozilla Firefox version 24.0 or above, Microsoft Internet Explorer version 10 or above) will then automatically open the URL *http://localhost* with the appropriate port (e.g. *http://localhost:9000* for the first instance, *http://localhost:9001* for the second, etc.).
5. You can then use a web browser (Google Chrome version 32.0 or above, Mozilla Firefox version 24.0 or above, Microsoft Internet Explorer version 10 or above) to access the relevant BVS Cockpit sessions via the network by entering the IP address of the PC: **http://PC-IP-Adresse:9000** – (e.g. 192.168.0.25) (first instance only, continue sequentially with 9001, etc.)

If you want to use an industrial camera with BVS Cockpit, for image acquisitions you have to install a suitable USB3 Vision or GigE Vision compliant hardware driver. For the Balluff *Camera* BVS CA series the driver's name is **mvGenTL_Acquire-x86_64-X.yy.z.exe** which is available from the Balluff Website www.balluff.com.

NOTE

Company IT departments often implement special network structures. Sub-networks are explicitly generated to divide computers into different access areas. The BVS Controller or **SMART***CAMERA* must, of course, be located within the same access area as the PC. If problems arise, please ask your IT administrator for help.

1.2 URL parameters

1.2.1 Language of the user interface

BVS Cockpit is displayed in the language that is set in the browser. In most cases this is the language of the operating system. In order to display the user interface in another language you have to adapt the language settings of your browser. Alternatively BVS Cockpit can be called with the parameter locale and a language code. E.g.: *http://[URL of BVS Cockpit]/?locale=de* for german language, *http://[URL of BVS Cockpit]/?locale=zh* for chinese language.

NOTE

BVS Cockpit features different languages for the user interface. If the language of your browser is not supported the user interface is displayed in English.

1.2.2 Full screen mode

A full screen view of the image without the possibility for interaction, may be activated by calling BVS Cockpit with the parameter *visualization* and its value *imageonly*. E.g. *http://[URL of BVS Cockpit]/?visualization=imageonly*. In this view, the image of the first tool "**Get image**" within the inspection program and the graphical elements of the results of all other tools are displayed. The result table is not displayed.

2 Note, Attention, Danger

2.1 Symbols

NOTE

This symbol indicates general notes.

2.2 Explanation of the warnings

Always observe the warnings in these instructions and the measures described to avoid hazards. The warnings used here contain various signal words and are structured as follows:

SIGNAL WORD
Type and source of the hazard
Consequences if not complied with
→ Measures to avoid hazards

The individual signal words mean:

ATTENTION
Indicates a danger that can lead to **damage** or **destruction** of the product.

DANGER
The general warning symbol in conjunction with the signal word DANGER identifies a hazard which, if not avoided, will certainly result in **death** or **serious injuries**.

All due care and attention has been taken in preparing this manual. In view of our policy of continuous product improvement, however, we can accept no liability for completeness and correctness of the information contained in this manual. We make every effort to provide you with a flawless product.

In the context of the applicable statutory regulations, we shall accept no liability for direct damage, indirect damage or third-party damage resulting from the acquisition or operation of a Balluff product. Our liability for intent and gross negligence is unaffected. In any case, the extend of our liability shall be limited to the purchase price.

3 About this user manual

Purpose of this manual	This manual provides you with an overview of how to operate BVS Cockpit. BVS Cockpit is a browser-based software package that enables you to create and configure inspection programs, without the need for any programmer or detailed knowledge of image processing. For a quick introduction to the software, we recommend that you read the manual in the specified order.
Content of this manual	The user interface is explained in the "**User interface**" section and the subsequent "**Tool Box**" section explains the tools available for creating an inspection program. Finally, in the section on "**Creating an initial inspection program**", you will learn how to create your first inspection program. The learning process is supported with other use cases, with example programs, in the section entitled "**Use cases**".

3.1 Graphics and symbols

The following symbols are used in the "**Tool Box**" section of this operating manual, to describe the wizard:

	This symbol is used to introduce explanations for each step of the wizard.
	Possible actions and hints are introduced with this symbol.

4 User Interface

4.1 System Menu

The System menu is located in the top right section of the user interface. It consists of the following menu items:

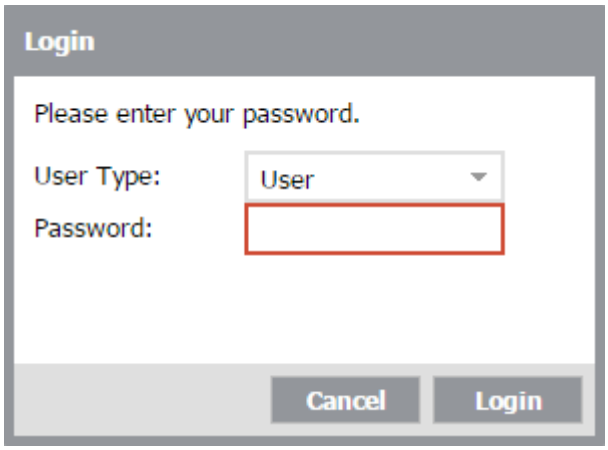
Menu item	Description
	User login
	Camera selection
	System settings
	Help system

4.1.1 User login

NOTE

The user management is disabled by default. It can be activated in "*System Settings -> User management -> Password protection active*".

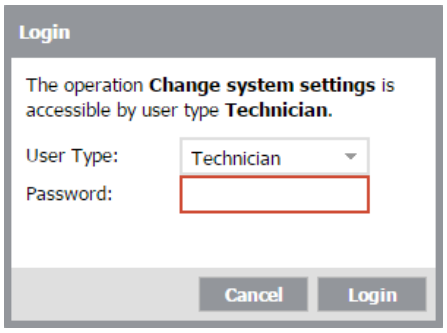
By clicking on user login you can authorize yourself.



There are four different users, each of them with different access rights:

Action	Required user level
View results	-
Start/stop inspection program	User, Technician, Expert, Admin
Save image	-
View inspection program	User, Technician, Expert, Admin
Modify tolerances	User, Technician, Expert, Admin
View LOG files	Technician, Expert, Admin
Trigger via user interface	Technician, Expert, Admin
Change to another camera	Technician, Expert, Admin
Modify inspection program	Technician, Expert, Admin
Modify system settings	Technician, Expert, Admin
Change inspection program	Expert, Admin
Change password	Admin

Whenever you execute an action you are not authorized to, a dialog will appear. In this dialog you can authorize yourself as a proper user to gain access.



Simultaneous access of multiple PCs

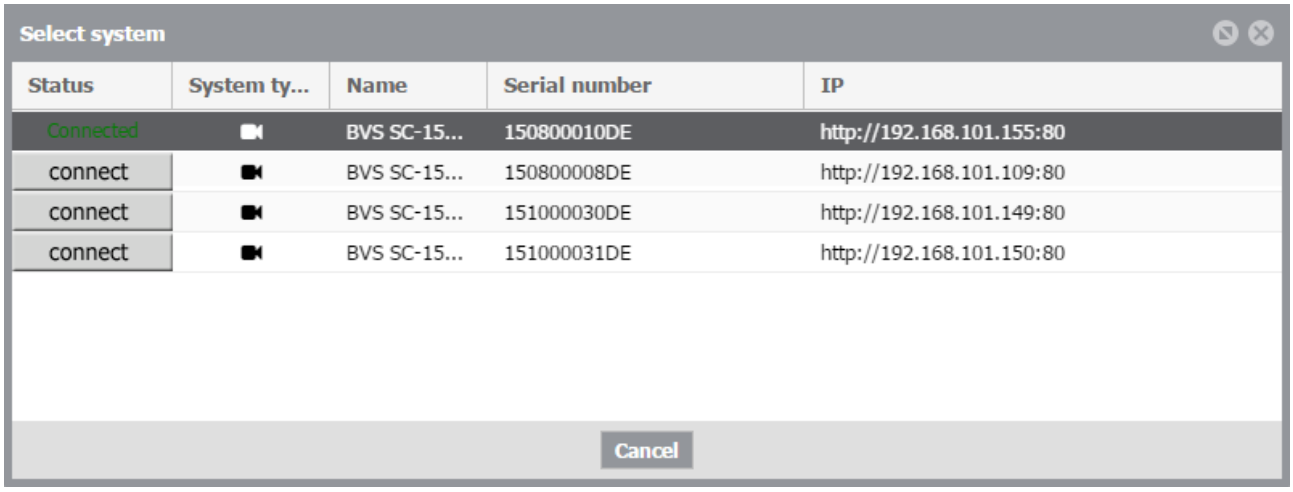
It is possible to access non-restricted areas such as the **monitor** section or the **statistics** section from multiple PCs at the same time. For this purpose, the icon of the User Login indicates the accessing users (here two users at the same time): 

In order to avoid simultaneous changes from multiple users only one user is allowed to be logged in at the same time. The user is logged out automatically and relocated to the **monitor** section if another user logs in.

If the user management is disabled the system also makes sure that only one user accesses restricted areas such as the **system settings** or the **configuration** section at the same time.

4.1.2 Camera selection

Camera selection enables you to connect with another camera or another PC system with mvIMPACT-CS software installed.



The dialog box shows all available devices. Note, that devices which are connected to the network but cannot be reached by the browser are not shown in this list.

The table comprises the following information:

Setting	Description
Status	Shows the status of the current device (connected). The "**connect**" button enables you to connect with another camera or PC system. When connecting, a new browser page opens, which displays the user interface for the device selected.
System type	Distinguishes between two categories of device: • A camera • A PC system with BVS Cockpit software installed
Name	Displays the name of the device, which can be freely specified. "BVS SC" is the default. You can use system settings to change the name of the device.
Serial number	Displays the serial number of the device.
IP	Displays the IP address of the device.

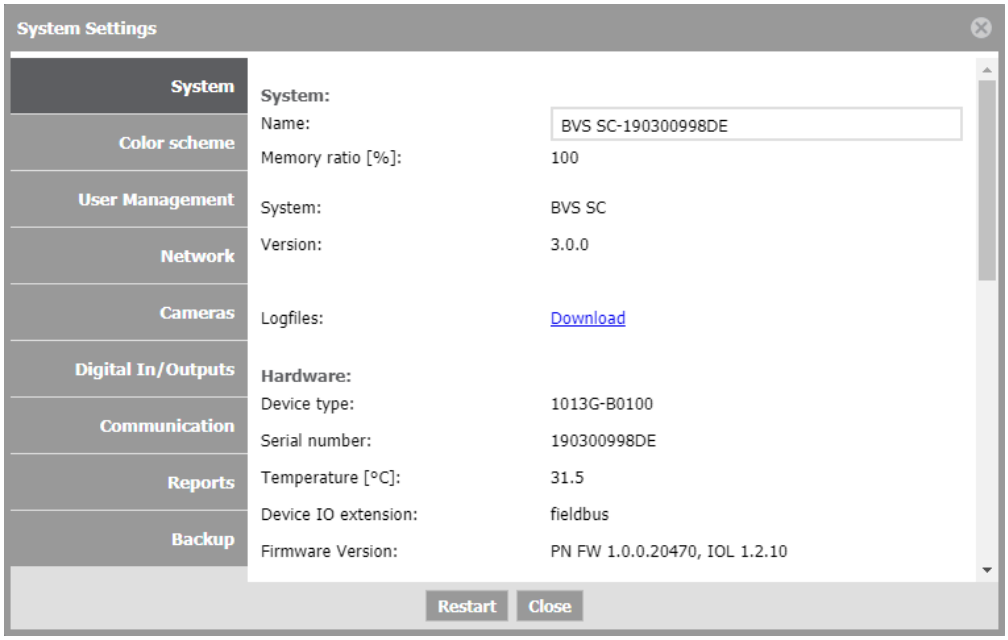
Not for SMART*CAMERA***:**

If the camera selection is empty, you have to install a tool which recognizes devices in a network automatically like Avahi or Bonjour:

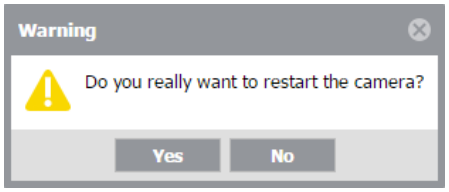
- http://avahi.org/
- https://developer.apple.com/bonjour/

4.1.3 System Settings

Information about the BVS Cockpit are summarized here. Additionally, the appearance of the user interface can be altered. The system settings are categorized as different Sub- Categories (System, Hardware, System Time, License).



You can restart der camera via "*System Settings*". After clicking the button "**Restart**", a warning will be displayed, whether you want to restart the camera or not:



As soon as you confirm with "**Yes**", the camera will restart.

System

According to the plattform, different subcategories are displayed.

System

This section contains important information about the system:

Setting	Description
Name	Customer-specific name under which this device is visible in the production environment. The name must only contain ASCII-conformant characters.
Memory ratio [%]	Portion of the available main memory (RAM) to be used for this BVS Cockpit instance. The total amount of 100% of available main memory will be shared between all BVS Cockpit instances. As the available main memory we define the memory that is not used by foreign software. The size of the available main memory is determined on startup and on every change of the inspection program.
System	The system BVS Cockpit is running on.
Version	The currently installed version of BVS Cockpit.
Logfiles	A link to the LOG files.

NOTE

The oldest inspection results will be automatically pruned if the used memory by the current instance reaches the specified ratio. If no free memory is available for the current instance, a warning message will be shown until enough memory is available or the memory limit is adapted accordingly.

LOG files allow to track changes and errors on the system. You can access the log files by clicking on the hyperlink "**Download**".

There are four kinds of log files:

- **User log files (user.log)**: These LOG files are user readable and allow to track all changes made in the system. They can be used to investigate which changes have been made to the inspection program or the system settings at which time. These files contain all modified parameters and settings and thus may contain sensitive information.
- **System log files (system.log)**: These LOG files contain error logs and crash reports and may help developers to track bugs and errors. Customer service may request access to them. These files don't contain sensitive information.
- **Client log files (client.log)**: These LOG files contain events and errors of web browsers used to access BVS Cockpit. They may help developers to track bugs and errors, as well.
- **Update log files (update.log)**: These LOG files protocol the installation of BVS Cockpit updates. They can be used to get information about the currently installed version of BVS Cockpit and potential errors during update to a new version.

NOTE

LOG files may also be accessed through the shared folders (e.g. *\\sc-150800015de\logs* using the **SMART***CAMERA*).

NOTE

In order to access this folder you have to use the following credentials:

- Login: *expert*
- Password: *expert*

NOTE

If you cannot access the shared folders, please have a look at the appendix "Troubleshooting Table" → "Accessing the shared folders is not possible".

Hardware

This section contains important hardware information of the device.

Setting	Description
Serial number	Serial number of the device (only on camera).
Temperature	The current temperature of the device. Will be updated when revisiting the system page. If the value 0 is displayed here, please execute the Inspection program in order to update the value.
Firmware version	Firmware version of the field bus device board (BVS SC).

System Time

The system time, time server and time zone can be set.

Setting	Description
Use NTP server	Determines whether a time server should be used to get the current time.
NTP server	The address of the time server to ask for the date and time. The time zone still has to be set manually.
Use time/date of the browser	Synchronizes the system time with the time, date and time zone of the currently connected client.
Time Zone	Specifies in which time zone the camera is located.

NOTE

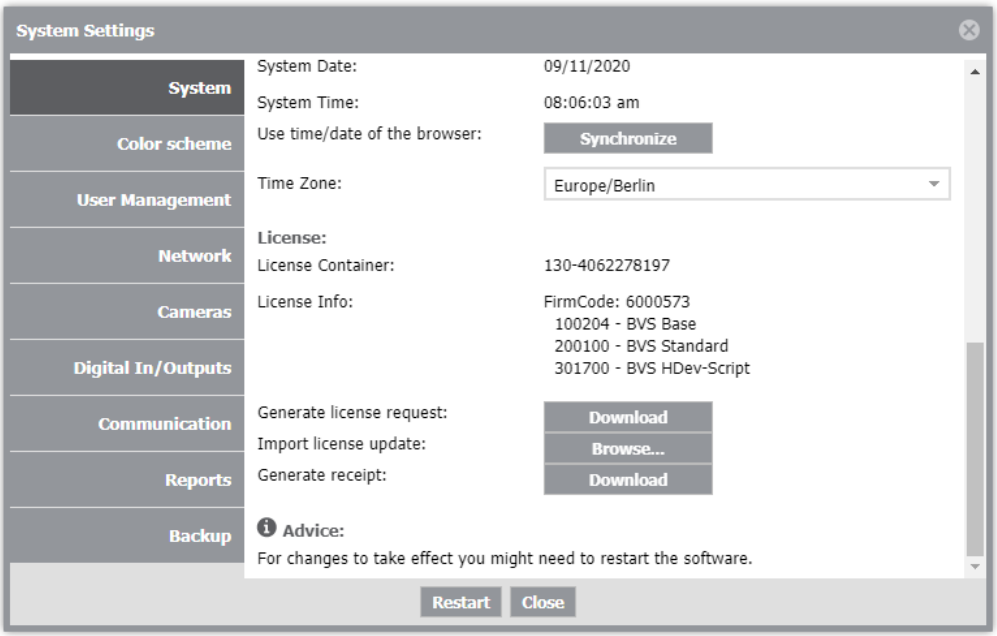
If the date of the camera has not been initialized it starts with 1.1.1970. As soon as a PC connects to the camera the date and time are adopted from the PC automatically.

NOTE

When using a NTP server the function *Use time/date of the browser* will only adjust the time zone, as the date and time are overwritten by the time server after a short time.

License

Here you can view licensing informations.



Setting	Description
License Container	The id of the license container of this Camera.
License Info	Shows a firm code and the associated product codes. These product codes determine the available tools and functions.
Generate license request (only for SMART*CAMERA*)	Generates and downloads a license request file.
Import License Update (only for SMART*CAMERA*)	Opload and import the licenses from the provided license update file.
Generate receipt (only for SMART*CAMERA*)	Completes the license update by generating and downloading the receipt file.

Upgrade license

You can upgrade your licenses at any time. Please visit our Balluff web shop or contact your Balluff sales person. A short manual about the upgrade process is available in the download section of the product page of the license upgrade.

The update procedures differ from product platform to product platform.

Color scheme

You can use the color scheme settings to adjust the appearance of the user interface. To activate the changes you have to activate the checkbox *Use Style*:

Setting	Description
Application name	The name of the software. If the field stays empty BVS Cockpit is used as name. The name must contain only ASCII-conformant characters.
Use Style	Activates or deactivates the usage of the customized appearance.
Menu headline	Defines the color of the upper blue headline.
Highlight	The highlight color. It appears also in the headline and is also used for activated buttons.
Left panel	The background color of the left area. This area contains the tools (**configuration** section) respectively the inspection results (**monitor** section).
Center panel	The background color of the center area in which the image is displayed.
Buttons	The color of (inactive) buttons. The color is also used for buttons as well as the title bar of dialogs and the result table.
Change logo	Activates or deactivates the usage of the customized logo. The logo is displayed at the upper right.
Company logo file	By clicking on the button you can select an image file on your local file system. The image is then uploaded to the camera and used as logo.
Logo size	This setting allows to scale the logo in order to fit into the headline.
Show splash screen at start	Activates or deactivates the splash screen when loading this page.
Show links to manuals	Activates or deactivates the button to the manual links in the help area on the right.

User Management

In the user management the password protection can be activated. If activated some areas such as the **configuration** section or the system settings require the user to log in.

NOTE

The chapter **user login** lists all possible actions for each user.

Setting	Description	Predefined password
Password protection active	Activates or deactivates the password protection.	
Password for 'User'	The password for user type *User*	*user*
Password for 'Technician'	The password for user type *Technician*	*technician*
Password for 'Expert'	The password for user type *Expert*	*expert*
Password for 'Administrator'	The password for user type *Administrator*	*admin*

NOTE

All passwords must contain only ASCII-conformant characters.

NOTE

With activated password protection only the user *Administrator* is able to access the *User Management* page.

Network

All network settings of the camera can be configured here. A reboot of the camera is required in order to apply the changes.

Setting	Description
Obtain an IP address automatically	Specifies whether the IP address of the camera will be set dynamically via DHCP or statically.
IP	If the IP address is static, it can be set here.
Gateway	If the IP address is static, the gateway can be set here. If the gateway stays empty the camera may only be accessed within the same subnet.
Port	Defines the port BVS Cockpit is listening on. The default is 80.
Activate second network interface	Enables connections to BVS Cockpit Webinterface vie Fieldbus network port. If this option is deactivated there is no network connection between Fieldbus and Gigabit port.
Use Avahi/Bonjour	If activated, cameras are automatically identified in within the network

NOTE

The default Gateway is responsible for routing to and from other networks. Even if both network ports are active, usually only one gateway address is required.

NOTE

Use different subnet masks, when using both network interfaces (Gigabit Ethernet and Fieldbus). Otherwise you can lose data packages because of unclear routing.

NOTE

Only for SMART*CAMERA*:

It is recommended that the remote peer's network interfaces (PC or switch) are configured to use **auto-negotiation**. If a fixed network speed is setup, then **half-duplex** mode should be chosen.

Cameras

Here logical cameras which may be selected in tool **"Set up camera"** can be mapped to physical cameras for image acquisition.

NOTE

This category is reachable by keyboard shortcut STRG+ALT+L as well

Einstel-lung	Beschreibung
Name	A user defined name to describe this logical camera, which can be used in tool **"Set up camera"**. The name *Default Camera* for the first logical camera cannot be altered.
Camera	Physical camera that is mapped to a logical camera. All connected cameras are available in a drop down list. Alternatively, a camera simulation ("File Device") can be selected to display stored images. When using the camera simulation ("File Device"), the inspection changes into a "**Simulation Mode**" state. This state is reflected in both, the menu bar of the browser application and in the status information being sent. For leaving the simulation mode, select a physical camera and deactivate the entry "**Simulated inputs and data**" in the communication settings of the system settings, if applicable. Cameras that are already mapped to another logical camera are not available. Cameras that are used by a different instance of BVS Cockpit are labelled as *blocked* and cannot be selected. **NOTE** Cameras are also labelled as blocked if they are used by another instance of BVS Cockpit as I/O devices. If necessary, deallocate the camera in category **Digital In/Outputs** also.
Type	Type of the selected physical camera.
Delete	Deletes the logical camera. The associated physical camera is deallocated and can now be mapped to a different logical camera.
Add camera	Adds an additional logical camera.
Update	Updates the list of available physical cameras.

NOTE

If the connection to the camera that is currently in use camera is lost, a dialog box is displayed in which the user may choose whether to stop the inspection processing or just pause it until the camera is reconnected. Additionally, the status System Error is set and thereby external device are informed about the loss of the camera connection (see chapter "Connection to the Customer Control System" and the following section "Digital In/Outputs").

In configuration mode loss of the camera is indicated by a result message in tool **"Set up camera"**.

Digital In/Outputs

Configurations for signals used by the inspection program mapped to the available digital in/outputs of the hardware to allow proper control of attached devices.

Setting	Description
I/O type	Defines whether it is an input, an output, the flash line, the trigger line, or a status information signal. Inputs can be read using the "**Get inputs**" tool. Outputs can be set using the "**Set outputs**" tool. Advanced settings regarding the flash and trigger can be made in the "**Set up camera**" tool. Details about each status information signal are listed in the table "Structure of status word" in chapter "Connection to the Customer Control System".
Name	A user defined name to describe this configuration which can be used in the above mentioned tools to address this In-/Output. Note, that the name of status information signals cannot be altered.
I/O device	Hardware device which is used for this In-/Output. (e.g. the active camera) I/O devices that are used in a different instance of BVS Cockpit are labeled as *blocked* and can not be selected. **NOTE** Cameras are also labeled as blocked, if they are mapped to a logical camera in a different instance of BVS Cockpit. If necessary, deallocate the camera in category **Cameras** also.
Line	Hardware line of the device which is used for this In-/Output/status information signal.
Inv.	If activated, the in/output/status information signal gets inverted. Inverted in/outputs/status information signals are interpreted as active on low voltage.
LED	If activated, the state of the in/output/status information signal is shown with the camera LED (only supported on **SMART***CAMERA*).
Duration	Defines the pulse duration of an output in milliseconds. The default value of 0 means, that the output is active until another value is set manually using the "**Set outputs**" tool. Choosing pulse durations larger than zero the output is active for the defined time span and retransferred into its inactive state afterwards.
Delay	Defines the delay of the trigger or an output in milliseconds. The default value of 0 means, that the trigger will start without delay or rather will the output toggle when the tool "**Set outputs**" is running. Choosing delay larger then zero the frame acquisition will be delayed or rather the output will be delayed with respect to the of frame acquisition.
Reconfigure I/Os *Expert parameter*	Each modification of parameters is generally applied to the digital in/outputs of the hardware immediately. Just to be safe, this button causes a retransmission of the complete set of parameters to the hardware.

NOTE

For the I/O type, all status information signals can be selected as listed in table "Structure of status word" in chapter "Connection to the Customer Control System". The sole exception being the status information signal "Input Container".

NOTE

When switching to the File Device the simulation mode has to be activated if the inspection program contains the tools "**Get inputs**" or "**Set outputs**". This is signaled via a Pop up Box. You can turn the simulation mode on and off in the Communication section of the system settings.

Complex tool connection stay intact when switching between cameras.

Loading inspection programs with the tools "**Get inputs**" or "**Set outputs**" which uses In-/Outputs configurations not present in the current list results in adding the missing

configurations to the list. Same for the signals "trigger" and "flash" of the tool "**Set up camera**". The device and line have to be configured manually for the automatically added configurations.

If no free line is available for a device configuration, another configuration has to be deleted or the simulation mode has to be activated to start an inspection program with the tools "**Get inputs**" or "**Set outputs**".

NOTE

The system allows to assign pulse durations or delays different from zero to the outputs. Depending on the used camera, the assignments can be limited. As soon as you reach the assignment limitation, the other fields for pulse durations and delays will become disabled.

Outputs with a delay of 0 are set in the moment the tool "**Set Outputs**" is executed. If the delay is greater than zero the output is set delayed relative to the moment of the image acquisition.

If the tool "**Set up camera**" uses "level" as trigger mode then a delay will be ignored.

There may appear undesirable effects if the sum of duration and delay exceeds the calculation time of an inspection result.

There may appear undesirable effects if the delay is larger than 0 and the next trigger arrives before the pulse of a output is finished.

Additional timing information can be found in chapter "Connection to the customer control system".

Communication

The communication settings allow to setup how to communicate with plugged devices such as PLCs or control PCs. These settings affect all data that is send with the "**Send results**" tool to these device types.

Setting	Description
Mode	The communication interface to use: • Fieldbus • UDP • TCP • Serial (nur **SMART***CAMERA* Lite)
Data byte order [Fieldbus]	The composition of integer data (little or big endian, not available for all communication interfaces)
Port [UDP, TCP]	The UDP/TCP communication port.
Send status information	Enables sending of status messages.
Status	Shows the current state of the communication.
Simulated inputs and data	Activates a mode, where simulation data can be specified for the digital inputs and the received input data. This state is reflected in both, the menu bar of the browser application and in the status information being sent. For leaving the simulation mode, deactivate the entry "**Simulated inputs and data**" and change to a physical camera in the "**Setup camera**" tool, if applicable.
Status fieldbus [Fieldbus]	Shows the current state of the connection.
Overflow result buffer to PLC [Fieldbus]	Indicates an overflow in the result buffer to the controller.
IO-Link device class	Shows what sort of IO-Link device is connected to the IO-Link master port
Fieldbus	Specific informations about the Fieldbus configuration oft the device (additional information is provided in the hardware manual).
Profinet	Specific informations about the Profinet configuration oft the device (additional information is provided in the hardware manual).
IO-Link Mode	Defines the mode of operation for the IO-Link master port (additional information is provided in the hardware manual).

Reports

The reports settings allow to define which data the reports will contain and to set up where the reports are stored on.

The reports are saved as XML files on the server under a folder with the same name as the inspection program. These XML files can be viewed with the help of a web browser. The visualization can be modified by editing the XSL file "style/report.xsl" and the CSS file "style/report.css". If a folder named "style" already exists under the given root path, this folder will be copied instead of a new one being automatically generated. For further information regarding XSL and CSS files, please consult other sources.

NOTE

Reports are only created in monitoring mode.

Connection settings

Setting	Description
Protocol	Protocol to be used to transmit report data. See table below for a list of supported protocols.
Server	The IP address or the URL (both starting with ftp://) of the server to store the reports on. E.g.: *//ftpserver/reports/* or *//192.168.1.5/reports/*
User	Name to be used for authentication on the server or network share. When specifying a domain user for authentication over SMB, the domain has to also be specified, i.e.: *domain/user*
Password	Password to be used for authentication on the server or network share.
Allow self-signed certificates (unsafe)	The FTP connection is still encrypted, but the identity of the sever is not validated. Weakens the security of the connection, but allows the use of self-signed certificates.
Path	Folder to be used for report storage. E.g.: *R:\Reports* or *\\server\Reports*.
Connection	Starts a connection test to the server using the settings defined here.

NOTE

To set up a server or network share to record the reports, please consult your IT department or external sources.

Standard reports

Standard reports are generated after every inspection, provided any of the options below is configured and the inspection result (OK/NOK) matches the reports configuration.

Setting	Description
OK inspection results	Determines whether results of successful inspections are stored as XML reports
OK images (JPG)	Determines whether images of successful inspections are stored as JPG files
OK images (RAW)	Determines whether images of successful inspections are stored as BMP files
NOK inspection results	Determines whether results of failed inspections are stored as XML reports
NOK images (JPG)	Determines whether images of failed inspections are stored as JPG files
NOK images (RAW)	Determines whether images of failed inspections are stored as BMP files

Periodic reports

Periodic reports are generated after the first inspection since start, followed by every x-th inspection as specified by the options below. The inspection result (OK/NOK) does not affect whether reports are generated or not.

Setting	Description
Interval	Sets the inspection interval (between 2 and 1,000,000) at which to create a report
Inspection results	Determines whether results of and inspection are stored as XML reports
Images (JPG)	Determines whether images of an inspection are stored as JPG files
Images (RAW)	Determines whether images of an inspection are stored as BMP files

NOTE

A report can be dismissed, if the connection to the server is too slow. For possible solutions see the "Troubleshooting Table" of the respective hardware manual.

Naming scheme

Setting	Description
Imprint filename into image	Determines whether the filename of the image will be imprinted into the image.
Filename text constant	A constant string which can be used for creating the report and image filenames. Valid characters are "*a-z*", "*A-Z*", "*0-9*", "-", and "_".
Filename components 1-4	Defines the inspection result data the filenames of the report and image files are consisting of.

NOTE

Invalid characters in the filename are replaced by the underscore character "_".

NOTE

For enabling an inspection result to show up in the lists of filename components, it has to be marked in the first column of the result table. If the *System Settings* window is opened and the tab Reports is selected at this moment, another tab has to be chosen. When returning to the tab Reports, the lists of the filename components are updated.

NOTE

Google Chrome and other Chromium based Browser forbid loading of local websites because of security concerns. This could lead to reports being displayed incorrectly when opened in those browsers. Please use a different browser or a dedicated XML viewer to open report files. The following browsers were tested as working: Mozilla Firefox (up to version 60.6.1esr) , Microsoft Edge (up to version 42.17134.1.0), Microsoft Internet Explorer (up to version 11.648.17134.0).

Supported Protocols

Protocol	Description	SMART*CAMERA*	SmartVisionController	PC version
FTP	Enables saving reports to an FTP server.	✓	✓	✓
FTPS	Enables saving reports to an FTP server. The connection is encrypted using SSL/TLS.	✓	✓	✓
SMB	Enables saving reports to an SMB network share. The minimal version supported is SMB 2.0	✓	✓	✗
Path	Enables saving reports to an arbitrary location on your file system.	✗	✗	✓

NOTE

The FTPS protocol requires a correctly configured FTP server and current CA certificates.

Not for SMART*CAMERA*:

An update of the bundled CA certificates may be required. You can find these in your installation directory in the *cacerts* subfolder under the name of *cacerts.pem*. An up-to-date bundle is generated by the developers of libcurl from the data provided by Mozilla and provided free of charge under https://curl.haxx.se/docs/caextract.html.

Backup

Here you can create or restore a backup.

Creating a backup

By pressing the "**Download**" button a backup of the instance the user is working with is created and downloaded to the client PC.

The backup contains the following:

- firmware package for **all** instances of BVS Cockpit on the device
- system settings of the **current** instance
- all inspection programs of the **current** instance

This means that on systems with multiple instances **each instance has to be backed up separately**.

The default filename consists of the following:

- date and time (*yymmdd-hhmmss*)
- name of the platform
  - *BVC-SmartCamera* for **SMART***CAMERA*
  - *BVC-PC* for BVS Cockpit
  - *BVC-Controller* for Balluff SmartVision Controller
- interface
  - *IO*
  - *Fieldbus*
- Port number (not for **SMART***CAMERA*)
- BVS Cockpit version number

The backup file can be renamed as needed.

NOTE

Backup creation might fail if any inspection program contains file names with unsupported characters. Only the following characters are allowed in file names:

A-Z a-z 0-9 .,_-!#$%&'()+;=@[]^`{}~ and the space character.

Restoring a backup

By pressing the "**Select file**" button, you can select a previously created backup of an instance to restore.

ATTENTION

Loss of data

Make sure that a backup of the current configuration exists before restoring a backup.

Not for SMART*CAMERA*:

This also applies to all other instances of BVS Cockpit on the same PC or SmartVision Controller.

ATTENTION

Loss of data

The restore operation **can not be cancelled** after it has been initiated. Ensure the connection between the server and the client **does not get interrupted**, i.e. due to a network error, by closing the browser window or reloading the tab. Otherwise the BVS Cockpit instance might be left in an inconsistent state.

ATTENTION

Loss of data

Not for SMART*CAMERA*:

If you have multiple instances running on the same PC or SmartVision Controller, please be aware that the restore operation will

- replace the software version of **all instances** by the version included in the backup (the compatibility with the inspection programs of the other instances can not be guaranteed),
- have to **terminate** all other instances in order to complete the process,
- only restore the data **of the instance included** in the backup,
- **irreversibly overwrite** the data of the currently active instance.

NOTE

BVS Cockpit Windows only:

Restoring a backup requires for the included BVS Cockpit version to be installed. This requires administrator level access on the server PC. Since the elevation prompt can not be confirmed automatically due to security reasons, this step has to be done manually. For this reason, restoring a backup is only possible from a local client.

NOTE

The backup can only be restored by an BVS Cockpit instance running on the same platform it was created on. This means that restoring a backup created on an **SMART***CAMERA* on a PC or SmartVision Controller is not possible.

4.1.4 Help system

The integrated help system provides summary information on the respective elements of the user interface.
The button "**Manual**" opens the detailed manual in PDF format.

4.2 Action menu

The Action menu provides access to three action areas and can be found in the top left-hand section of the user interface:

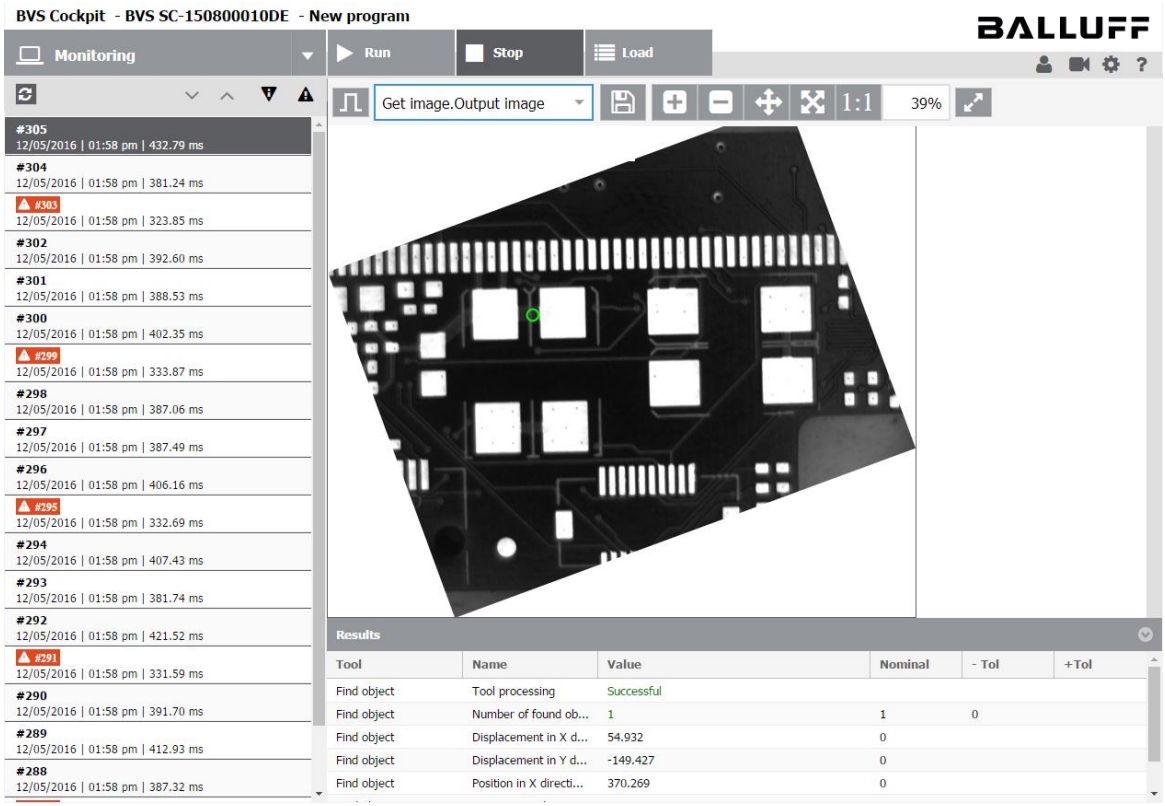
- **Monitoring** shows the inspection results. This is also the default page that is displayed automatically when connecting.
- **Configuration** enables you to create and manage inspection programs.
- **Statistics** shows a statistical analysis of the inspection results.

When starting, BVS Cockpit opens the last inspection program opened. The information displayed in all three action areas refers to the inspection program that is currently open.

4.2.1 Monitoring - display inspection results, open and create new inspection programs

You can use **Monitoring**

- to display the actual results for the current inspection,
- to open inspection programs, and
- to create new inspection programs.



The menu items have the following functionalities:

- **Run:** starts the inspection program. The results are stored in a ring buffer in main memory (RAM).
- **Stop:** stops the inspection program.
- **Load:** opens up the manage inspection programs dialog.
- **Trigger** ⎍ **:** Causes a trigger.

Overview of inspection results

The table on the left shows the list of inspection results. Each inspection results consists of

- a result number,
- time stamp of image acquisition and
- execution duration (the execution duration does not comprise time that is spent on waiting for a trigger, camera exposure time or writing reports to a server)

Inspections results are highlighted with a red exclamation mark, whenever a tool, which affects the inspection result, Failed (bad image).

As soon as you select an entry, the associated image and its detailed results are shown in the results table at the bottom of the screen. In the **configuration** section you can specify which detailed results have to be shown here. Note that graphical elements such as lines, regions, or rectangles are visualized in the image area only, but not displayed in this result table.

When selecting the button, the refresh of the table on the left will stop at the same time. You can stop the refresh of the table on the left by clicking on . With this button you can also activate the refresh functionality again. Therefore with this button it is possible to check single inspection results without disappearing from the list and while the inspection is still running.

With the buttons / you can navigate to the next / previous inspection results page. With the buttons / you can navigate to the next / previous inspection result with a bad image.

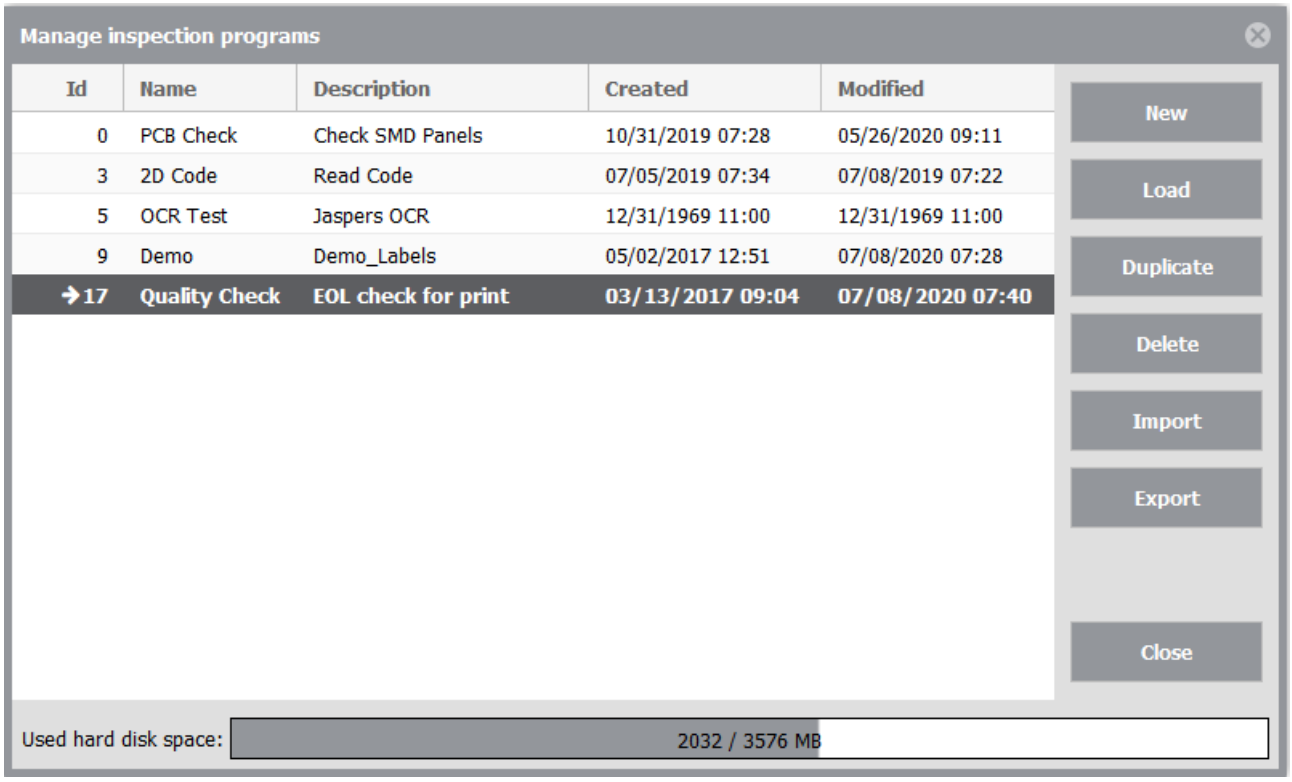
NOTE

All inspection results are stored in a ring buffer in the main memory (RAM). After rebooting these results are lost. Please use the function *reports* in order to persistently store the results on a FTP server.

The number of buffered inspection results (typically several hundreds) depends on the number of selected result outputs. Due to memory limitations images may not be available for all inspection results especially on a SMARTCAMERA. For older inspection results only result values are available but no images. Saving images in BMP format is only possible for the latest OK images and the latest NOK images, whereras 4 times more NOK images are available as OK images. On a SMARTCAMERA with grayscale sensor 32 respectively 8 images are available. For large inspection program on the SMARTCAMERA it can be even less whereas on platforms with more main memory a lot more images in BMP format are available.

Manage inspection programs

By clicking on the button Load a dialog opens up where you can manage the inspection programs on the camera.



The arrow marks which inspection program is currently loaded. The values of the fields *Id*, *Name* or *Description* can be modified by double-clicking on them. By using the *Id* a connected PLC is able to identify and load an inspection program. The *Name* of an inspection program helps the user to identify the inspection programs and is also shown in the **Configuration** section. The *Description* is optional and can be used for to place detailed information about the inspection program.

The available hard disk space is displayed at the bottom. A warning is shown when the free disk space falls below 100 MiB. If the free disk space decreases even more, additional warning messages are displayed. No more warnings will be displayed as soon as the free disk space exceeds 100 MiB.

NOTE

Only Id 0 can be used for multiple inspection programs at the same time. The remaining Ids can only be related to a single inspection program.

These actions can be executed in this dialog:

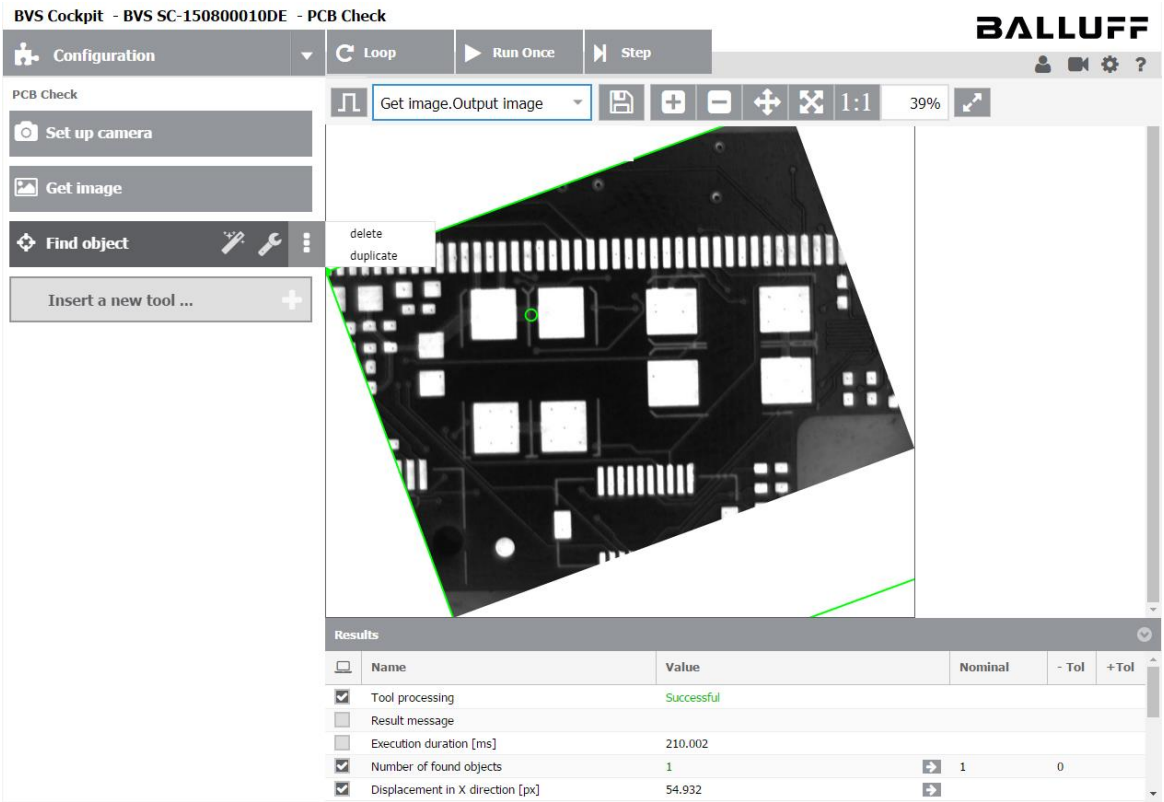
Action	Description
New	Create a new inspection program. The currently loaded inspection program will stay loaded.
Load	Loads the selected inspection program.
Duplicate	Duplicates the selected inspection program.
Delete	Deletes the selected inspection program.
Import	Imports an inspection program as a ZIP file. The currently loaded inspection program will stay loaded.
Export	Exports the currently selected inspection program and stores it on the local PC. **NOTE** File names containing a UTF-8 symbol are not supported.

NOTE

The inspection program is saved automatically after each modification.

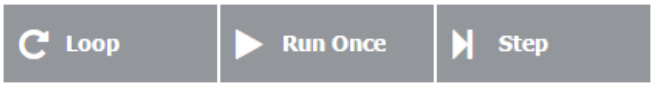
4.2.2 Configuration - create an inspection program

You can use **Configuration** to create inspection programs.



Understand and operate layout and control elements

Use the menu items to control inspection program progress.



These have the following functionalities:

- **Loop:** the program takes images continuously and runs through to the selected program step. If a trigger mode has been selected in "**Set up camera**", it is waiting for a trigger event.
- **Single run:** takes a single new image. In this case, the program also runs through to the selected program step. A new image is taken regardless of the trigger mode in "**Set up camera**". If the inspection program contains multiple program steps "**Get image**," an image is acquired in each of them.
- **Step:** serves to run through all inspection steps consecutively, one-by-one. A new image is taken at the program step "**Get image**".
- **Trigger** ⎍ **:** Emits a trigger if the **Loop** is activated and a trigger event is waited for. This causes an image to be acquired. If the inspection program contains multiple program steps "**Get image**," an image is acquired only in the step which currently waits for a trigger.

Underneath the menu bar, the column on the left shows the list of the individual program steps for the current inspection program. You can select a program step by clicking it with the mouse. If a step takes significant time for execution, this is visualized during its execution by replacing the tool symbol with a wait symbol.

The column on the right is divided into two sections. The upper section shows the **resulting image for the selected program step**; the lower section presents the **results of the selected program step**.

Adapt the display of the resulting image

Above the resulting image, you can use the zoom menu to specify the respective zoom factor, which is shown on the right (in this screenshot, this factor is "71%").

Icon	Description
	Zoom in on the resulting image.
	Zoom out on the resulting image.
	Automatically scale the resulting image to fit the space available on the user interface. If the "**Find object**" or "**Read Code**" tool is used, the image will be rotated, so that the object is always seen in the same orientation. Furthermore it will crop the image, so that the scale factor will remain the same.
	Automatically scale the resulting image to fit the space available on the user interface. If the "**Find object**" or "**Read Code**" tool is used, the image will be rotated, so that the object is always seen in the same orientation. Furthermore it will **not** crop the image, so that the scale factor will be adapted continuously.
1:1	Display the resulting image with pixel precision, this corresponds to a zoom factor of 100%.
	Save the image currently displayed, as long as the raw data for the image is still on the server. In the following dialog, the file name of the image to be saved can be defined. Furthermore, this dialog also shows the destination path where the image is stored. To access the images, open Windows Explorer and enter this path in the address bar. The default directory for saved images is the same as the one the "*File Device*" uses to read its images.
	Hides the online help and the result table. Enlarges the result image. Press again to display the result table and the online help.

You can adjust the size ratio between image and result table by moving the dividing line, or collapse/expand the result table.

NOTE

Please use only the buttons in the zoom menu of the resulting image to zoom display in or out. Using special browser resizing commands (such as CTRL and Plus/Minus keys) may result in unwanted display depending on the browser you are using.

Underneath the image, the pixel value below the mouse cursor is displayed. For color images, the pixel value consists of a hue, saturation and intensity triplet (HSI), for gray images, the value denotes an intensity value (I).

Work with the results

In the column headed , you can select the results that you want to display in the **Monitoring** section, collect statistics for and include into reports. Displaying results in the **Monitoring** section might cost additional runtime.

NOTE

Floating point values are always displayed with three digits after the decimal point and if necessary rounded to three digits. You can enter and confirm more precise values. These values will be used internally.

Modify tool tolerances

For some tool results, you can define tolerances. Tolerances are shown in the result panel at the bottom of the screen. If the current value is within the allowed range, it will be displayed in green color. If the value is out of range the tool processing will fail and the value will be displayed in red color. Additionally, the result message points at the reason.

Results

	Name	Value	Nominal	- Tol	+Tol
☐	Tool processing	Successful			
☐	Result message				
☐	Execution duration [ms]	19.341			
☐	Edges used for the focus calculation				
☑	Area of interest				
☑	Calculated focus value [%]	74.51	50	0	

The allowed value range can be modified by clicking on the values in the columns "**Nominal**", "**-Tol**" and "**+Tol**". Only positive values may be entered. To completely remove the upper or lower limit, the contents of the fields can be removed. E.g. by setting *Nominal* to **50**, *-Tol* to **0** and removing the contents of *+Tol* all values greater than 50 are accepted. Thus no upper limit is defined and values up to infinity are accepted.

To completely remove the tolerance and accept arbitrary values the contents of the field "*Nominal*" can be deleted. By entering a valid value to this field the tolerance can be reactivated.

By clicking the button → the Nominal value is set to the current value.

Create new inspection program

To create a new program, click on the **Load** button in the **Monitoring** section and then click on "**New**". Then load the new program by clicking the button **Load** and switch back to **Monitoring**.

Each new inspection program created consists of the elements

- "**Set up camera**",
- "**Get image**" and
- "**Insert a new tool …**"

"**Set up camera**" and "**Getimage**" are **Tools** - that is to say, individual tools that are needed for an inspection program. All tools are available in a "**Tool Box"**. The functionality of individual tools is explained in the **"Tool Box**" section.

The tools are used in the order in which they are listed. In the process, output data is generated that can be used in subsequent tools. A tool cannot access data from a subsequent tool.

Whilst setting up a program step, this is executed immediately. If a program step is selected, it is placed into setting mode and immediately shows the effect of the setting. Tools, such as "**Set up camera**" or reading digital input signals also operate in live mode, as the result can be affected by external influences, such as camera positioning or lens settings. Other tools are recalculated as soon as parameters are changed.

NOTE

Floating point values are always displayed with three digits after the decimal point and if necessary rounded to three digits. You can enter and confirm more precise values. These values will be used internally.

Insert other tools

You can use the "**Insert a new tool …**" button to insert a new tool:

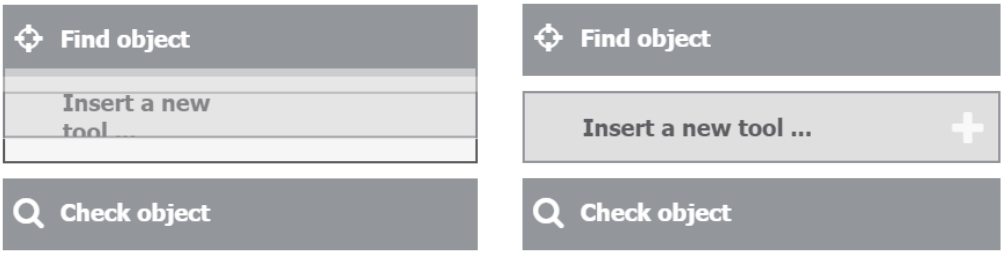
The ✚ button opens the "**Tool Box**", which lists all the tools available for you to insert.

The new program step is inserted at the current position of "**Insert a new tool …**". Therefore, make sure you start by placing "**Insert a new tool …**" in the required position.

Whilst inserting a program step, this is executed immediately.

Move program steps

You can use Drag & Drop to move program steps:



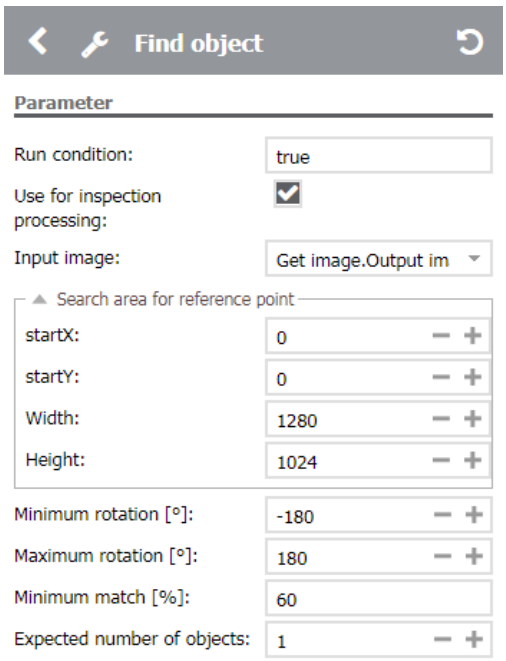
1. To do this, use the mouse to click on the required program step and
2. hold the mouse button as you drag the required element to the required position.
3. You can then release the mouse button.

Tool configuration using the wizard

Many tools have a wizard ; the questions posed by the wizard and the answers to them provide an easy way of setting up the program step. After a new tool has been inserted, if it has a wizard, this opens automatically.

Modify tool settings

A click on the button opens up the settings of the selected tool an allows to modify them.



Even the name of a tool can be modified after clicking on the text in the header.

As soon as a tool outputs a warning, for example if a value is out of range, the header will be displayed in yellow.

As soon as a tool outputs a failure, the header will be displayed in red.

Link data between tools

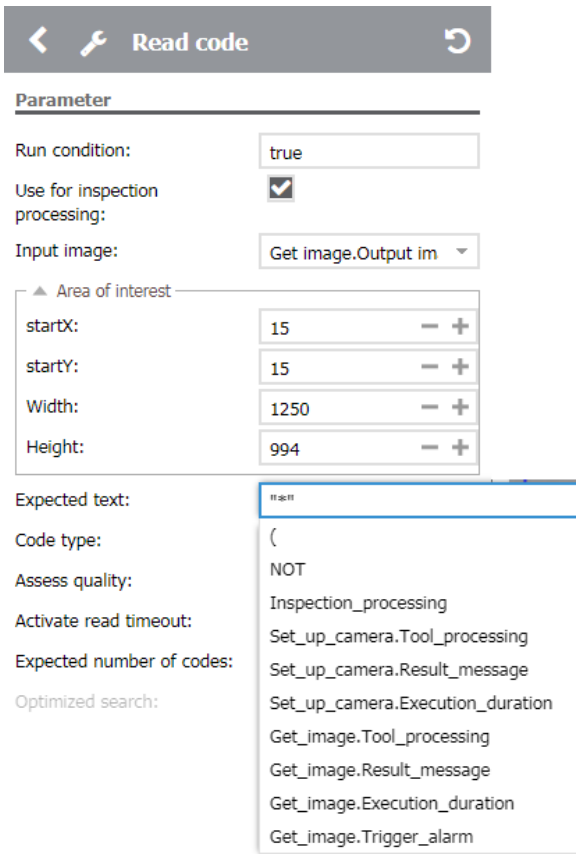
In most cases the settings of a tool are static values that are only manually changed by the user.

However, in many cases, it makes sense to use results of other tools earlier in the process (known as "data links"), rather than statistical values. The settings are then changed dynamically for each program sequence.

To link a setting with the result of a previous tool, you first need to select the tool required and use the button to switch to the tool settings. Then click on the text field for the setting that you want to link and remove all of its contents. A drop-down selection box will appear, which shows the possible results for linking. In this context, *Inspection processing* represents the aggregated result of a current inspection program run, taking the results of all tools into account which have been passed so far. The aggregated result is the combination of each available single tool result and is provided as an input for each tool. Choose the proper result to link to.

NOTE

An error message will occur below the text field if a result is linked to an incompatible data type.



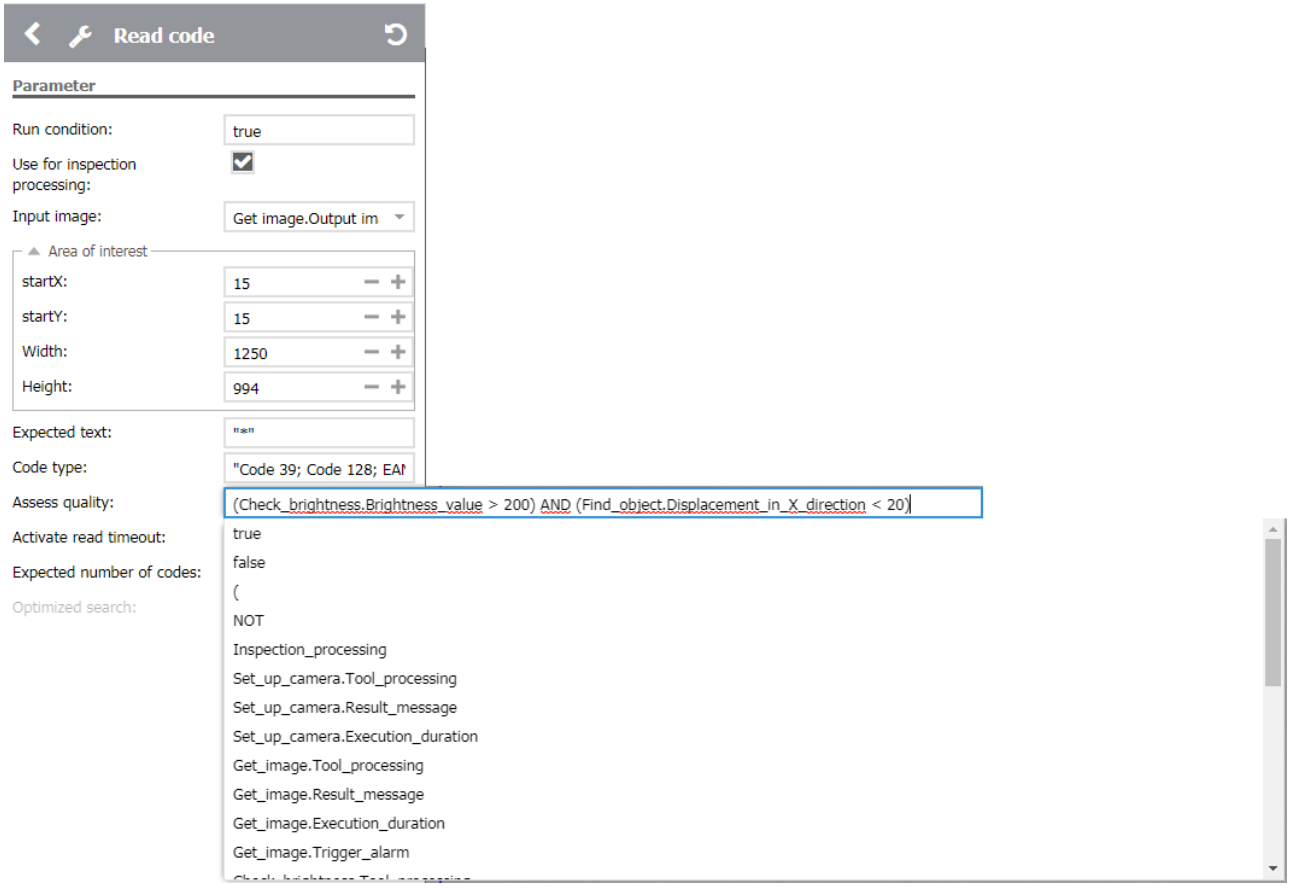
Some settings are not specified in text form and must have a link, such as, for example, the input image to be used. Mandatory parameters are highlighted in red, if they have not yet been linked. The application attempts to set up these links automatically. Nevertheless, you may still need to make manual changes to the links.

Use formulas

Most text fields do not only allow using static values or links but also complex formulas. Possible expressions of a formula are:

- All results of predecessor tools e.g. *Get_image.Tool_processing*
- Mathematical operators: +, -, *, /

- Logical operators: =, <, <=, >, >=, *!=, AND, OR, !, NOT*
- Brackets: *(,)*



Auto-completion supports in typing the formula by suggesting inputs depending on the context. The auto-completion opens up and updates automatically after typing a blank.

When entering a formula it has to be considered that the resulting type of the formula fits the data type of the text field, e.g. it is not possible to assign the field "*Expected text*" a formula that calculates a number. In case of an error there is an error message pointing to the reason such as incompatible types or syntax error.

NOTE

When using formulas it is possible to assign a floating-point number to an integer. In this case the number gets rounded.

Further options

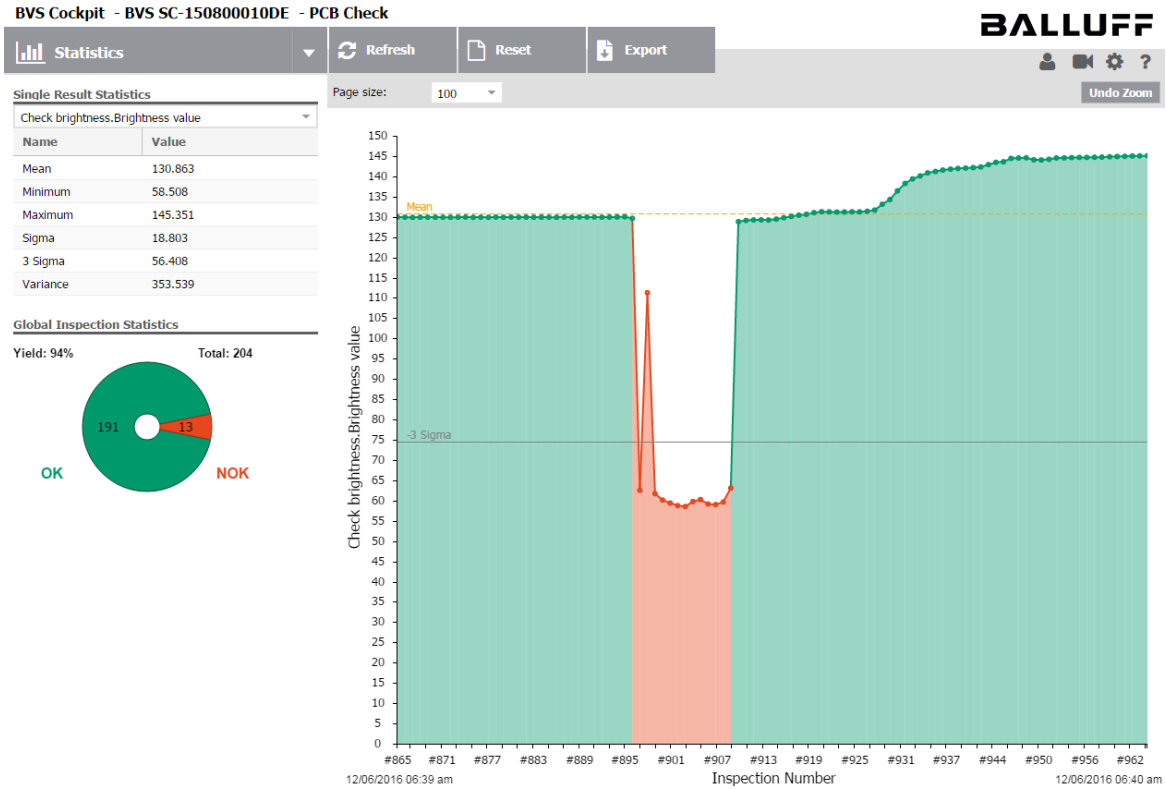
⋮ Opens a context menu with further options for the selected tool.

- delete:
  - Deletes the selected tool.
- duplicate:
  - Creates a copy with all trainings data and settings of the selected tool and adds it after the tool. Data links are preserved by this operation.

4.2.3 Statistics - analyze inspection results

The **Statistics** section displays ongoing statistics for inspection reports, enabling data to be analyzed.

> **NOTE**
>
> The visualized statistics data always relates to the currently loaded inspection program.



Single Result Statistics

In this area you can choose a result to analyze. The list contains all numeric results that are shown in the **Monitor** section.

> **NOTE**
>
> There are no results to choose from, if the inspection program has never been started.

The following statistics are available for the chosen result. They relate to all inspection program runs since creating the inspection program respectively since the result was chosen to be watched in the **Configuration** section.

Name	Description
Mean	Average value of the result
Mini-mum	Greatest occurring value of the result
Maxi-mum	Smallest occurring value of the result
Sigma	Standard deviation respectively average error of the result
3 Sigma	Three times the standard deviation. 99,7% of the values are within the range [mean - 3 Sigma, mean + 3 Sigma]
Variance	The spreading of the values. If the values are similar the variance is small. If they are widely spread the variance is big.

Global Statistics

The global statistics show how often the inspection program has been executed. It also shows the yield and the number of successful and failed inspection results.

Statistics diagram

Visualization

The statistics diagram shows the development of the selected result. Additionally, to the value both the mean value and the thresholds - 3 sigma and + 3 sigma are visualized.

The color of the data point and the color of the line both indicate whether the value is within its tolerance range. The color below the line shows whether the whole inspection result was successful (green) or failed (red).

NOTE

If the inspection program has not been executed since it has been loaded there are no values to visualize in the statistic diagram.

Interaction

You can use the "Page size" to choose whether to visualize the last 100, 200, 500 or 1000 results.

When moving the mouse on a data point in the diagram it shows the value, timestamp, and marked with colors whether the inspection result was successful in this run.

A double-click on a data point switches to the **Monitor** section and shows all details of this inspection result.

To zoom within the diagram drag a box around the area to magnify. Use the button "**Undo zoom**" to demagnify again.

Refresh

Using the button *Refresh* updates all three areas of the **Statistics** section.

Reset

When using the button Reset all statistics data of this inspection program is reset and all results in the **Monitor** section are deleted.

Export

When using the button *Export* all numerical results and statistics of the current running inspection program are downloaded as a CSV file.

NOTE

The export will only provide the results, which have been generated since the last load of the inspection program. This is the difference to the "Single Result Statistics" part, which refers to the complete lifetime of the inspection program.

5 Tool Box

5.1 Image capturing tools

5.1.1 Set up camera

Task

Used for selecting the camera and setting camera parameters.

Wizard

A camera calibration enables the system to transform pixel into world coordinates. In this way it is possible to carry out measurements or positioning in mm.

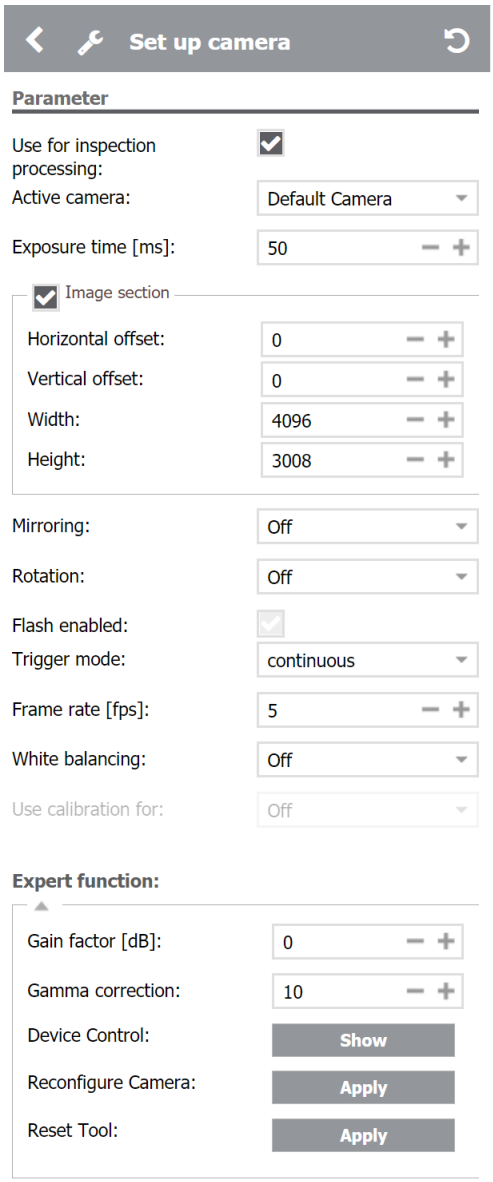
NOTE

Only physical cameras can be calibrated. If the selected logical camera is associated with the camera simulation ("File Device") in category **Cameras** of the **System Settings**, calibration is not possible.

Page	Explanatory text	Button	Comments
1 - Introduction	You are about to change the calibration of the current camera setup. After changing the calibration, you may have to reconfigure existing tools.		Based on camera calibration subsequent tools are enabled to carry out measurements or positioning tasks in world coordinates, i.e. in mm. The measurement plane is defined by the surface of the calibration plate. Hence, this is a 2-dimensional calibration: objects or contours of objects can be measured only if they are lying within this plane. Does the height of the object or the distance object to camera change the measurement results will be biased. Due to the perspective projection of the lens the resulting error is proportional to the difference in distance. For example: assuming the measurement plane to be defined at 100cm distance to the camera. If an object at 99cm distance is measured the relative measurement error is 1%. A new calibration process is to be carried out if the following situations occur: 1. The relative position of the camera to the measurement plane has changed. For example a new measurement plane is to be defined or the position of the camera has changed. 2. The objective is manipulated or modified, for example after adjusting the focal length or the aperture.
2 - Delete calibration	Delete the existing calibration or create a new one. If you want to delete an existing calibration, click on "**without calibration**". After that, the wizard is terminated. If you want to recalibrate the camera, click on "**with calibration**".	**without calibration** **with calibration**	The calibration affects all inspection programs on the camera. It is not possible to create different calibrations for individual inspection programs.
3 - Choose calibration plate	Five calibration plates with different size each can be selected: 40mm, 80mm, 160mm,	**40mm** **80 mm** **160 mm** **320 mm**	The calibration can only be carried out with the denoted calibration plates. Modified calibration plates and/or description files are not supported. The size of the calibration plate has to be chosen appropriately, such that the image is completely covered by the plate. Typically, just a part of the calibration plate is visible in the Image.

	320mm, 640mm and 1000mm. Select the plate which is right for your application.	**640 mm 1000 mm**	**NOTE** If you print the PDF template calibration plates, please keep in mind, that you 1. do not scale the print (100% setting) and 2. the paper printouts are less precise than solid calibration plates. For printing DIN A0/A1-plates by a print shop, please consider the shop's guidelines regarding document margins.
4 - Capture calibration image	Capture an image with the calibration plate clearly visible. Position the calibration plate in a way so that at least one search pattern is visible. Click on "Capture" to capture the calibration plate. To simplify the acquisition of an image of the calibration plate in triggered applications, a dialog box is displayed. It can be used to trigger an image.	**Capture**	A dialog box is shown, The calibration plate has to cover the whole image area. Pay attention that a least 10x10 circles are visible in the image in order to achieve accurate calibration results. Furthermore, at least one finder pattern has to be visible in order to conduct the calibration. Finder patterns are the special groups of circular rings. There are more information in the *expert knowledge* section.
5 - Calibration	Calibration successfull! Pixel error: _____ Calibration plate not found Click on "**Finish wizard**" to close the wizard.	**Finish wizard**	The calibration is finished and stored on the camera. It will be applied to all inspection programs. If no calibration plate was found, you have to capture a new image with an improved image quality. Take care that the plate is illuminated homogeneously and of sharp contrast. The mean calibration error is a measure for how precise subsequent measurement are possible. The precision is best if image contours are sharp and lie within the measurement plane. Then, a relative precision of about 0.001 to 0.0001 is feasible. If object contours are blurred or the contours are not lying exactly within the measurement plane then the precision is typically around 0.01 to 0.001. The relative precision is the ratio of absolute precision and the size of the field of view. For example, for a field of view of 100x100mm and a relative precision of 0.001 the absolute precision will be 0.1mm. The quality of the lens also impacts the measurement quality.

Parameters



Parameters	Description	Data type
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool
Active camera	Selection of a logical camera. Logical cameras are associated either with physical cameras or with the camera simulation in category **Cameras** of the **System Settings**. The camera simulation ("File Device") can be used to display stored images.	
Exposure time [ms]	Camera shutter speed in milliseconds (0,01 to 1000 ms)	Double
Image section	Size 16 x 16 to the maximum resolution of the connected camera (e.g. 1280 x 1024 for the **SMART***CAMERA*). If certain image areas, such as conveyor belt edges, etc. need to be left blank, this is done by reducing the image section. For a selected camera, the maximum size of the image is taken into account if the checkbox remains deactivated. After activating the checkbox, options for cropping are available. The parameters x, y, width and	[Int]

	height are entered using numeric values. However, they can also be set graphically by dragging and moving the blue box in the image.	
Mirroring	For the mirroring of the camera image are different modes available: • none - the mirroring is disabled • left / right - reflection at the vertical center line • top / bottom - reflection at the horizontal center line • both - combination of the two upper modes, corresponds to a rotation of 180 °. Mirroring is disabled as long as the **calibration** is active.	
Rotation	The camera image can be rotated. • 0 ° - no rotation • 90 ° - rotation through 90 ° • 180 ° - rotation through 180 ° • 270 ° - rotation around 270 ° Rotation is disabled as long as the **calibration** is active.	
Flash enabled	Activates the digital flash output.	Bool
Trigger mode	• continuous - no external triggering active. • level - acquires images, as long as a high or low level, depending on the inversion checkbox (see system settings / digital inputs / outputs), is on the input. This feature functions only with the physical input (not possible with command sent through TCP/UDP/Fieldbus). • edge - acquires images, if a low->high transition or high->low transition, depending on the inversion checkbox (see system settings / digital inputs / outputs), takes place on the input).	
Image batch/Import images	Opens a dialog to import images for the camera simulation.	
Image batch/Choose images	Opens a dialog to select the images for the camera simulation.	
Frame rate [fps]	Specifies the image refresh rate of the camera, which is used for the image acquisition (the camera simulation uses the stored images to achieve the frame rate). For performance reasons, tt is recommended to adjust the frequency to the inspection's speed. This will reduce the CPU load, because no unnecessary images will be acquired.	Double
White balance	Off - White balancing is disabled. one-time - By pressing the button with the hand symbol, the white balance is determined once according to the white balance area. As long as *one-time* is selected, this white balance setting is applied. A dialog box to trigger an image is displayed to simplify white balancing in triggered applications. continuous - The white balance is determined for each aqcuired image individually according to the white balance area.	
White balance area	Defines the area in the image used to determine the white balance. The white balance area has to be inside the image section. If the white balance area is moved outside of the image section, it will be reset to the position prior to moving. If the image section is moved so that the white balance area lies outside the image section after the move, then the white balance area will be centered inside the image section.	[Int]
Use calibration for	Several modes are available as soon as the camera has been calibrated. There are more information in the *expert knowledge* section. • nothing: Calibration is not applied • result: Calibration is applied to measured positions, measurements or positioning tasks are performed in mm, the image remains unchanged • image: Calibration is applied to the image, the rectified image is visualized, measurements or positioning tasks are performed in mm The calibration includes the settings of **mirroring** and **rotation**, which were set during the calibration process. These settings are applied to the image and current changes to the camera settings are ignored.	
Gain factor [dB] *Expert parameter*	Controls the gain as an absolute physical value in dB. This is an amplification factor that is applied to the image signal. However, this increases the noise of the image signal.	Double
Gamma correction *Expert parameter*	Not for **SMART***CAMERA:* Controls the gamma correction of pixel intensity. Tone value corrections such as changes in brightness and contrast, using a correction function of the form A=E γ , are called gamma correction. In a very simplified way, such a correction increases the brightness of the dark areas of the image with γ < 1, while with γ > 1 the brightness of the light areas of the image is reduced.	Double
Device Control	Not for **SMART***CAMERA:*	

Expert parameter	Opens a dialog with device information of the selected camera. Here, the firmware of the camera can be updated.	
Reconfigure Camera *Expert parameter*	Resets the internal camera settings to the default state and retransmits the current set of tool parameters to the camera. It is possible to change the internal camera settings with the help of a third-party application and to save them on the camera. In this case, a warning appears above this button that the correct function of the camera cannot be guaranteed.	
Reset Tool *Expert parameter*	Resets all camera-related tool parameters to default values and transmit them to the camera.	

NOTE

Different parameters are displayed depending on which camera is selected.

The actually used image can differ slightly from the selected area of interest. It is shown when selecting the *Get Image* Tool.

NOTE

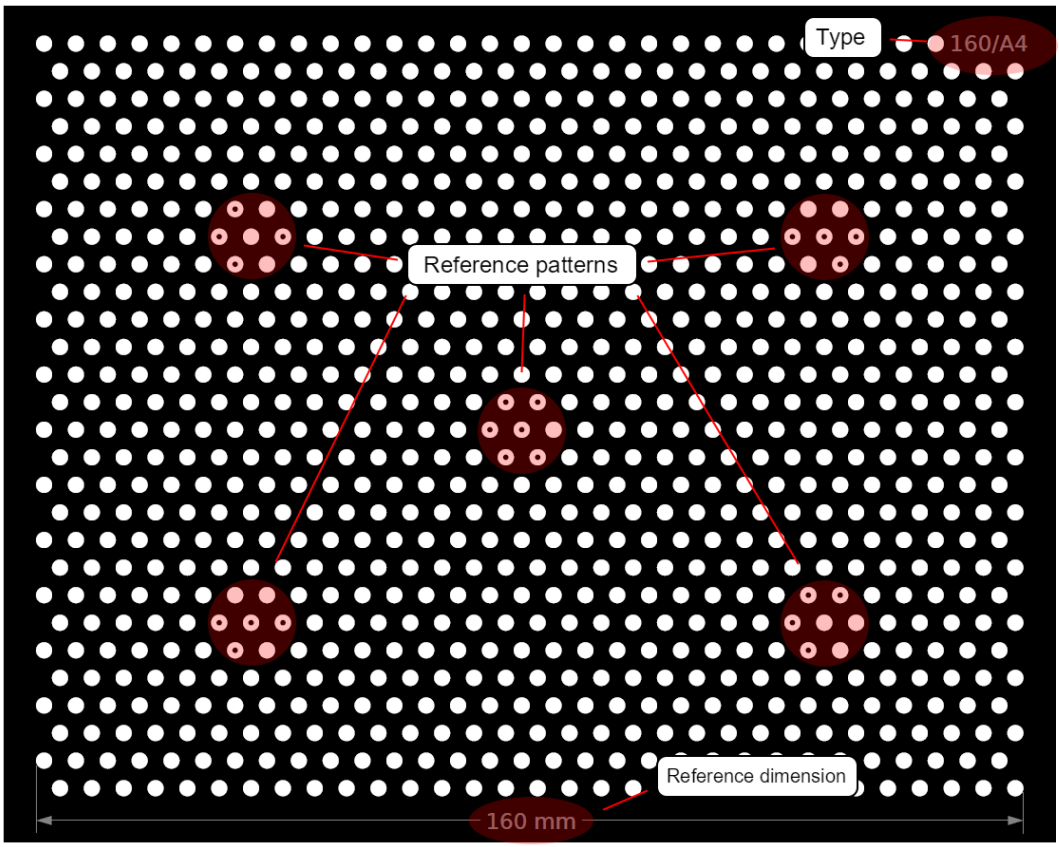
Flash and trigger have to be assigned to specific I/Os within the system settings in order to work properly.

NOTE

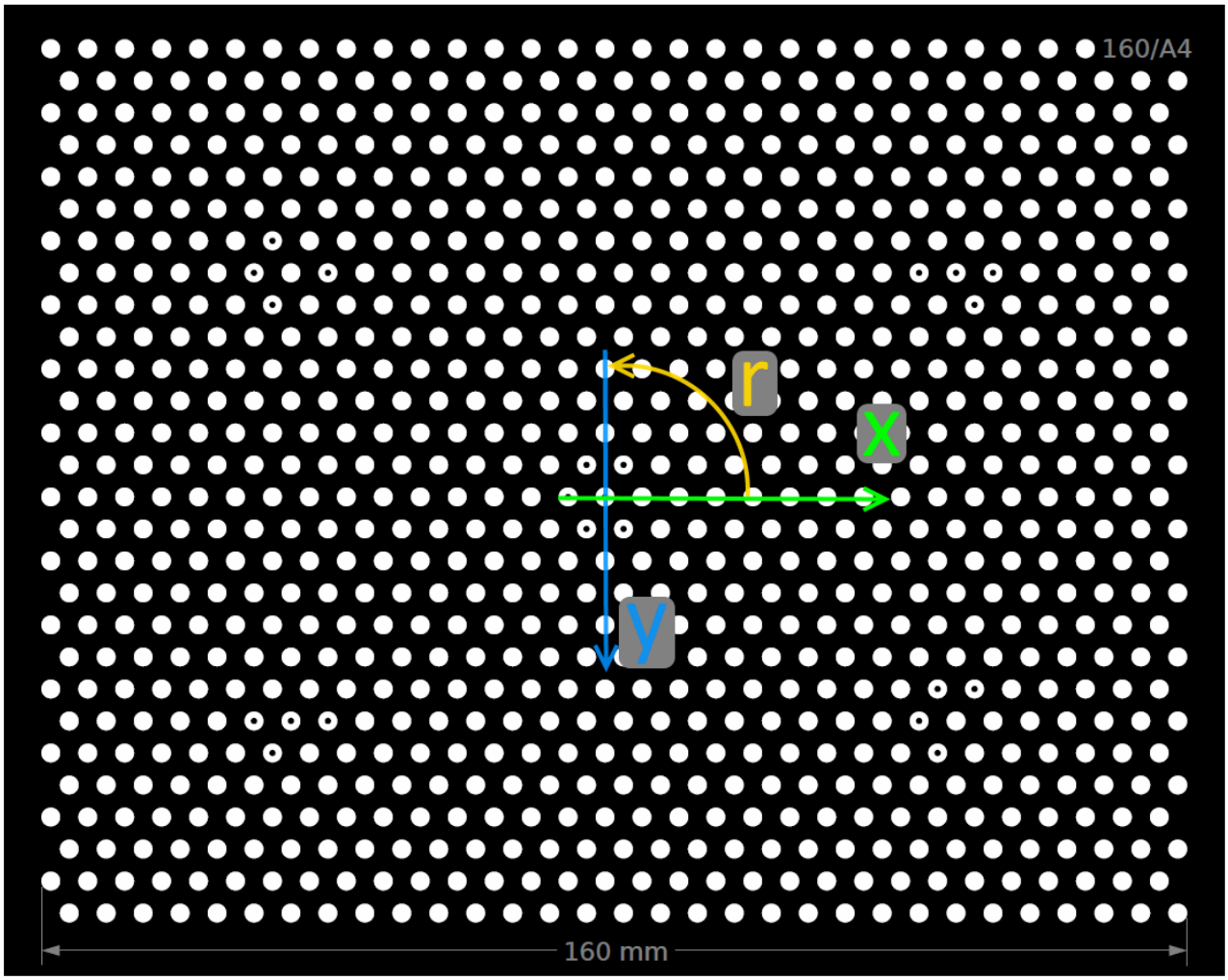
If the trigger mode is set to *onHighLevel* or *onLowLevel*, the camera captures images while the corresponding level is active. If a trigger is signaled, e.g. via fieldbus, UDP or by a user's click on "Step" in the configuration menu, an active level is simulated on the respective trigger input for a short time span. Depending on the settings for frame rate and exposure time, the camera may capture multiple frames or even none at all.

Expert knowledge

For several tools it can be an advantage, if the camera system extracts the true units of measurement in the acquired image. A calibration is the basis for units of measurement. An object with known geometric structures is shown to the camera system. With the acquired information the size, for example, of an object is determined, likewise the lens distortion is compensated.



For calibration, the needed calibration plate (here: type 160/A4) includes geometric structures in the form of circles. Additionally, there are five reference patterns on the calibration plate (red circles).



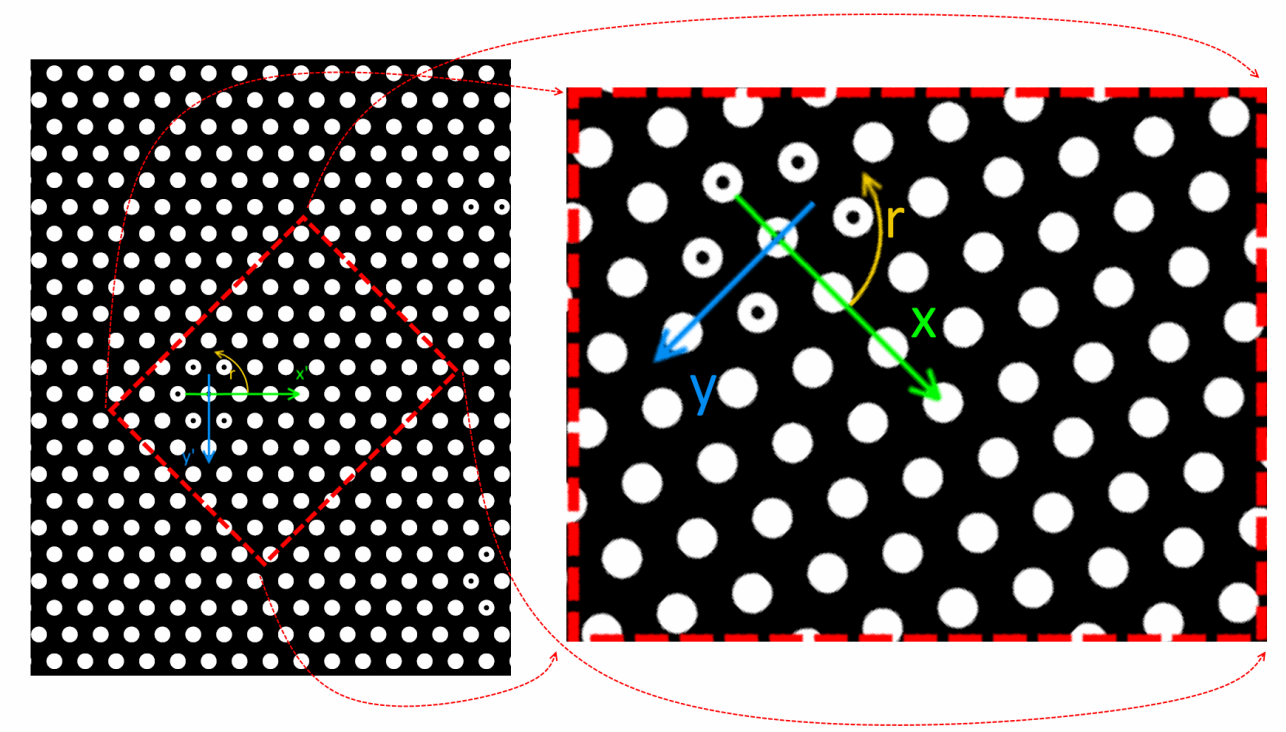
This image shows that the coordinate zero lies in the center of search pattern in the middle. The x-axis points to the right, the y-axis points down. The angle's zero position is along the x-axis, the angle is turned counterclockwise.

NOTE

To get a good calibration result, the view of the camera system should cover the complete calibration plate.

NOTE

The available PDF files include a reference dimension at the bottom. This reference dimension can be used to check the scale of the printout.



By rotating the calibration plate in relation to the optical axis of the camera system, you can rotate the coordinate system accordingly. The left part of the image shows the section, which the camera acquires (marked with the red frame). The right part of the image shows the calibration plate, how the camera system looks on it during the calibration process. In this sample, an object, which changes it position within the camera system's field of view to the right with one unit, moves proportionally within the calibrated coordinate system in +x and -y. The angle's reference system remains the same.

NOTE

Depending on the calibration plate's position and the orientation, the coordinate zero can also lie outside the acquired reference image. To determine the coordinate zero, the camera system has to see at least one reference pattern.

NOTE

If you want to measure properties relative to each other like the distance between two edges in the tool "**Measure objekt**", the calibration plate's position and orientation, which the calibration is based on, has no effect on the measurement result.

Chose the calibration mode depending on the used tools and the requirements concerning accuracy and speed.

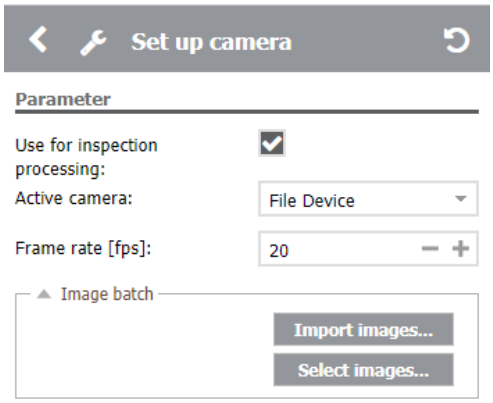
Use calibration for	Remark	Application	Advantages	Disadvantages
nothing	Acquired image is used directly in following tools. No rectification applied. Scaling needs to be evaluated manually.	No special requirements on evaluation of positions and measurement by tools like "**Read code**", "**Find object**", "**Measure object**" OR using	• no additional processing time	• scaling has to be evaluated manually • image distortion based on

		a premium quality object lens.		object lens quality
result	Calibration is only applied on result of tool "**Measure object**". Image itself is not rectified, which can lead to problems using object lenses with a strong fisheye-effect and trying to detect circles and lines. Scaling is evaluated automatically.	High requirements on results of tool "**Measure object**" using average quality object lens.	• scaling is evaluated automatically • results of "**Measure object**" stable on full image domain	• additional processing time of circa 1ms
image	Acquired image is rectified. Image distortions, like deviations on its border, are straightened. Rectified image is used in all following tools. Scaling is evaluated automatically.	High requirements on results of all tools using a low to average quality object lens.	• scaling is evaluated automatically • results of all tools stable on full image domain	• additional processing time of up to 30ms

Results

Result	Description	Data type	Default value	Tolerance adjustable
Tool processing	• Successful: No errors occurred. • Failed: Error on tool processing			✗
Results report	Text description of the cause of the error **NOTE** The error message "Image acquisition frequency is too high." indicates that the frequency of image acquisition is too high compared with the execution time of the inspection program. This may cause a lag between the current action in front of the camera and the displayed and processed images. Reduce the frequency of an external trigger or (in case of trigger mode *continous*) the value of parameter *Frame rate*.	String		✗
Execution time [ms]	The time this single tool needs for execution	Double	0	✗

Camera simulation "File Device"



Camera simulation can be used to display once more a previously acquired batch of images in an inspection program. Thus, it is e.g. possible to reproduce critical scenes. This is useful, if e.g. tool training was improved and verification is required. The batch of images is in a folder on the camera and is shared in the network.

NOTE

Currently, the File Device supports the image formats Windows Bitmap (BMP) and JPEG. Concerning resolution, images must provide either 8 bit per pixel for grascale images or 24 bit per pixel for color images. The image width and height may not exceed 8192 pixels, respectively.

Store image batch

The web interface is used to store a batch of images. To do this, click on the disk icon in the toolbar. BVS Cockpit then stores the image currently being displayed in the batch of images.

Access to the image batch

The folder containing the batch images is called *images*. You can find this folder in the workspace of each instance (e.g. *workspace0/images*, *workspace1/images*, etc.)..

Special case SmartCamera

The **SMART***CAMERA* shares the images folder in the network as *\\SC-150800010DE\images*. Here, "150800010" needs to be replaced with the serial number of the relevant camera.

NOTE

In order to access this folder you have to use the following credentials:

- Login: *expert*
- Password: *expert*

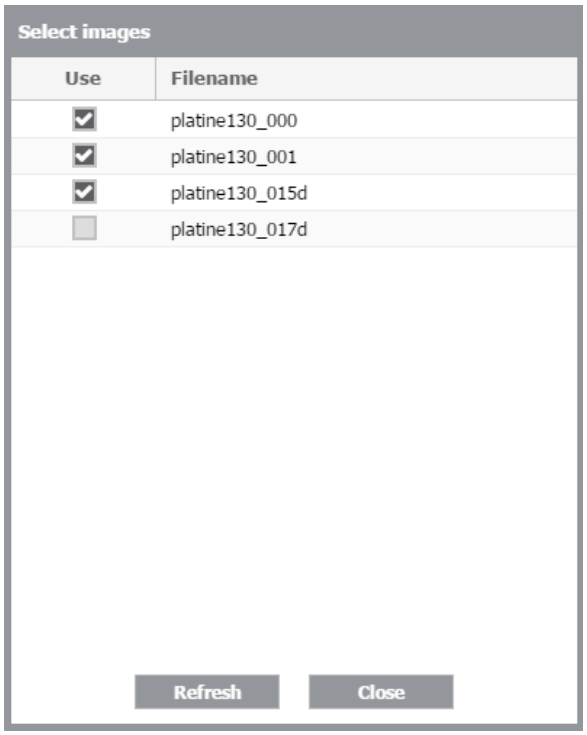
NOTE

If you cannot access the shared folders, please have a look at the appendix "Troubleshooting Table" → "Accessing the shared folders is not possible".

Display image batch

To display a batch of images using camera simulation, select a logical camera that is associated with the camera simulation ("File Device") in category **Cameras** of the **System Settings**. All BMP and JPEG images from the *images* folder are then used in order from the camera simulation as an input image for the inspection program.

Manage image batch



The images used in the simulation mode can be selected over a dialog which can be accessed via the *Select images* button. Within the dialog the filename and a checkbox are displayed. With the selection of the checkbox the image is used in the simulation mode. After copying images to the network share this list has to be updated with a press of the *Refresh* button.

NOTE

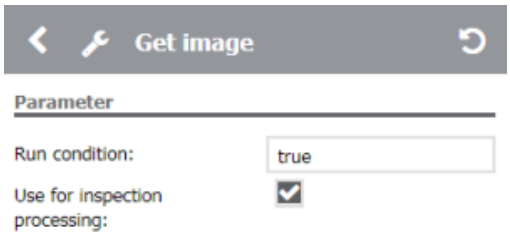
The camera stores the batch of images in the flash memory. Continuous writing and deleting of memory cells may compromise the life of the flash memory.

5.1.2 Get image

Task

Acquires an image from the camera specified in "**Set up camera**".

Parameters



Parameter	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, the outputs of this tool will be set to default values: • Arithmetic types: 0 • Strings: empty string • Images: empty images	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool

Results

Result	Description	Data type	Default value	Tolerance adjustable
Tool processing	• Successful: No errors occurred. • Failed: An error occurred with image acquisition			✗
Results report	Text description of the cause of the error **NOTE** The error message "Image acquisition frequency is too high." indicates that the frequency of image acquisition is too high compared with the execution time of the inspection program. This may cause a lag between the current action in front of the camera and the displayed and processed images. Reduce the frequency of an external trigger or (in case of trigger mode *continous*) the value of parameter *Frame rate* in tool **Set up camera**.	String		✗
Execution time [ms]	The time this single tool needs for execution. Does not comprise time that is spent solely on waiting for a trigger.	Double	0	✗
Image	The image of the acquisition.	Image	Blank image	✗
Trigger alarm	True: An image was lost because the triggers are faster than the camera is able to process the images. Only one image fits into the buffer (additionally to the one currently processed). False: While processing no or only one trigger signal was issued. No images were lost.	Bool	False	✗

5.2 Basic tools

5.2.1 Check focus

Task

Checks the focus in an image area independently from its absolute brightness. For example, this tool can be used to check focus settings of a camera lens or if a specimen is within focus range.

> **NOTE**
>
> This tool should not be used to check the homogeneity of an image section. It is recommended to use "**Check brightness**" for that purpose.

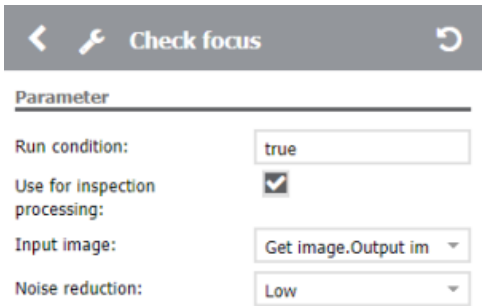
Wizard

The wizard is used to define the sections in the image, where the image sharpness is checked.

Page	Explanatory text	Button	Comments
1 - Define areas to check / Finish wizard	Define areas with the brush (mark /erase). Focus the camera, until you obtain the maximum focus value. Click on "**Finish wizard**" to close the wizard.	**Delete mask** **Set maximum mask** **Reset mask** **Finish wizard**	The sections in the image, where the image sharpness is checked, can be edited here. Using the zoom will change the size of the brush/rubber. Using the buttons will expand the mask to the whole image, remove it completely or reset it to the initial state.

> **NOTE**
>
> The area of interest in inspection programs, which were created by an older version of BVS Cockpit (< 1.3.1), will be converted automatically when the inspection program is loaded.

Parameters



Parameters	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, the outputs of this tool will be set to default values: • Arithmetic types: 0 • Strings: empty string • Images: empty images	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool

Input image	Image on which the operation should be executed.	Image
Noise reduction	Sets desired noise reduction for current input image. In most cases, setting "Low" leads to a fair compromise between noise reduction and processing time.	Enum ("Strong", "Medium", "Low", "None")

Results

Result	Description	Data type	Default value	Tolerance adjustable
Tool processing	• Successful: No errors occurred. • Failed: The definition is outside the tolerance range or a general error occurred			✗
Results report	Text description of the cause of the error	String		✗
Edges used for the focus calculation	Top one percent of strongest gray value differences. NOTE This property can be used, for example, to visualize the area of an object which the lens is focused on.			
Calculated focus value	The calculated numeric degree of definition as %. Focus value equals to 100%, if two neighboring pixels differ in gray value levels as much as the maximum gray value difference of all pixels in the masked region. NOTE Using a strong noise reduction setting can elimate small chunks of pixels with high gray value devations w.r.t. their neighboring pixels.	Double	0.0	✓
Area of interest	Image section has been verified. NOTE Only displayed in the monitor mode. If the calculated focus value is within the tolerance range then this area is displayed green, else in red	[Double]	The dimension of the input area on interest	✗

NOTE

An image transformation may cause that the area of interest is outside the image bounds. In this case the tool calculates the focus value 0.

5.2.2 Check brightness

Task

Checks the brightness in an image section. Also allows to count the number of pixel within a adjustable range.

> **NOTE**
>
> This tool should not be used to evaluate focus setup of a camera lens. It is recommended to employ tool "**Check focus**" for that purpose.

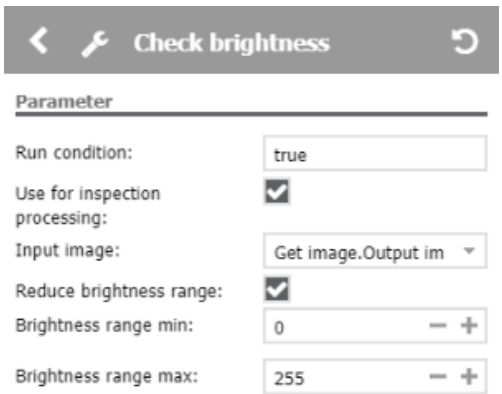
Wizard

The wizard is used to define the sections in the image, where the image brightness is checked.

Page	Explanatory text	Button	Comments
1 - Define areas to check / Finish wizard	Define areas with the brush (mark /erase). Click on "**Finish wizard**" to close the wizard.	**Delete mask** **Set maximum mask** **Reset mask** **Finish wizard**	The sections in the image, where the image brightness is checked, can be edited here. Using the zoom will change the size of the brush/rubber. Using the buttons will expand the mask to the whole image, remove it completely or reset it to the initial state.

> **NOTE**
>
> The area of interest in inspection programs, which were created by an older version of BVS Cockpit (< 1.3.1), will be converted automatically when the inspection program is loaded.

Parameters



Parameters	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, the outputs of this tool will be set to default values: • Arithmetic types: 0 • Strings: empty string • Images: empty images	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool
Input image	Image on which the operation should be executed.	Image
Reduce brightness range	Activates counting of pixel within a gray scale range.	Bool
Brightness range min	Minimum (including) gray scale value of pixel to count.	Int16
Brightness range max	Maximum (including) gray scale value of pixel to count.	Int16

Results

Result	Description	Data type	Default value	Tolerance adjustable
Tool processing	• Successful: no errors occurred. • Failed: the brightness is outside the tolerance range or a general error occurred			✗
Results report	Text description of the cause of the error	String		✗
Execution time [ms]	The time this single tool needs for execution	Double	0	✗
Brightness value	The calculated current brightness value. If **Reduce brightness range** is set to active, **Brightness value** is calculated considering only gray scale values within the selected range.	Double	0	✓
Homogeneity value	The calculated homogeniety value in percent. A value of 0% corresponds to an even distribution of completely dark and completely bright pixels, 100% will be calculated if pixels have the same gray level value. Corresponding to **Brightness value**, if **Reduce brightness range** is set to active, **Homogeniety value** is calculated considering only gray scale values within the selected range.	Dobule	100	✓
Number of pixels in brightness range	Number of pixels within the to be checked image section featuring a gray scale value as defined by **Range min** and **Range max**. If **Reduce brightness range** is active, the number of pixels will be shown and tolerance range can be changed.	Uint64	0	✓
Pixels in brightness range	Region with pixels that are used for brightness calculation.	Region		✗
Area of interest	Image section has been verified. If the calculated brightness value is within the tolerance range then this area is displayed green, else in red	Region	The dimension of the input area on interest	✗

NOTE

An image transformation may cause that the area of interest is outside the image bounds. In this case the tool calculates the brightness value 0.

5.2.3 Check color

Task

Checks the color, consisting of hue (H), saturation (S) and intensity (I), in an image section.

For the conversion from RGB to HSI we use the following equations:

$$M1 = ((2 \cdot R) - G - B) / \sqrt{6}$$

$$M2 = (G - B) / \sqrt{2}$$

$$H = \mathrm{atan2}(M2, M1)$$

$$S = \sqrt{(M1^2 + M2^2)}$$

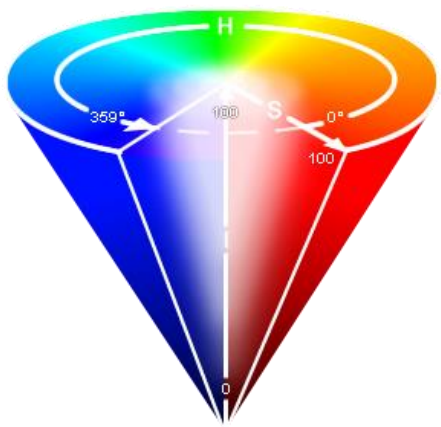
$$I = (R + B + G) / 3$$

NOTE

The tool is only available on cameras which supports color images.

NOTE

The hue (H) is given as an angle of 0° to 359°. All visible colors are arranged on a circle. 0° corresponds approximately to red, 60° to yellow, 120° to green, 180° to turquoise, 240° to blue and 300° to magenta.

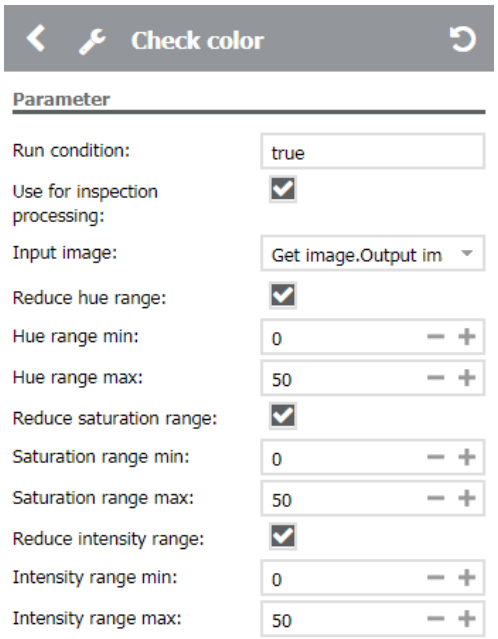


Wizard

The wizard is used to define the sections in the image, where the color values are checked.

Page	Explanatory text	Button	Comments
1 - Define areas to check / Finish wizard	Define areas with the brush (mark /erase). Click on "**Finish wizard**" to close the wizard.	**Delete mask** **Set maximum mask** **Reset mask** **Finish wizard**	The sections in the image, where the color values are checked, can be edited here. Using the zoom will change the size of the brush/rubber. Using the buttons will expand the mask to the whole image, remove it completely or reset it to the initial state.

Parameters



Parameters	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, the outputs of this tool will be set to default values: • Arithmetic types: 0 • Strings: empty string • Images: empty images	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool
Input image	Image on which the operation should be executed.	Image
Reduce hue range	Activates counting of pixel within a range of hue values.	bool
Hue range min	Minimum (including) hue value of pixels to count. Must be within [-180, 359].	Int16
Hue range max	Maximum (including) hue value of pixels to count. Must be within [0, 540].	Int16
Reduce saturation range	Activates counting of pixel within a range of saturation values.	bool
Saturation range min	Minimum (including) saturation value of pixels to count. Must be within [0, 100].	Int16
Saturation range max	Maximum (including) saturation value of pixels to count. Must be within [0, 100].	Int16
Reduce intensity range	Activates counting of pixel within a range of intensity values.	bool
Intensity range min	Minimum (including) intensity value of pixels to count. Must be within [0, 100].	Int16
Intensity range max	Maximum (including) intensity value of pixels to count. Must be within [0, 100].	Int16

Results

Result	Description	Data type	Default value	Tolerance adjustable
Tool processing	• Successful: no errors occurred. • Failed: the color is outside the tolerance range or a general error occurred			✗
Results report	Text description of the cause of the error	String		✗
Execution time [ms]	The time this single tool needs for execution	Double	0	✗
Number of pixels in color range	Number of pixels within the checked **Area of interest** that are within **Range min** and **Range max**. If **Reduce hue range, Reduce saturation range** and **Reduce intensity range** are not activated, total count of pixels in **Area of interest** is returned. Tolerance range can only be changed if **Reduce hue range, Reduce saturation range** or **Reduce intensity range** is activated.	Uint64	0	✓
Pixels in color range	Region with pixels that are used for calculation.	Region		✗
Area of interest	Image section has been verified. If the calculated color value is within the tolerance range then this area is displayed green, else in red	Region	The dimension of the input area on interest	✗
Hue (H) [°]	The calculated current hue value. The value is represented as angle in the range of [0°-359°]	Double	0	✓
Saturation (S) [%]	The calculated current saturation value. The value is represented as percentage from [0-100%]	Double	0	✓
Intensity (I) [%]	The calculated current intensity value. The value is represented as percentage from [0-100%]	Double	0	✓

NOTE

An image transformation may cause that the area of interest is outside the image bounds. In this case the tool calculates 0 values for H,S and I.

5.2.4 Filter image

Task

Filters an image and makes it available for the other tools at the output.

Wizard

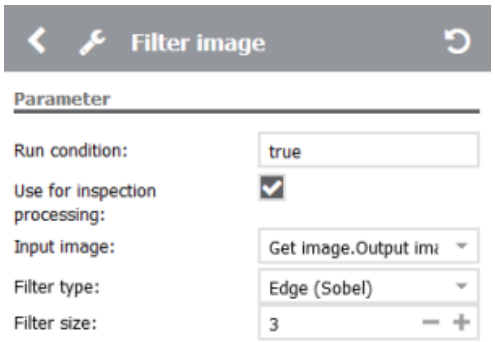
The wizard is used to define the sections in the image on which the filter should be applied to.

1 - Define areas to check / Finish wizard	Define areas with the brush (mark /erase). Click on "**Finish wizard**" to close the wizard.	**Delete mask** **Set maximum mask** **Reset mask** **Finish wizard**	The sections in the image on which the filter should be applied to, can be edited here. Using the zoom will change the size of the brush/rubber. Using the buttons will expand the mask to the whole image, remove it completely or reset it to the initial state.

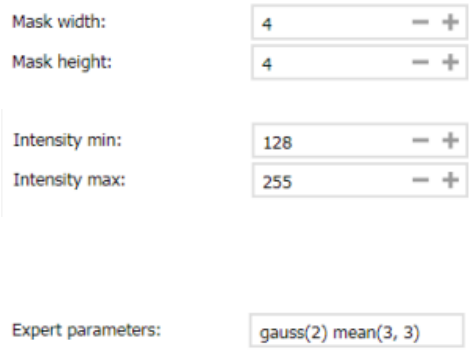
NOTE

The area of interest in inspection programs, which were created by an older version of BVS Cockpit (< 3.1), will be converted automatically when the inspection program is imported.

Parameters



Depending on filter type different parameters are displayed:



Parameters	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, the outputs of this tool will be set to default values: • Arithmetic types: 0 • Strings: empty string • Images: empty images	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result or not.	Bool
Input image	Image on which the operation should be executed.	Image
Use gray image	When checked, the image is converted to a gray values before applying the filter. This parameter is only available for color cameras.	Bool
Filter type	Sets the filter type. **Filter type** / **Description** None: No filter Smoothing (Gaussian): Gaussian Filter, smoothing via gauss function. Smoothing (Median XY): Median filter, rectangular. Smoothing filter, preserves edges and removes outlier. Smoothing (Median circle): Median filter, circular. Smoothing filter, preserves edges and removes outlier. Opening: Opening filter, rectangular. Removes bright areas, smaller than filter size. Closing: Closing filter, rectangular. Removes dark areas, smaller than filter size. Dilation: Dilation filter, rectangular. Enlarges bright areas. Erosion: Erosion filter, rectangular. Enlarges dark areas. Threshold: Creates a binary image. Areas between minimum and maximum become white, other areas become black. Edge (Sobel): Creates a edge image. Pixels with distinctly different gray values to neighbors become bright, other become dark. Expert_Mode: Allows manual configuration and combination of filters	Enum
Filter size	Sets the filter size for Smoothing (Gaussian), Edge (Sobel) and the radius for Smoothing (Median circle). The side length of the quadratic filter mask of the Smoothing (Gaussian) and Edge (Sobel) filter is (2 * filter size + 1).	UInt16
Mask width	Sets the mask width for rectangular filters. The size of the filter mask is calculated (2 * width + 1).	UInt16
Mask height	Sets the mask height for rectangular filters. The size of the filter mask is calculated (2 * height + 1).	UInt16
Intensity min	Sets the minimum intensity for the Threshold filter.	UInt16
Intensity max	Sets the maximum intensity for the Threshold filter.	UInt16
Expert parameters	A list with filter parameters for the expert mode. Filters are defined in the format ***filter(parameter1, ... parameterN)***, multiple filters are separated by spaces. **Example:** "*mean(3, 3) gauss(5)*" **Allowed filters** / **Corresponding standard filter (same number and meaning of parameters)** gauss: ✓ median: ✓ median_circ: ✓ mean: open: ✓ close: ✓ hist: scale_max: scale: dilation: ✓ erosion: ✓ thresh: ✓ sobel_amp: ✓ kirsch_amp: Additionally the following filters are available: **Filter type** / **Description**	String

mean	A mean value filter, rectangular. Expects mask width and mask height as parameters. The side length of the filter mask is calculated (2 * specified value + 1).	
hist	Histogram linearization, expects no parameters.	
scale_max	Scales the gray values of an image to 0..255, expects no parameters.	
scale	Scales the gray values between minimum and maximum to 0..255. Expects minimum and maximum as parameters.	
kirsch_amp	Calculates an approximation of the first derivative of gray values and is used as an edge detector. Expects no parameters.	

Results

Result	Description	Data type	Default value	Tolerance adjustable
Tool processing	• Successful: no errors occurred. • Failed: an error occurred while filtering, i.e. invalid expert filter settings.			✗
Results report	Text description of the cause of the error	String		✗
Execution time [ms]	The time this single tool needs for execution.	Double	0	✗
Output image	The filtered output image	Image	The input image	✗

5.3 Analytical tools

5.3.1 Find object

Task

Establishes the position of an object in the image and rotates the image, so that the object is always seen in the same orientation. Thus, other tools can thus take a position relative to the object. The tool allows to train multiple different object types. When executing the tool it finds and aligns the object type that fits best to the trained data. The objects center of gravity is visualized as a circle.

Wizard

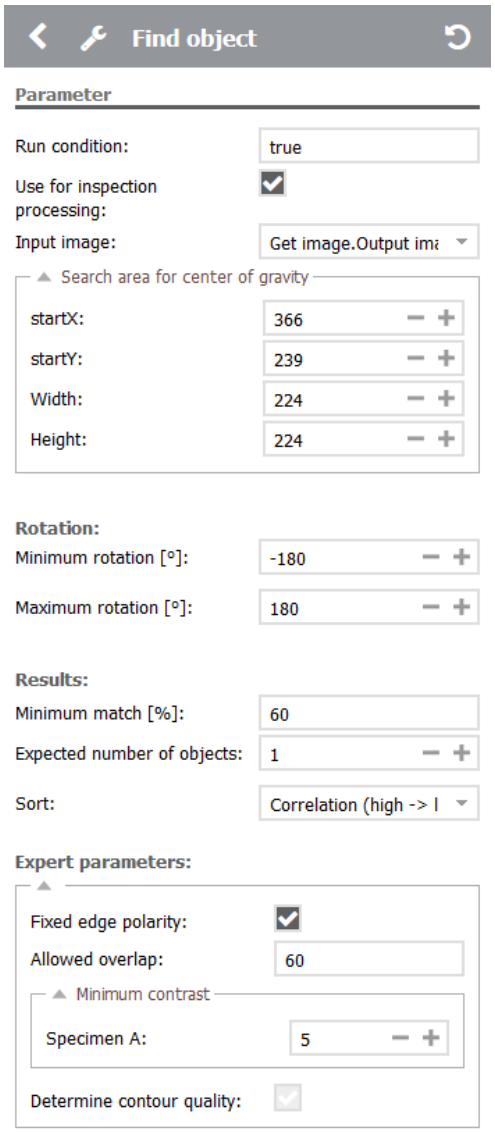
The tool is trained using example images. The background and the target position are displayed.

Page	Explanatory text	Button	Comments
1 - Introduction	In the next steps test objects are defined and positioned interactively. Subsequent tools need take no further account of the element's orientation.		
2 - Choose object type	Train new object type - or - Edit trained object type The name of the new object type must only contain ASCII-conformant characters. If you want to delete existing training data, click on "**Remove all**".	**Remove all**	A new object type can be trained or an existing one can be modified. The tool is reset by using *Remove all*. Using the recycle bin allows to remove individual object types.
3 - Define background	Optionally: Capture static background without test object. If you wish to acquire a background image, remove the element under the camera, so that only the background is visible. Then click on "**Capture**", to acquire the background as a reference image.	**Capture**	This is a optional step that is used to automatically suggest a correct mask (next page) If the object is as large as the image itself capturing a background image makes no sense.
4 - Reference	Position reference object in reference position. Center an element directly under the camera and click on "**Capture**". The tool will orientate all future elements in exactly the same way as this element.	**Capture**	The target position of the object is learned.
5 - Constant area	Define object areas with the brush (mark /erase). Remove the highlighting in areas subject to variations, so that the only areas highlighted are those that appear the same for all elements. In the toolbar, you can switch between eraser (remove highlighting) and paintbrush (extend highlighting). To improve stability and speed remove the mask on self similar structures and regular patterns.	**Delete mask** **Set maximum mask** **Reset mask** **Display model contour**	The area that should be used for learning the element can be edited here. Using the zoom will change the size of the brush/rubber. Using the buttons will expand the mask to the whole image, remove it completely or reset it to the initial state. The model contour button appears if a trained object type already exists. The checkbox can be used to display the model contour of the already trained object type.

6 - Check Object Features	The picture shows the features (model contour) that are used to find the object. Check the visualized characteristics: • Are any characteristics missing which are needed to identify the object? • Are these characteristics present in every object of this object type? • Do characteristics differ between dissimilar object types? If important characteristics are missing or to many characteristics are used, you can go back to try improving the result using other reference images or changing the mask. Although the mask is correct, if essential characteristics are not recognized, you have to adapt the illumination to get a better contrast.		The image visualizes the characteristics, which the tool has learned. The tool will only search for these characteristics.
7 - Set Reference Point	Optional: Set the reference point. The coordinates of the reference point, which is marked as a green cross, can either be entered using the input fields or it can be moved to any desired location on the image. By defining a reference point, one can exactly define the position of the found object which will be output as coordinates. This can be useful, for example, in applications where a robot has to grip an object at a specific position. Click on "**Finish wizard**" to close the wizard. Click on "**Next object type...**" to close the wizard.	**Finish wizard** **Next object type...**	This step is optional. Using the button "**Next object type...**" will relocate you to page 2 where you can train additional object types.

The wizard generates a parameter record, which you can amend manually using tool settings. If the object is found, the tool displays the rotated output image. If the object is not found, the initial state of the image is displayed.

Parameters



Parameters	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, the outputs of this tool will be set to default values: • Arithmetic types: 0 • Strings: empty string • Images: empty images	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool
Input image	Image on which the operation should be executed.	Image
Search area for center of gravity (x, y, width, height)	Permitted position variation of the center of gravity with regard to the input image.	[Double]
Minimum rotation [°]	Starting angle of the permitted object rotation. If both this and parameter **Maximum rotation [°]** are set to 0, found objects will always be reported with a rotation of 0, disregarding their actual angular deviation. This can be useful for coping with rotationally symmetric objects like pins and nuts.	Double
Maximum rotation [°]	Finishing angle of the permitted object rotation. See also **Minimum rotation [°]**.	Double

<table>
<tr><td>Minimum match [%]</td><td>The minimum quality to be achieved for an object to be accepted. Objects of lesser quality are rejected.</td><td>Double</td></tr>
<tr><td>Expected number of objects</td><td>The maximum number of objects to be found. Allowed value range is 1 to 100.</td><td>Int</td></tr>
<tr><td>Sort</td><td>Determines the criteria by which found objects will be sorted after the search operation.
<table>
<tr><th>Sort mode</th><th>Description</th></tr>
<tr><td>Correlation (high → low)</td><td>Objects are sorted by their correlation. Objects with the highest correlation are returned first</td></tr>
<tr><td>Position x (left → right)</td><td>Objects are sorted by the x coordinate of their reference point from left to right</td></tr>
<tr><td>Position x (right → left)</td><td>Objects are sorted by the x coordinate of their reference point from right to left</td></tr>
<tr><td>Position y (top → bottom)</td><td>Objects are sorted by the y coordinate of their reference point from top to bottom</td></tr>
<tr><td>Position y (bottom → top)</td><td>Objects are sorted by the y coordinate of their reference point from bottom to top</td></tr>
</table>
NOTE

Sorting takes place after the object search. This means that the results are not dependent on the sort mode. Only the order of results will change with the sort mode.</td><td>Enum</td></tr>
<tr><td>Verify COG position
Expert parameter</td><td>Verifies whether the center of gravity of the matched object resides within the given search area. The tool result is 'Failed' if at least one verification does not pass.</td><td>Bool</td></tr>
<tr><td>Fixed edge polarity

Expert parameter</td><td>Setting this parameter to true will result in finding only matches which feature the same kind of edge intensity transition as the reference object. If this parameter is deactivated, a bright object on dark background will be found even the training used a dark object on bright background.
NOTE

Depending on trained object, changing this parameter can occupy the camera for up to 5 seconds.</td><td>Bool</td></tr>
<tr><td>Allowed overlap [%]
Expert parameter</td><td>Defines how much individual matches may overlap. A high value allows high overlap. Calculating overlap uses objects enveloping rectangle. If two objects overlap equally, the one having a higher matching score will be used.</td><td>Double</td></tr>
<tr><td>Minimal contrast
Expert parameter</td><td>Required contrast for an edge on current image to be relateable to a certain object type.
NOTE

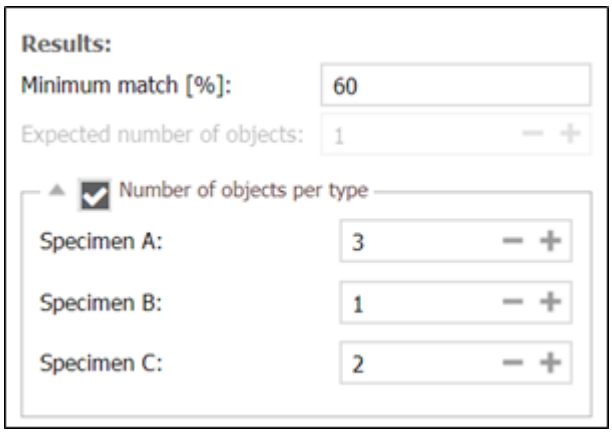
In practice, edges usually are wider than a single pixel due to selected optics, camera resolution as well as the edge itself. Using mouse indication to determine the gray value difference of both areas next to the edge provides a possible estimate for minimal contrast. On third of the determined gray value difference can be taken as a first approximation for minimal contrast.</td><td>Int</td></tr>
<tr><td>Determine contour quality
Expert parameter</td><td>This parameter allows to visualize the match quality of a taught objects to the currently found object. Depending on local match quality, the shown contour is broken down in segments which are colored accordingly to match quality and therefore allows an visual estimation. Green represents a good match, yellow corresponds to a weak match, and red indicates no match.
Calculation of contour qualties takes additional computing time and therefore should not be activated in time critical applications. If this parameter is deactivated after performing an visual analysis, no addiotional computing time will be used.</td><td>Bool</td></tr>
</table>

	NOTE The visualization won't give a perfect estimation, it should be considered as an simplified approximation. The percentage of green colored contour elements does not exactly represent the score value. Images with heterogenius backgrounds do affect the visualization.	

NOTE

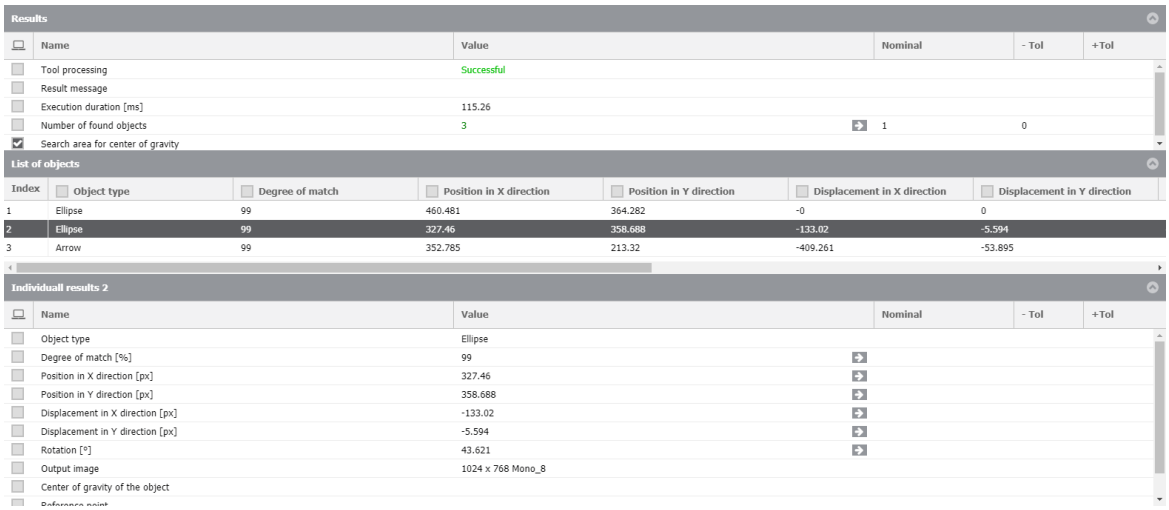
By reducing the size of the search area or the range of rotation angle, the algorithms of "Find Object" can significantly sped up.

Optional parameters



Parameters	Description	Data type
Number of objects per type	If more than a single object is trained, this parameter will appear. Setting this parameter to *true*, an individual number of expected matches can be defined for each object type. Consequently, parameter **Expected number of matches** is deactivated and can't be changed as long as **Number of objects per type** is activated. Deleting trained objects in wizard resulting in overall number of trained objects below 2 will make **Number of objects per type** disappear. Furthermore, the value defined in **Expected number of matches** will be used again.	Bool
Object name	Defines the maximum number of expected matches per object type.	[Int]

Result tables



Results

Name	Value	Nominal	- Tol	+Tol
Tool processing	Successful			
Result message				
Execution duration [ms]	115.26			
Number of found objects	3	1	0	
Search area for center of gravity				

List of objects

Index	Object type	Degree of match	Position in X direction	Position in Y direction	Displacement in X direction	Displacement in Y direction
1	Ellipse	99	460.481	364.282	-0	0
2	Ellipse	99	327.46	358.688	-133.02	-5.594
3	Arrow	99	352.785	213.32	-409.261	-53.895

Individuall results 2

Name	Value	Nominal	- Tol	+Tol
Object type	Ellipse			
Degree of match [%]	99			
Position in X direction [px]	327.46			
Position in Y direction [px]	358.688			
Displacement in X direction [px]	-133.02			
Displacement in Y direction [px]	-5.594			
Rotation [°]	43.621			
Output image	1024 x 768 Mono_8			
Center of gravity of the object				

Table 1: Results

This table contains all general results belonging to the tool.

Result	Description	Data type	Default	Tolerance adjustable
Tool processing	• Successful: no errors occurred. • Failed: the tool is not trained, the number of found objects is outside the specified tolerance range, or a general error occurred.			✗
Results report	Text description of the cause of the error	String		✗
Execution time [ms]	The time this single tool needs for execution	Double	0	✗
Number of found objects	The actual number of found objects.	Int	0	✓
Search area for center of gravity	Area of permitted position variation of the center of gravity of the objects.	[Double]	Same as the input search area	✗
Model contour (good match)	Approximated visualization of taught object contours that match well with found object. Only visible if **Determine contour quality** is active. Region is shown in green color.	Region		✗
Model contour (weak match)	Approximated visualization of taught object contours that match weakly with found object. Only visible if **Determine contour quality** is active. Region is shown in yellow color.	Region		✗
Model contour (no match)	Approximated visualization of taught object contours that do not match with found object. Only visible if **Determine contour quality** is active. Region is shown in red color.	Region		✗

Table 2: List of all found objects

This table contains the list of all found objects. Is shown as soon as "Expected number of objects" is above 1.

Result	Description
Object_1	Represents a found object with its number. The properties of each object are displayed columnwise in this table. When an object is selected in this table, its properties are displayed in detail in the third table.

Table 3: Properties of the selected object

This table contains the properties of the result object that has been selected in table 2. Is shown as soon as "Expected number of objects" is above 1.

Result	Description	Data type	Default	Tolerance adjustable
Object type	Shows the name of the detected object type.	String		✗
Degree of match [%]	Match of the found object with the trained object, a value of 100 means full match, 0 means no match.	Double	0	✓
Position in X direction	Horizontal absolute object position w.r.t. upper left image origin. Position will be returned in world coordinate system if a calibration was performed in the tool **Set up camera**.	Double	0	✓
Position in Y direction	Vertical absolute object position w.r.t. upper left image origin. Position will be returned in world coordinate system if a calibration was performed in the tool **Set up camera**.	Double	0	✓
Move in X direction	Horizontal object move with respect to the target position	Double	0	✓
Move in Y direction	Vertical object move with respect to the target position	Double	0	✓
Rotation	Rotation angle variation with respect to the target position. Rotation angle will be returned in world coordinate system if a calibration was performed in the tool **Set up camera**.	Double	0	✓
Output image	Rotated output image.	Image	Blank image	✗
Center of gravity of the object	Center of gravity of the found object.	Overlay		✗
Reference point	Reference point of the found object.	Overlay		✗
Model contour	Model contour of the found object. Only visible if **Determine contour quality** is not active.	Region		✗

If *Tool Processing* is expected to be displayed as Failed subject to the finding of an object, that can be reached by setting feasible tolerances. Possible settings are listed in the table.

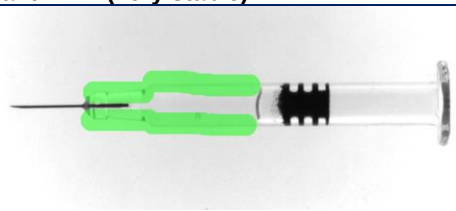
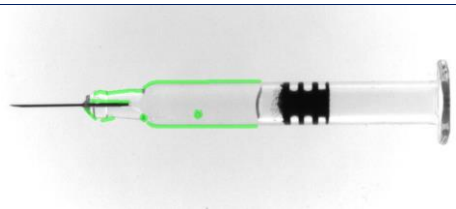
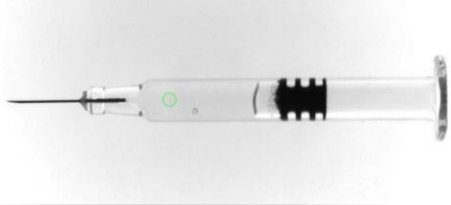
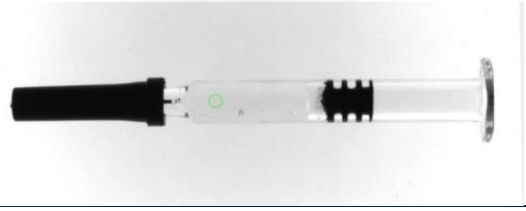
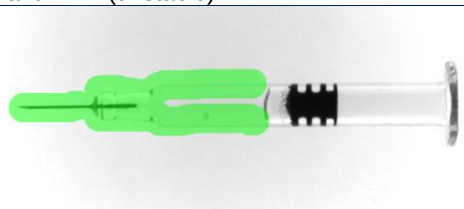
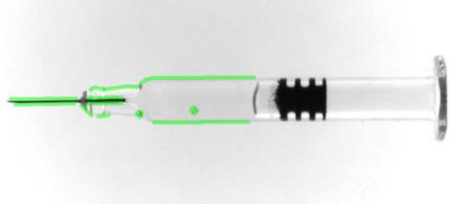
Description	Nominal	-Tol	+Tol
Tool Processing Failed if no object was found (default)	1	0	empty
Tool Processing Failed if an object was found	0	empty	0
Tool Processing Successful independent if an object was found	empty	empty	empty

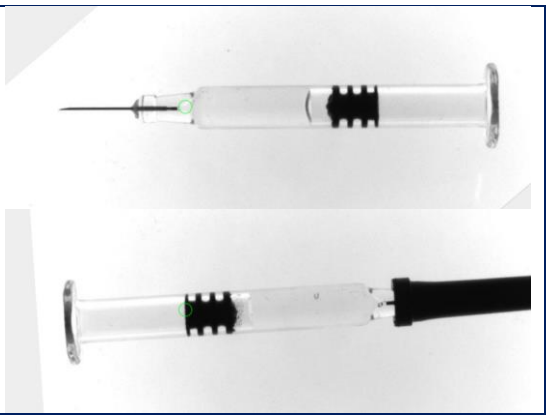
Expert knowledge

Search features

Choose features which are available in all scenarios

If you define features which are not available in all images or are changed significantly, you will get a poor search result quality in the production environment. The result is that you have to reduce the minimum threshold to determine a successful search. This increases the probability of recognizing unwanted objects wrongly; it is also possible to receive wrong position and angle results.

	Variant A (very stable)
Mask **NOTE** Doesn't contain needle, as it may be covered by cap.	
Determined features	 Independent from the manifestation
Production environment **NOTE** Object is found reliably with or without cap.	 
	Variant B (unstable)
Mask **NOTE** Includes needle that may be covered by cap.	
Determined features	 Depends on the manifestation "without cap"

Production environment **NOTE** Needle is covered by cap so not all trained features are visible. This results in a wrong placement / result.	

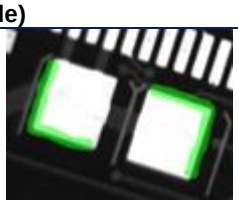
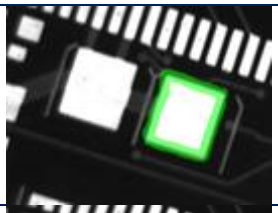
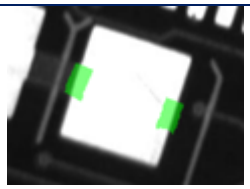
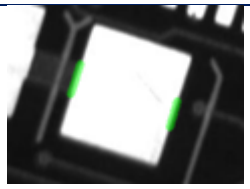
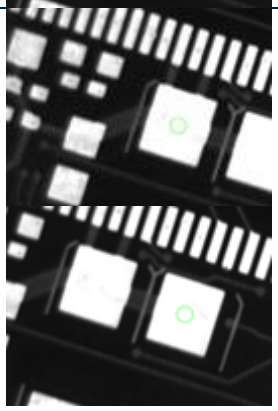
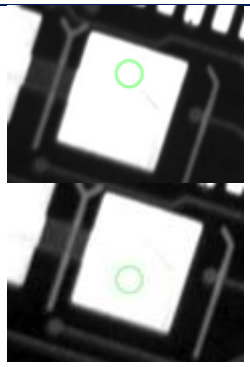
Choice of significant features

You should not choose features which can be found several times on an object in the production environment. With the help of the reference image, the wizard will perform a plausibility check. This will include a check whether a feature is present several times in the image. The error message "*Plausiblity check failed: Position deviation too high*" indicates that you have to mask additional features. For this, you can define additional mask areas or you can reduce the mask area using smaller features. Alternatively, you can reduce the search area via the AOI.

NOTE

Do not choose features which can move, e.g. line segments on a line

To get an optimum search result, you should use features, which spans an angle of 90°.

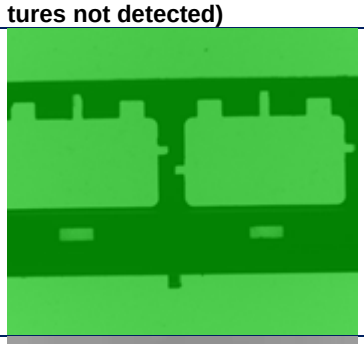
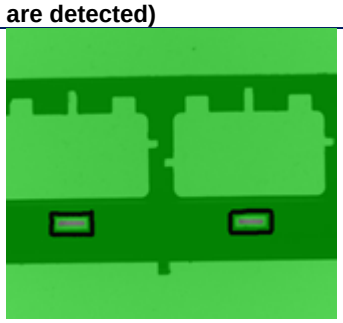
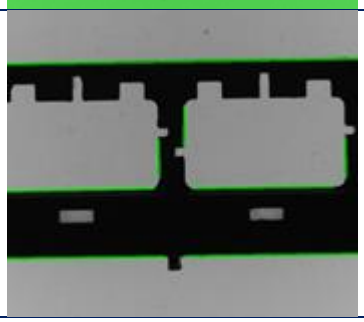
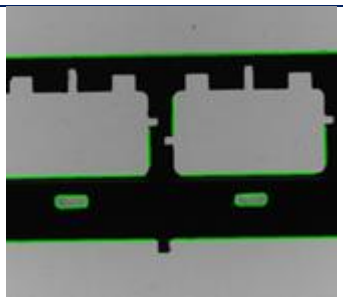
	Variant A ✔ (very stable)	**Variant B ✘ (unstable)**	**Variant C ✘ (unstable)**
Mask			
Determined features			
Production environment			
Description	Significant defined object position	Several possible object sources (feature is not unique, multiple similar features in the image)	Found position can move on the object

Size of the feature

The tool detects automatically major features in the defined mask area. The size of the mask specifies the minimum size of the feature: the smaller the mask, the smaller the feature can be.

The wizard will show the relevant determined features after the mask was defined. If the desired feature is not shown correctly, you can reduce the mask in the previous step. Especially with different sized features you should define these features with the help of different mask regions. In this case you have to specify the size of the feature for each mask region individually.

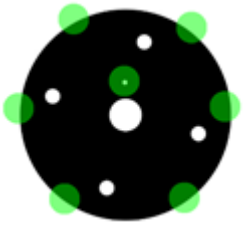
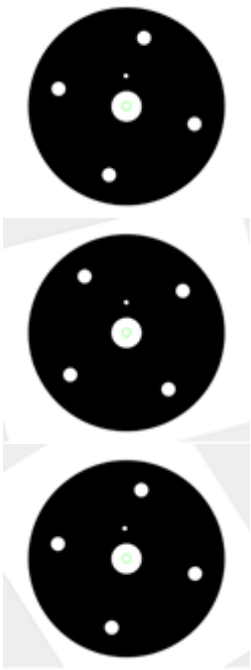
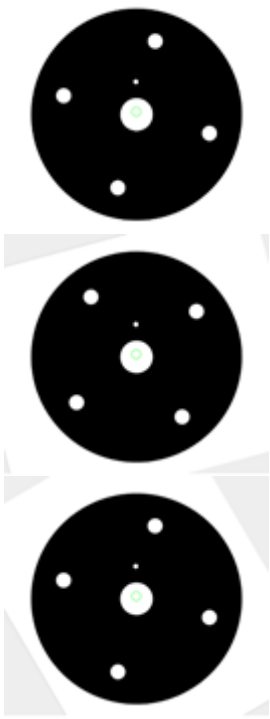
- Depending on mask size: Large mask area -> large feature
- Sizing for each mask area individually

	Variant A (mask is whole image, small features not detected)	**Variant B (explicitly masked small features are detected)**
Mask		
Determined features		

Determination of the rotation

To achieve a good rotation compensation quality for each part, you should attach importance to a uniform distribution of features for the determination of the rotation and position.

The position of the disc is important in the following example. The orientation is given by the small hole next to the disc center, so it is important that this small hole always stays in the same position. The four outer holes however should not influence the orientation and may vary.

	Variant A (unstable)	Variant B (unstable)	Variant C (stable)
Mask			
Determined features			
Production environment			
Explanation	Excessive amount of wrong features, which are used to determine the rotation. The orientation of the dics is completely wrong (small hole not always in same position)	Excessive amount of features, which are used to determine the rotation. The orientation of the disc is better, however, there are angle deviations (degree range, small hole not exactly in same position).	Correct amount of feature types.There are only circle segments marked to detect the position. The disc is oriented with the best quality (small hole always in same position).

Effects in combination with the optics

The accuracy of "**Find object**" is a result of the image acquisition quality, i.e.:

- Lens distortion for different object positions
- Modification of the object image size
- Influences according to the object height

You can improve the accuracy with

- high-quality optics and/or
- telecentric optics.

5.3.2 Check object

Task

By being trained to recognize good and bad elements, the tool can identify variations regarding borders or brightness for test samples.

Requirement

The object must have been orientated in the image with the "**Find object**" tool.

Wizard

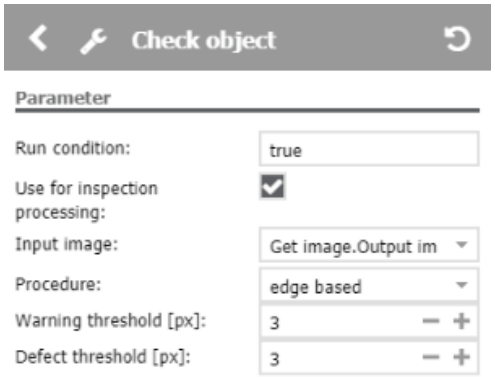
The tool is trained using example images. Some good elements and, as an option, some bad elements, are displayed.

Page	Explanatory text	Button	Comments
1 - Introduction	This tool checks an object section on modified image elements against learned samples. The tool only functions on orientated objects. Therefore, in the inspection program process, a "**Find object**" tool needs to be used first. On the wizard pages that follow, you can configure the tool interactively.		
2 - Delete training data	You can train the tool from scratch or acquire other training images to improve tool stability. If you want to delete existing training data, click on "**Delete training data**". In this case, you are automatically forwarded to the next page of the wizard.	**Delete training data**	Deleting training data enables you to reset the tool.
3 - Learn good parts	On this page, the tool learns the permitted image variances for the objects, by several reference objects being displayed. Position several reference objects one after the other and confirm each with "**Learn good element**". To restart the learning process, you can switch to the previous page at any time and delete the training data there. Then click on "**>**" to go to the next page of the wizard.	**Learn good element**	By displaying various good elements, the tool learns the extent to which various elements differ and thus specifies the range in which the variation of an element may lie.
4 - Learn bad parts	On this page, you can present the tool with bad elements to improve the training result. Position several bad elements one after the other and confirm each with the "**Learn bad element**" button. Then click on "**>**" to go to the next page of the wizard.	**Learn bad element**	By displaying one or more bad elements, the range in which an element is recognized as bad is restricted. The degree of selectivity between good and bad elements is increased. If faults at an element can occour on more than an single location, at least one bad element needs to be displayed per error location.
5 - Define areas to check	On this page, the tool learns the areas of the element that need to be checked for variations. Remove the highlighting in areas that do not need to be checked for variation, so that the only areas highlighted are those where the variance implies a bad element. In the toolbar, you can switch between eraser (remove highlighting) and paintbrush (extend highlighting). Then click on "**>**" to go to the next page of the wizard.		

6 - Finish wizard	Configuration of the tool is now complete. On the next page, you can check and amend parameters set automatically. Click on "**Finish wizard**" to close the wizard.	**Finish wizard**	

The wizard generates a parameter record, which you can also amend manually using tool settings.

Parameters



Parameters	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, the outputs of this tool will be set to default values: • Arithmetic types: 0 • Strings: empty string • Images: empty images	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool
Input image	The image on which the operation should be executed.	Image

Procedure	The procedure with which the objects should be compared.
	• *edge based*
	  ○ robust against fluctuating light conditions
	  ○ comparison of structured surfaces (e.g.: coin minting checking)
	• *intensity based*
	  ○ only for homogeneous lighting scenarios
	  ○ comparison of gray scale value ranges (e.g.: lamp test)
		Enum ("edge based", " intensity based")

Warning threshold [px]	Only variances beyond this set threshold are determined.	Int
Defect threshold [px]	Only variances beyond this set threshold are valued as errors. Smaller errors are only highlighted in color.	Int
Sensitivity [%]	Defines how sensitive the algorithm reacts on changes of intensity. If the sensitivity is too low major defects are not detected. If the sensitivity is too high even small changes in intensity (such as noise or changing illumination) may be interpreted as defects.	Int

Results

Result	Description	Data type	Default value	Tolerance adjustable
Tool processing	• Successful: No errors occured. • Failed: At lease one of the result properties is outside the set tolerance limits or a general error occurred.			✗
Results report	Text description of the cause of the error.	String		✗
Execution time [ms]	The time this single tool needs for execution	Double	0	✗
Size of the largest defect	Size of the largest defect in pixels.	Int	0	✓
Areas with warnings	Output of areas, which exceed warning threshold and are below defect threshold.	Region	Blank region	✗
Areas with defects	Output of areas, which exceed defect threshold.	Region	Blank region	✗
Number of defects	Number of cdefects found.	Int	0	✓
Number of defects and warnings	Sum of the number of found defects and warnings.	Int	0	✓
Total area of defects	Sum of the area of found defects.	Int	0	✓
Total area of defects and warnings	Sum of the total area of found defects and warnings.	Int	0	✓

Expert knowledge

Requirements

Illumination

To avoid shadow and reflection effects, a homogeneous illumination should be used for the object to be tested; also if the tool "**Find object**" has to orientate the object.

Type of illumination	**Homogeneous**, e.g. with diffuse incident light through frosted glass screen, where the object is located
Image series	
Usable for procedure "brightness-based"	✓
Type of illumination	**Inhomogeneous**, e.g. with single point source of light above the area, where the object is located
Image series	
Usable for procedure "brightness-based"	✗

Pose adjustment with the Tool "Find object"

Choose details, which are available in good parts and bad parts with the same forms. Please have a look at the section "Search details" in the description of the Tool "**Find object**".

Training of good and bad parts

In order that the tool can differentiate the good from the bad parts, it is necessary to train at least two good parts. With it, the tool determines the allowed bandwidth of the surface details, which the object must have to be indicated as a good part. With a mask you can limit the area where the tool looks for details.

The tool examines an internal threshold by analyzing the masked area of the bad part images. This threshold is used to differentiate between good and bad parts. The tool is able to distinguish more reliably between good and bad parts when showing multiple bad part images. However, the tool now tends to specify objects as good parts if they have not been trained as bad parts explicitly. For this reason we recommend to use a mask in the areas of the image in which you can find the differences between good and bad parts.

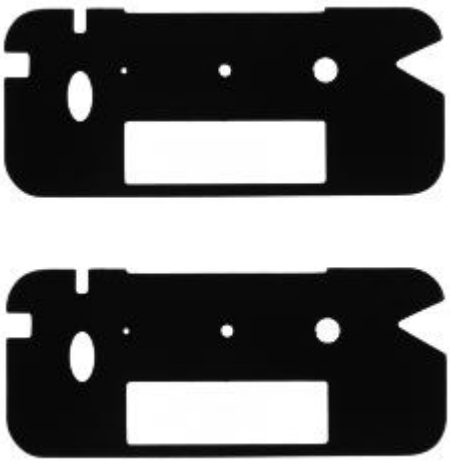
NOTE

Several bad part images will be useful, if you can train the tool with nearly all different scenarios of bad parts.

In the following we will show three approaches for the task to ensure the presence of three holes in a stamping.

Only good part images + Mask over the whole area

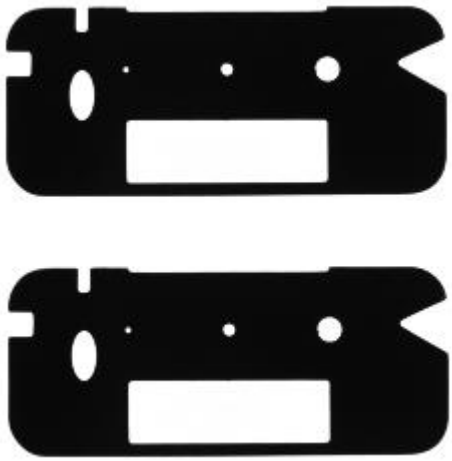
The tool determines the whole detail bandwidth using two training images.

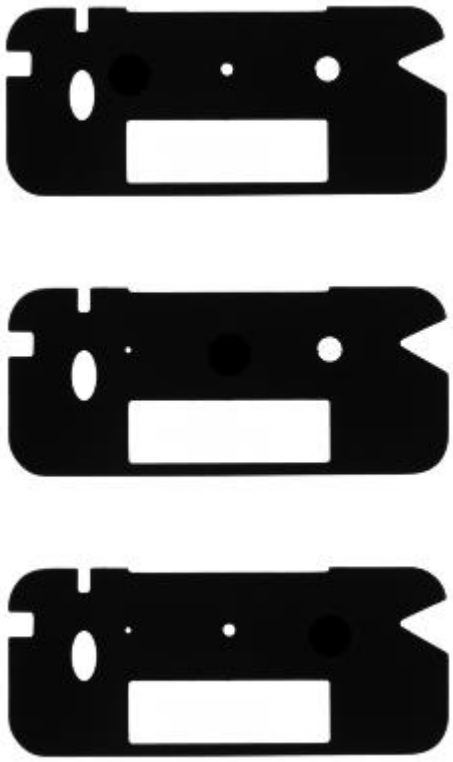
Description	**Only good parts** ✔
Mask	
Good parts	
Bad parts	

Good and bad part images + Mask with error location of each possible error

The tool is trained with at least two good part images. Additionally, for each error (here: covered hole) a bad part image will be taken. The mask is used to indicate the error location of each possible error.

In this case, the tool will check, if all three holes are present. If there are differences outside the mask, no error will be detected.

Description	**Good parts and bad parts A**
Mask	
Good parts	

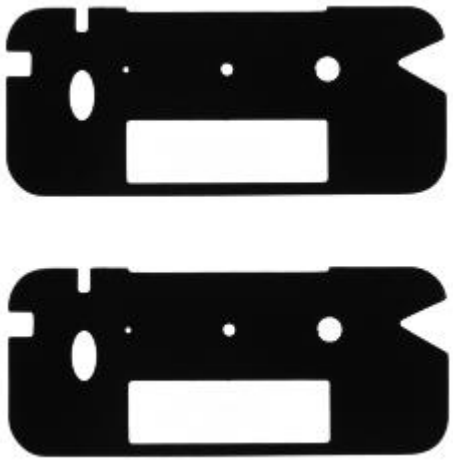
Bad parts	

Good and bad part images with all errors + Mask with error location of each possible error

The tool is trained with two good parts and one bad part. The bad part will show all possible errors and the mask marks all error positions.

In this case, the tool cannot train correctly which will lead to possible impairments of the tool's selectivity.

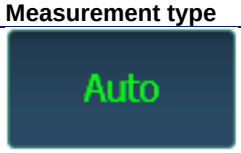
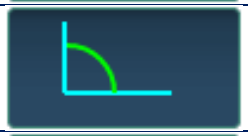
Description	Good parts and bad parts B
Mask	

Good parts	
Bad parts	

5.3.3 Measure object

Task

Measuring objects in the image. The tool is able to measure distances, angles and radii. Possible measurement types are:

Measurement type	Description	Value range
	The measurement type is determined automatically based on the selected edges Distance measurement will be chosen if two lines are nearly parallel (angle < 10°), otherwise an angle measurement will be made	
	Distance between two lines The lines may be edges in the image or arrows of distance measurements (check *expert knowledge* for further details)	The range of values can be defined arbitrarily setting the nominal value and the +/- tolerances.
	Angle between two lines The lines may be edges in the image or arrows of distance measurements	Angle values are limited to the range of -180° to 180°.
	Minimum distance between line and circle The line may be an edge in the image or an arrow of a distance measurement	The range of values can be defined arbitrarily setting the nominal value and the +/- tolerances.
	Distance between line and point The line may be an edge in the image or an arrow of a distance measurement The point may be a circle center or an intersection between two lines	The range of values can be defined arbitrarily setting the nominal value and the +/- tolerances.
	Distance between two points The points may be a circle centers or intersections between two lines each	The range of values can be defined arbitrarily setting the nominal value and the +/- tolerances.
	Distance circle to point The point may be a circle center or an intersection between two lines This measurement type can also be used to determine the radius of a circle	The range of values can be defined arbitrarily setting the nominal value and the +/- tolerances. For radii, the value is limited to the range from reference value*0.5 to reference value*1.5.
	Minimum distance between two circles	The range of values can be defined arbitrarily setting the nominal value and the +/- tolerances.

The reference value is the distance/radius/angle that has been trained within the wizard. Nominal value and tolerances have to be within the valid value range as measurements are only found within this range. Nominal value and tolerances can only be changed if they are within the valid range.

The default value range of all circle-circle, circle-line and line-line measurements is +/10% of the reference value. But, the range can be changed arbitrarily. Do not choose the range too big in order to avoid a mix-up with other lines or circles lying in the tolerance region. Note, that huge tolerance ranges can lead to an increased processing time.

NOTE

For measuring circles it should be noted that the nominal value is depending on the circle's diameter used for creating this particular measurement. The nominal value can range from 50% to 150% of the circle's reference diameter. Circles outside of this range won't be detected. Furthermore, for setting a nominal value of a circle measurement, the tolerance needs to be considered. As an example, for a measurement set up with a reference circle of 100mm, the nominal value can be set to 60mm, if the lower tolerance is not bigger than 10mm.

Wizard

The tool is taught by using a reference image and showing where to measure.

Page	Explanatory text	Button	Comments
1 - Introduction	This tool is able to measure distances, angles and radii on an object.		
2 - Delete measurements	You can train the tool from scratch or teach other measurements. If you want to delete existing training data, click on "**Delete training data**". In this case, you are automatically forwarded to the next page of the wizard.	**Delete training data**	All configuration data is deleted.
3 - Reference object	Capture a reference image to generate measurements on it.	**Capture**	In order to be able to define measurements the image should have high contrast in these areas.
4 - Edit mask	To limit the measurement to specific objects click on "**Edit mask**".	**Edit mask**	When editing the mask, all existing measurements will be removed.
	Define object areas with the brush (mark /erase). Mark the objects you want to measure. In the toolbar, you can switch between eraser (remove highlighting) and paintbrush (extend highlighting).	**Delete mask** **Set maximum mask** **Reset mask**	The area that should be used for measuring the object can be edited here. Using the zoom will change the size of the brush/rubber. Using the buttons will expand the mask to the whole image, remove it completely or reset it to the initial state. Areas which are marked by the mask are more intensely examined for suitable measure edges.
5 - New measurements	Select the desired measurement type by clicking on the respective icon. Select two edges to generate a measurement between them. You can define an unlimited number of measurements. Go to the next wizard page to finish creation of measurements. Select the level of detail for lines and circles in the reference image	***Slider*** **Fine/Coarse**	Selectable objects (edge in the image, arrow from a distance measurement, or intersection point of an angle measurement) are displayed in cyan. They differ depending on the measurement type chosen. Click on the first object. It will be rendered in yellow after selecting it. In order to deselect it just click a second time on it. Click on the second edge in order to complete the measurement. The measurement is created. Repeat these steps to define multiple measurements. If two lines are selected, depending on the position of the lines either the distance between the first line to the clicked position on the second line is measured or the angle between the two lines (check *expert knowledge* for further details). To measure the radius of a circle, select the center and the circular edge. To start a measurement from the intersection of two lines, create an angle measurement and afterwards select the point of intersection. By using the slider you are able to show fine edges. It is also possible to measure from coarse to fine edges.

6 - Finish wizard	Click on "**Finish wizard"** to close the wizard.	**Finish wizard**	Every measurement defined within the wizard are shown in the results of the tool. They can also be deleted there.

NOTE

If more edges are expected to be detected, change the detail level and mark the edges via the mask. If this does not help consider increasing the contrast of the image (change exposure or aperture) or magnifying the image (reduce the distance from the camera to the object or choose a lens with longer focal length).

Detail level and measurement type features

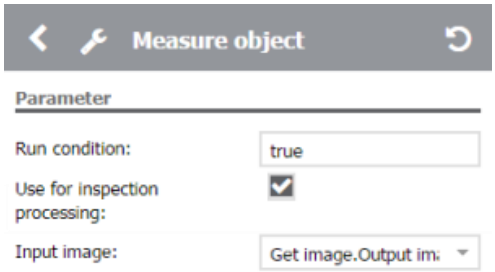
Two detail levels are available to extract and measure edges.

- Choose detail level **Fine** to measure small or short edges or edges with a high contrast. Detail level Fine will extract more short lines.
- Choose detail level **Coarse** to detect blurry contours. Then, slight deformations like dents on the contours are tolerated.

Marking a measure edge using the mask within wizard mode leads to a more intense examination for suitable measure features.

Lines	• Edges must exhibit a length of at least 20 pixels for detail level **Fine** or 40 pixels for detail level **Coarse** respectively for being detected as a single line segment. • For line detection there is always a small gap between the line and the limiting corner. For detail level **Coarse** this gap is bigger.
Circles	• Circles are detected exhibiting a minimal diameter of 20 pixels. • Circles must be sufficiently covered by contours to be detected robustly. • Small circles have to be more or less full circles, like 70 to 90% coverage. For bigger circles a coverage of 50 to 60% can be sufficient. • Circle measuring is still possible even if multiple contour segments in the image are not connected to each other. • Basically, a circle has to be of an exact circular shape. Therefore, the viewing direction has to be orthogonal with respect to the circle. Otherwise the circle degenerates to an ellipse in the image. Only small deviations from a circular shape are tolerated. This still holds even for a calibrated setup with a dedicated measurement plane.

Parameters



Parameter	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, the outputs of this tool will be set to default values: • Arithmetic types: 0 • Strings: empty string • Images: empty images	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool
Input image	Image on which the operation should be executed.	Image

NOTE

In the case that the image section is changed in the "**Set up camera**" tool, the "**Measure object**" tool has to be retrained.

Results

Result	Description	Data type	Default	Tolerance adjustable
Tool processing	• Successful: No errors occurred. • Failed: The tool is not trained, one of the measurement could not be found or is out of tolerance.			✗
Results report	Text description of the cause of the error	String		✗
Execution time [ms]	The time this single tool needs for execution	Double	0	✗
Created measurements	All of the measurements defined in the wizard and their current value. The measurement can be deleted by using the trash can button.	Double	0	✓

Calibrate result

In the result table, you can calibrate the result via the button , i.e. you define the metric length of the measured distance. You reset the calibration by entering the value of zero, i.e. the results will be presented in pixel distance again.

If you have already calibrated a camera in "**Set up camera**", measurements will be done in mm and the described manual calibration will not work.

Display measurement result

The measured result (distances, angles and radii on an object) is shown in green when the measured result value is within the tolerance range. If the measured result value is out of the tolerance value, then the measured result is shown in red.

Display edge search areas

Select a measurement result row to display the search areas for all lines of this measurement. Selecting a measurement is possible in config mode or in the wizard step "New measurements". Edge search areas are displayed in brown as soon as a suitable edge can be detected, in red, as soon as no suitable edge is available. Beside the edge search area the position of the taught edge will be displayed.

Change search area

The search area can be extended, shrunk, or moved. In doing so, the taught line (highlighted by displaying it dashed during modification of the search area) must, at least partially, remain inside the search area.
After a new measurement, the search area will be determined via the tolerance values of this measurement automatically. As soon as the search area of the measurement is changed manually, the automatic adjustment of the search area of this measurement will be deactivated.

Change polarity

To change the expected polarity (direction of dark to light transition) of edges, there is a button in the search area. By clicking this button, the mode is changed. This button is brown, as soon as a suitable edge can be detected and the button will be red if no suitable edge can be found.

Mode	Button	Description
Dark-Light		Only edges with a transition from dark to light are considered as possible candidates.
Light-Dark		Only edges with a transition from light to dark are considered as possible candidates.
All-Edges		All edges are considered as possible candidates.

The button will be displayed rotated according to the orientation of the search area and the detected line.

Change search direction

To change the search direction for edges there is a further button in the search area. By clicking this button, the direction is changed. This button is brown, as soon as a suitable edge can be detected and the button will be red if no suitable edge can be found.

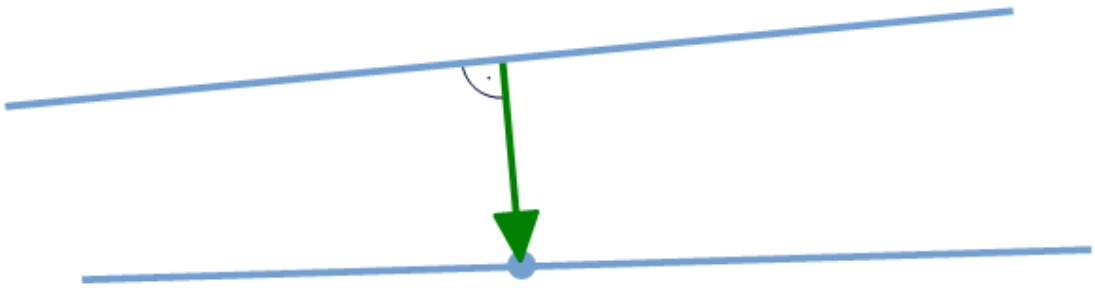
Mode	Button	Description
Center		The edge is chosen, which is next to the taught line position.
Left-Right		The search is made from left to right and the first suitable edge is chosen.
Right-Left		The search is made from right to left and the first suitable edge is chosen.

The button will be displayed rotated according to the orientation of the search area.

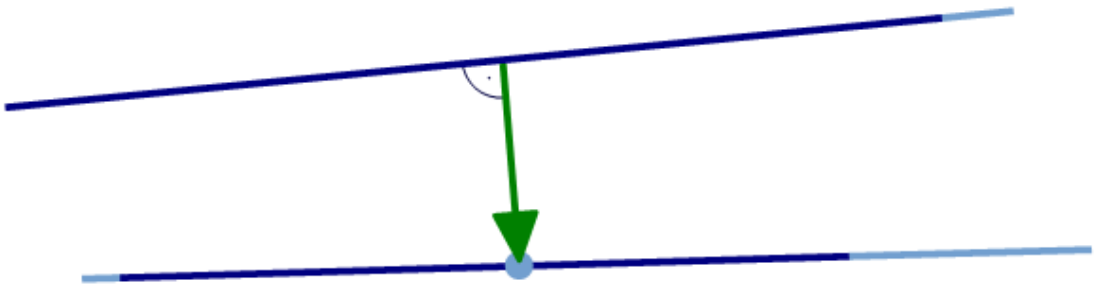
Expert knowledge

Distance measurement between two lines

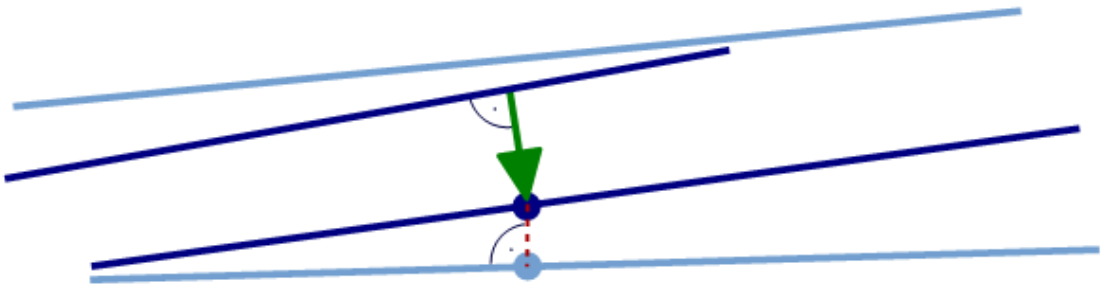
Distance measurement between two parallel lines is mathematically well-defined. In practice images rarely contain lines that are exactly parallel. In order to determine distances between two nearly parallel lines, the distance (green arrow) between the first line to the click position (light blue point) on the second line is measured.



If lengths of the current measure lines (dark blue) differ from the lengths of the reference lines during measure creation in the wizard (light blue), the position of the measure point does not change within the image.

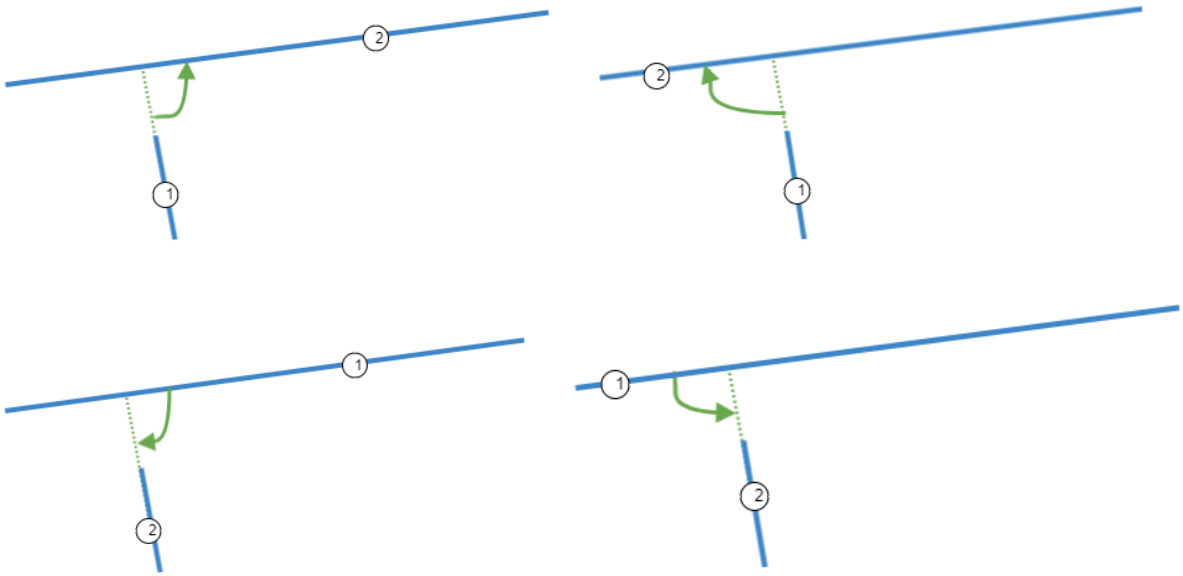


If positions of the current measure lines (dark blue) differ from the positions of the reference line (light blue), the intersection between the measure line with a guide line (red dashed) that originates from the clicked position (light blue point) perpendicular to the reference line is used as the measure point (dark blue point). The measurement (green arrow) is carried out as the distance between the first measure line to this measure point on the second line.

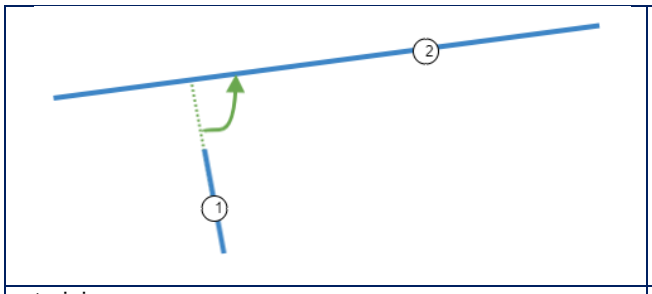
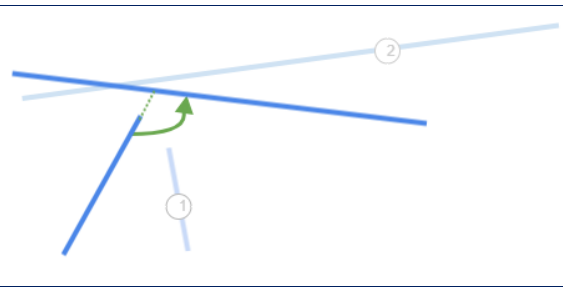


Angle measurement between two lines

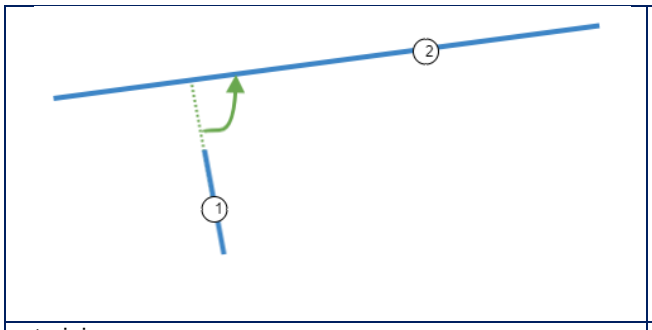
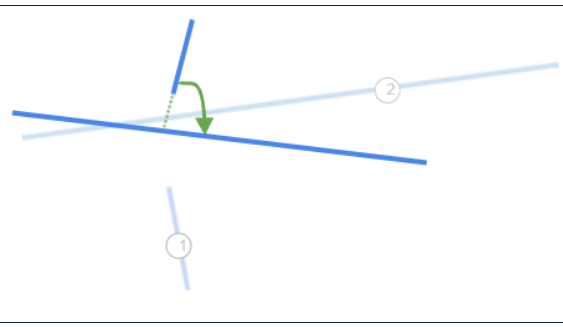
During the training, a click on two lines will initiate a measurement when making an angle measurement. In doing so, the order of the clicks defines which angle is measured.



If the position of the current measurement lines (dark blue) is different to the reference lines (light blue), the angle will be measured in the same way:

	
training	measurement

Angles with a counterclockwise measuring direction result in positive values, angles with a clockwise measuring direction result in negative values. Angles are analyzed in the range between -180 to 180 degrees, i. e. if an angle becomes e. g. higher than 180 degrees, the analysis changes to the corresponding angle from -180 degrees:

	
training	measurement

5.3.4 Check blobs

Task

Analyzes blobs in an input image. Blobs are coherent bright or dark regions in image that can be created and evaluated.

Wizard

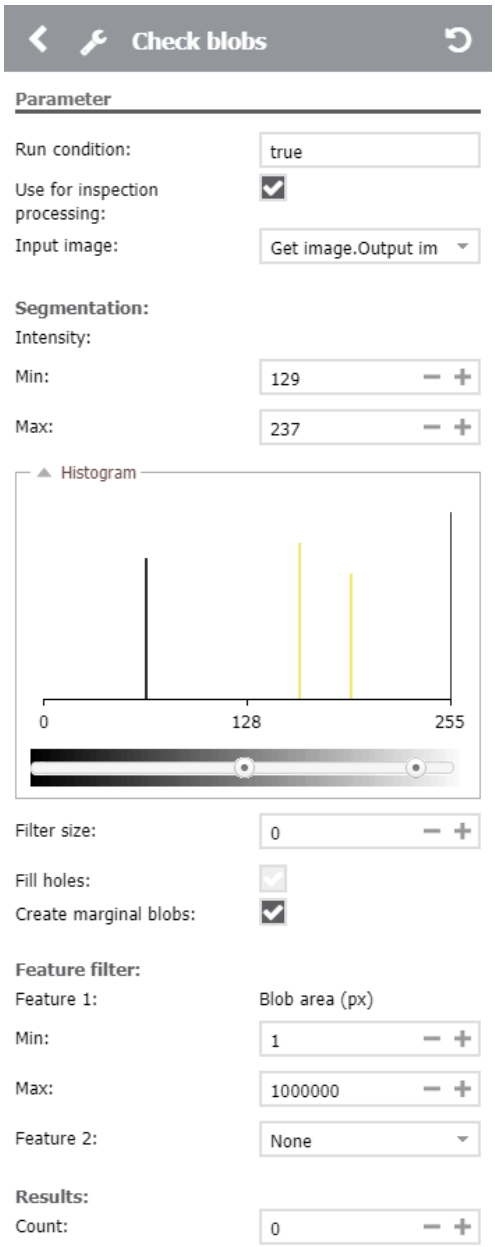
The wizard is used to define the sections in the image, where the blobs are analyzed.

Page	Explanatory text	Button	Comments
1 - Define areas to analyze / Finish wizard	Define areas with the brush (mark /erase). Click on "**Finish wizard**" to close the wizard.	**Delete mask** **Set maximum mask** **Reset mask** **Finish wizard**	The sections in the image, where theblobs are analyzed, can be edited here. Using the zoom will change the size of the brush/rubber. Using the buttons will expand the mask to the whole image, remove it completely or reset it to the initial state.

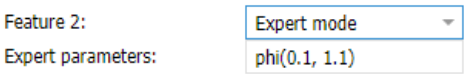
NOTE

The area of interest in inspection programs, which were created by an older version of BVS Cockpit (< 2.5), will be converted automatically when the inspection program is loaded.

Parameters



For expert analysis mode, which is called via "Feature 2", the following parameters are displayed:

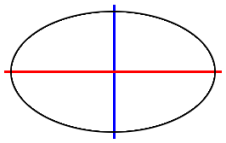
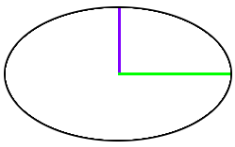
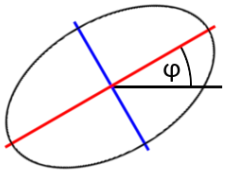


Parameters	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, the outputs of this tool will be set to default values: • Arithmetic types: 0 • Strings: empty string • Images: empty images	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool
Input image	Image on which the operation should be executed.	Image
Segmentation - Intensity - Min	Lower gray value boundary. Alternatively, these values can be set via the histogram.	UInt16
Segmentation - Intensity - Max	Upper gray value boundary. Alternatively, these values can be set via the histogram.	UInt16
Histogram	Fold-out window with a histogram of the gray values distribution. Only the gray values in the masked area are considered. The minimum gray value can be set with the left slider. The maximum gray value can be set with the right slider. Alternatively, the value can also be set via the input masks of Segmentation - Intensity - Min or Segmentation - Intensity - Max. The segmentation area is highlighted in the histogram in the same color as the blobs in the image. **NOTE** If the intensities 255 or 0 are the highest peaks in the histogram, these are only displayed max. 20% higher than the next higher value. A color image is converted to a gray image before calculation. The histogram shows the relative frequency in percent. Relative frequency means absolute frequency of gray values divided by the masked area of the image as floating point numbers.	Graph
Segmentation - Filter size	Filter size for blob pre-evaluation. • for values <= -0.5 a closing filter is applied (connects close blobs) • for values >= 0.5 an opening filter is applied (separates blobs with a thin connection) • between -0.5 and 0.5 no filter is applied.	Double
Segmentation - Fill holes	Determines, whether holes, created by the segmentation belong to the blob or not.	Bool
Segmentation - Create marginal blobs	Marginal blobs frequently cause difficulties, since their features can not consider parts of the blobs outside the image. This parameter determines thus, whether blobs that touch the border of the image or of the area of interest are included into blob analysis or not.	Bool
Feature filter - Blob area (px) - Min	Sets the minimal size of a blob (pixel count). Smaller blobs are discarded.	UInt16
Featuer filter - Blob area (px) - Max	Sets the maximal size of a blob (pixel count). Larger blobs are discarded.	UInt16
Feature filter - Feature 2	Sets additional evaluation options for the blobs. **Analysis type** / **Description** None / No additional evaluation Center of gravity x / X coordinate of the center of gravity for a blob Center of gravity y / Y coordinate of the center of gravity for a blob	Enum

	Contour: Contour length of a blob. Contours of enclosed holes are ignored. Generates analysis values >= 0. Circularity: Similarity of a blobs with a circle. Higher analysis values mean higher similarity of the blob with a circle. Generates analysis values from 0 to 1. Max. Diameter: Maximal distance between two boundary points of a blob Angle (-180°...180°): Angle of the semi-major axis of the enclosing ellipse (*) measured counterclockwise from the x-axis. Generates analysis values from -180 to 180 Angle (0°...180°): Angle of the semi-major axis of the enclosing ellipse (*) measured counterclockwise from the x-axis. Generates analysis values from 0 to 180 Major radius: Length of the semi-major axis of the enclosing ellipse (*) Minor radius: Length of the semi-minor axis of the enclosing ellipse (*) Expert mode: Allows the manual configuration of the additional evaluation.	
Feature filter - Feature 2 - Min	Sets the minimal filter value of a blob. Blobs with smaller values are discarded.	Double
Feature filter - Feature 2 - Max	Sets the maximal analysis value of a blob. Blobs with larger values are discarded.	Double
Feature filter - Feature 2 - Expert parameters	List of filter/analysis types and their parameters for the expert mode, separate by spaces. Analysis types are defined in the format ***analysistype(min, max)***. Example: "*ra(0.1, 1.1) rb(-0.2, 0.1)*" Allowed values and their meaning can be looked up in the HALCON documentation (keywords regions / features). The first analysis type is used as result analysis value.	String
Results - Count	Maximum number of blobs for detail analysis. Allowed value range is 0 to 255. • If this is set to 0, all blobs are counted that fulfill the criteria defined above. • If this is set to a value greater than 0, additionally detailed result values will be created for the blobs found.	UInt16
Results - Sort	Determines which blobs are used for detail analysis. Blobs are sorted and up to < **Results - Count** < selected.	Enum

Sort mode	Description
Blob area (large -> small)	Blobs are sorted by size descending
Blob area (small -> large)	Blobs are sorted by size ascending
Feature 2 (large -> small)	Blobs are sorted by feature 2 in descending order
Feature 2 (small -> large)	Blobs are sorted by feature 2 in ascending order
Center x (left -> right)	Blobs are sorted by the x coordinate of their center of gravity from left to right
Center x (right -> left)	Blobs are sorted by the x coordinate of their center of gravity from right to left
Center y (top -> bottom)	Blobs are sorted by the y coordinate of their center of gravity from top to bottom
Center y (bottom -> top)	Blobs are sorted by the y coordinate of their center of gravity from bottom to top

(*) Explanations to enclosing ellipses:

Enclosing ellipse	Semi-major and -minor axis	Length of semi-major and -minor axis	Counterclockwise rotation of the ellipse
	 **Semi-major axis** **Semi-minor axis**	 **Length of semi-major axis** **Length of semi-minor axis**	

Result tables

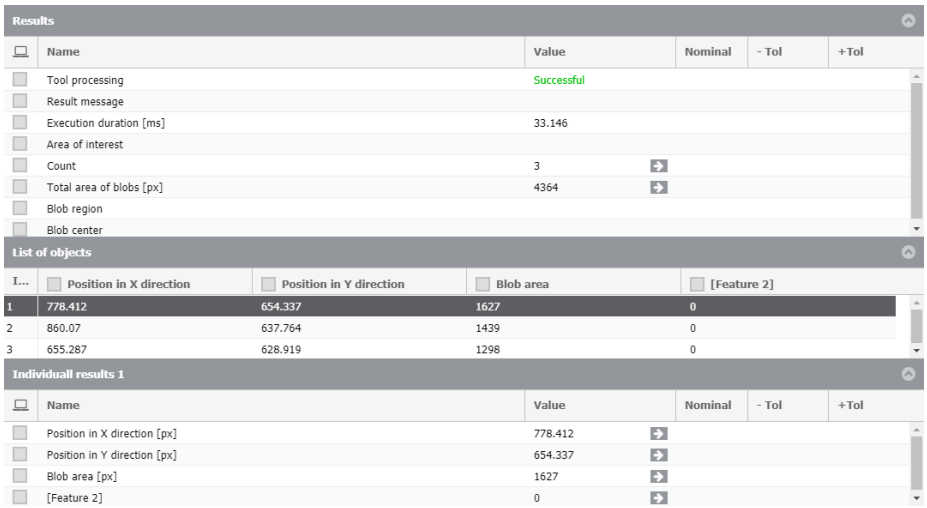


Table 1: Results

This table contains all general results belonging to the tool.

Result	Description	Data type	Default value	Tolerance adjustable
Tool processing	• Successful: no errors occurred. • Failed: an error occurred during evaluation, i.e. invalid expert analysis settings.			✗
Results report	Text description of the cause of the error	String		✗
Execution time [ms]	The time this single tool needs for execution.	Double	0	✗
Area of interest	The area where blobs have been searched.	Region	Same as the input area of interest	✗
Count	Number of found blobs	UInt16		✓
Total area of blobs (px)	Sum of the area of all blobs. If Count is set to a value other than 0, only the blobs which are part of the evaluation are considered when calculating the total area	UInt64	0	✓
Blob region OK	Enables drawing of the region of all 'OK' blobs in monitor mode	Region		✗
Blob region not OK	Enables drawing of the region of all 'not OK' blobs in monitor mode	Region		✗
Blob region not analyzed	Enables drawing of the region of all 'not analyzed' blobs in monitor mode	Region		✗
Blob center OK	Enables drawing of the centers of all 'OK' blobs in monitor mode	Region		✗
Blob center not OK	Enables drawing of the centers of all 'not OK' blobs in monitor mode	Region		✗

Table 2: List of all found objects

This table contains the list of all found objects. Is shown as soon as "Results - Count" is above 1.

Result	Description
Blob_1	Represents a found blob with its number. The properties of each bolb are displayed columnwise in this table. When a blob is selected in this table, its properties are displayed in detail in the third table.

Table 3: Properties of the selected object

This table contains the properties of the result object that has been selected in table 2. Is shown as soon as "Results - Count" is above 1.

Result	Description	Data type	Default value	Tolerance adjustable
Position in X direction [px]	The X position of the current blob	Double		✔
Position in Y direction [px]	The Y position of the current blob	Double		✔
Blob area [px]	The size (area) of the current blob	UInt32		✔
[Feature 2]	The calculated value of feature 2 for the current blob.	Double		✔

Color coding

Color of the blobs

For the benefit of the user, the blob tool represents blobs in different colors. This enables easy recognition and analysis of blobs which are within/out of their respective tolerances.

- Case 1: When the expected number of blobs is set to zero.
  - In this case, there are no blobs to be analysed. Therefore all the found blobs are displayed in yellow indicating "Blobs not analyzed".
- Case 2: When the expected number of blobs is set to more than zero.
  - In this case, blobs are analyzed and checked if they are within their tolerances.
  - Blobs which are within their tolerances are displayed in green color.
  - Blobs which are out of their tolerances are displayed in red color.
  - When the found blobs exceeds the expected number of blobs, the excess blobs are called "Blobs not analyzed". These blobs are displayed in yellow. They do not influence the tool result.

Color of the AOI (Area of interest)

The color of the AOI is green when all the anaylzed blobs are within their respective tolerances and all the general tool results are also within their respective tolerances.

Even if one of the general tool results or the blob results fall out of tolerance, the AOI is displayed in red.

5.3.5 Read code

Task

The tool recognizes, trains and reads 1D or 2D codes in the image. The tool searches for codes within the area of interest and detects them even if they are rotated or in different sizes.

Wizard

The tool is trained on various code types, by images with the code types to be recognized being displayed in the wizard. Further images with the same code type can be used to train the tool in order to improve the recognition rate.

Page	Explanatory text	Button	Comments
1 - Introduction	This tool enables bar and matrix codes, such as e.g. code 39, QR code and EC200 to be recognized, analyzed and trained. On the wizard pages that follow, you can configure the tool interactively. To save the training data permanently, all steps of the wizard must be completed.		
2 - Delete training data	You can train the tool from scratch, teach other code types or acquire additional training images. If you want to delete existing training data, click on "**Delete training data**". In this case, you are automatically forwarded to the next page of the wizard.	**Delete training data**	All configuration data is deleted. If "*Use training data*" is activated in the expert parameters, it will be deactivated.
3 - Add and train codes	Here, the tool learns which code types need to be recognized. Training allows for faster and more reliable recognition of known code types. To add a new code type and train the tool, 1. position a code in front of the camera and 2. click on "**Search code**". If the code required is not highlighted, 3. limit the search area by using the blue box, by rescaling it to the corner points of the frame, or repositioning it. If the code that you want to add and train is highlighted, 4. click on "**Add & train**". On the settings page, you can manually add and remove code types.	**Search code** **Add & train**	"**Search code**" searches the search area for all supported code types. This can take a few seconds. Only the code types that actually need to be found should be active. Otherwise, there is a risk of incorrectly finding codes in the image.
4 - Finish wizard	You have set up the code reader successfully. The trained data is stored permanently when you click on "**Finish wizard**". You are then taken to the settings page. Click on "**Finish wizard**".	**Finish wizard**	The type of code to be found and the training of individual code types is stored permanently at this point.

The code type and comparison text settings entered in the wizard can be edited on the settings page. If necessary, a code type can be retrained in the wizard, if a characteristic of a code type cannot be read in operation.

Parameters

Parameters	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, the outputs of this tool will be set to default values: • Arithmetic types: 0 • Strings: empty string • Images: empty images	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool
Input image	The image on which the codes are being searched.	Image
Area of interest (x, y, width, height)	Area of the image in which the codes are being searched.	[Double]
Code types 1D	List with types of bar codes (1D) that are searched in the image. All other bar codes are ignored during the search.	[Bool]
Code types 2D	List with types of matrix codes (2D) that are searched in the image. All other matrix codes are ignored during the search.	[Bool]
Expected text	The included text of all codes are compared with the expected text. The tool processing is successful if at least one code is found containing the expected text. Deviating codes are marked with red results boxes. The expected text must contain only standard ASCII conformant characters. You can use characters, such as ? and * as wild cards. Thus, for example, you would use 17* to ensure that the text read begins with 17. The preset parameter "*" accepts any codes.	String
Expected number of codes	The maximum number of codes the tool should try to find. If the text comparison is disabled (expected text is set to "*") the code retrieval is stopped as soon as enough codes have been found.	Int

<table>
<tr><td></td><td>

NOTE

Increasing *Expected number of codes* beyond ten may impair reactivity of BVS Cockpit in your web browser temporarily.

</td><td></td></tr>
<tr><td>**Sort mode**</td><td>

Determines the criteria by which found codes will be sorted.

<table>
<tr><th>Sort mode</th><th>Description</th></tr>
<tr><td>Position x (left → right)</td><td>Codes are sorted by the x coordinate of their midpoint from left to right.</td></tr>
<tr><td>Position x (right → left)</td><td>Codes are sorted by the x coordinate of their midpoint from right to left.</td></tr>
<tr><td>Position y (top → bottom)</td><td>Codes are sorted by the y coordinate of their midpoint from top to bottom.</td></tr>
<tr><td>Position y (bottom → top)</td><td>Codes are sorted by the y coordinate of their midpoint from bottom to top.</td></tr>
</table>

NOTE

Sorting takes place after the code search. This means that the results are not dependent on the sort mode. Only the order of results will change with the sort mode.

</td><td>Enum</td></tr>
<tr><td>**Use training data**
Expert parameter</td><td>

As soon as the tool is trained, you can chose this mode. As a result of the specific training, this mode achieves a better reading and detection rate based on the trained codes. However, it is possible that codes are not detected which differ from the trained codes (e.g. size or contrast).
This parameter will be only available, if the tool was trained. If this parameter is active and the training data is deleted, "*Use training data*" will be deactivated and will be available again after a new training.
If this parameter is activated, only the trained code types will be used for the code reading. For untrained code types, the settings in "*Search Mode 2D*" are used.

NOTE

If you only train one 1D code type, the tool will only detect codes, which have similar characteristics like contrast, edge quality and code size. For this reason, it is recommended to train at least two codes of one code type, which cover the possible variation of the code type.

A training with a dark, but high-contrast code and a bright, but blurry code makes it possible to detect codes within this variation range, like a bright code with medium edge quality.

If you want to detect different 1D code types, you have to train all variations of all code types.

</td><td>Bool</td></tr>
<tr><td>**Search Mode 2D**
Expert parameter</td><td>

Determines how 2D codes are searched and effects the reading and detection rate. There are three settings:

- **Robust**:
Recognizes all detectable codes the system knows. However, this mode needs processing power and has a moderate reading rate, but the detection rate is very high.
- **Balanced** (Standard):
This mode is a compromise between *Robust* and *Fast* and allows finding reliably codes although there are slight code differences concerning e.g. the structure.
- **Fast**
Offers a good reading rate. However, it can be possible, that codes are not

</td><td>Enum</td></tr>
</table>

	detected, which parameters like height, width, size, distortion, etc. differ from the usual values. **NOTE** Only the codes specified in "*Code types 2D*" will be search. The chosen search mode does not matter.	
Assess quality *Expert parameter*	Activates/deactivates code quality assessment and is used to specify the norm that is used as the basis for this quality assesment. If *Assess quality* is active, *Read timeout* can't be used. **NOTE** Assessing the code quality will increase the execution duration of this tool significantly.	Enum
Read timeout [ms] *Expert parameter*	Duration of manually selected timeout in milliseconds. Timeout compliance can only be guaranteed if search is either restricted to any selection of 1D code types or a single 2D code type.	Int
Output quality details *Expert parameter*	This flag specifies if the quality details used for the calculation of the overall quality should be presented as individual tool results. This flag is only visible if *Assess quality* is acitvated and only a single code type is selected.	Bool
View code details *Expert parameter*	This flag specifies if additional details of the read codes are computed and displayed as tool results. • Size of the smallest bar code stripe (for 1D code types) or side length of a module (for 2D code types) • 2D code types can be read even if they appear mirrored in an image. Information whether the code is mirrored or not can be computed for all 2D code types.	Bool

NOTE

For stable decoding of **bar codes**, the resolution must be set to **a minimum of 2 pixels per bar code stripe**. If a barcode is not found due to poor resolution, it might help to activate option *Optimized search*.
For reliable assessment of **code quality**, the resolution must be set to **a minimum of 3 pixels per bar code stripe**.
Displaying the size of a bar code stripe can be activated by expert parameter *View code details*.

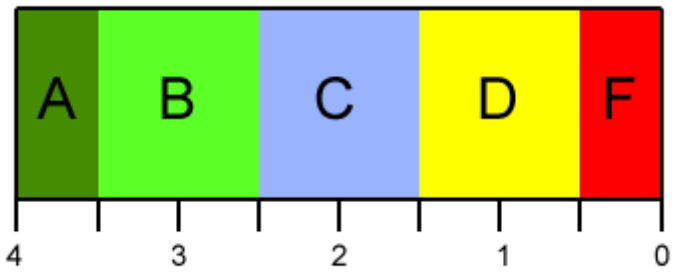
NOTE

For stable decoding of **2D code types**, the resolution must be set to a **minimum of 4 pixels per module**. For PDF417 code type, 3 pixels per module are sufficient. In order to read the smallest code type (**Micro QR Code**, consisting of 11 modules) the code has to have a minimum size of 44 * 44 pixels. If a code is not found due to poor resolution, it might help to activate option *Optimized search*.
Displaying the size of a module can be activated by expert parameter *View code details*.

Code quality

The code quality is determined for 1D cody types borrowing from ISO/IEC 15416. For 2D code types, quality assesment borrowing from ISO/IEC 15415 or borrowing from ISO/IEC TR 29158 can be chosen by *Asses quality*. For PDF417 only ISO/IEC 15415 is supported. Please note, that the norms contain instructions for the measurement setup and the measurement procedure. The computed values are, in terms of the norms, only valid, if these conditions are fullfilled.

The standards define a number of quality characteristics, whereby the total quality is given by the worst of all characteristics. To assess the quality, symbol grades according to the Anglo-American system are used (A, B, C, D, F). As a result, A represents the highest possible quality, whereas F means that the minimum requirement has not been satisfied:



Structure of the quality result string of 1D code types borrowing from ISO/IEC 15416

The string with the format *a (bcdefghi*) indicates the determined quality which has the following meaning:

Index	Characteristic	Description
a	Overall quality	The minimum or worst quality of all characteristics.
b	Decode	Will indicate, if the code can be read or not according to the specification; for this reason, the value can only be A (readable after specification) or F (not readable after specification). Even with the grade F, the code can be readable for the system.
c	Symbol contrast	To conform the contrast requirements of the gray value profile. Describes the difference between the highest and the lowest reflectance values of the gray value profile.
d	Minimal reflectance	Will indicate, if the minimum reflectance value is smaller or equal to the half of the maximum reflectance value. Is this the case, the quality is "A", otherwise "F".
e	Minimal edge contrast	Evaluates the minimal edge contrast of two neighbor symbol elements.
f	Modulation	The measure of the amplitude between two symbol elements specifies how reliable bars and gaps can be distinguished.
g	Defects	Indicates, to what extent irregularities or disturbances occur in quiet zone within the gray value profile.
h	Decodability	Describes, how far the true width of the symbol elements differ from the standard of the barcode types.
i	Additional requirements	Depending on the code type, the ISO/IEC 15416 defines additional requirements like the quiet zone widths, the wide/narrow ratio, the inter character gaps.

Structure of the quality result string of 2D code types borrowing from ISO/IEC 15415

The string with the format *a (bcdefghij)* indicates the determined quality which has the following meaning (all codes besides PDF417):

Index	Characteristic	Description
a	Overall quality	The minimum or worst quality of all characteristics.
b	Contrast	Describes the difference between the highest and the lowest reflectance values of the gray value profile. Calculation of this characteristic differs from calcualation of "Cell contrast" of ISO/IEC TR 29158.
c	Modulation	The measure of the amplitude between two symbols specifies how reliable they can be distinguished. Calculation of this characteristic differs from calcualation of "Cell modulation" of ISO/IEC TR 29158.
d	Fixed pattern damage	Measures and grades any damage to the finder pattern and quiet zone in the symbol.

e	Decode	Binary characteristic of the readability. If it can be decoded, then it will be A otherwise F. In this implementation it will always be A.
f	Axial nonuni-formity	Measures and grades the spacing of the mapping centers and tests for uneven scaling of the symbol along the X or Y axis. Reasons for this behavior can be the print quality or the tilt of the camera.
g	Grid nonuni-formity	Measures and grades the largest vector deviation of the grid intersections, determined by the theoretical position prescribed by the reference decode algorithm and the actual measured result.
h	Unused error correction	Measures and grades the reading safety margin that error correction provides.
i	Reflectance margin	Comparable with the characteristic "Modulation"; also takes into account if a module could be decoded.

For PDF417 the form is a (bcdefg) and has the following meaning:

In-dex	Characteristic	Description
a	Overall quality	The minimum or worst quality of all characteristics.
b	Start/Stop pattern	The quality of the start and stop patterns.
c	Codeword yield	Amount of Codewords that are decoded correctly.
d	Unused error correc-tion	Measures and grades the reading safety margin that error correction provides.
e	Modulation	The measure of the amplitude between two symbols specifies how reliable they can be distinguished.
f	Decodability	Describes, how far the true width of the moduls differ from the standard of the code types.
g	Defects	Indicates, to what extent irregularities or disturbances occur in individual symbol modules.

Structure of the quality result string of 2D code types borrowing from ISO/IEC TR 29158

The string with the format *a (bcdefghij)* indicates the determined quality which has the following meaning:

In-dex	Characteris-tic	Description
a	Overall qual-ity	The minimum or worst quality of all characteristics.
b	Cell contrast	Describes the difference between the highest and the lowest reflectance values of the gray value profile. Calculation of this characteristic differs from calcualation of "Contrast" of ISO/IEC 15415.
c	Cell modula-tion	The measure of the amplitude between two symbols specifies how reliable they can be distinguished. Calculation of this characteristic differs from calcualation of "Modulation" of ISO/IEC 15415.
d	Fixed pattern damage	Measures and grades any damage to the finder pattern and quiet zone in the symbol.
e	Decode	Binary characteristic of the readability. If it can be decoded, then it will be A otherwise F. In this implementation it will always be A.
f	Axial nonuni-formity	Measures and grades the spacing of the mapping centers and tests for uneven scaling of the symbol along the X or Y axis. Reasons for this behavior can be the print quality or the tilt of the camera.
g	Grid nonuni-formity	Measures and grades the largest vector deviation of the grid intersections, determined by the theoretical position prescribed by the reference decode algorithm and the actual measured result.
h	Unused error correction	Measures and grades the reading safety margin that error correction provides.
i	Reflectance margin	Comparable with the characteristic "Modulation"; also takes into account if a module could be decoded.

Tips on speeding up the tool "Read code"

If the tool does not reach the needed speed for your application, you can try the following steps to improve the speed:

1. **Reduce the code types via the wizard**
   The tool tries to find all chosen code types. More code types leads to a longer execution time of the tool. Therefor it is reasonable to search only for the code types available in the image. You can either show the wizard which codes are in the image or you can manually specify the types with the parameter "*Code type*".

2. **Reduce AOI**
   A further posibility is to reduce the Area of Interest (AOI). This will mean that codes are only search in the relevant area. It is sufficient that only parts of the code are located within the AOI.

3. **Choose unstructured background**
   Principally, the search algorithm assumes a code behind every structure. This initial suspicion is evaluated until the existence of a code can be rejected which will need additional processing time. If possible, the code should be placed above an unstructures background, like a uniform homogeneous, white surface.

4. **Fine-tune optimizations**
   If the complete range of code forms is known, like minimum and maximum code width, worst and best contrast between code and background, a training for speed optimizations make sense. The search algorithm can concentrate on the trained range and skip forms outside the range immediately. This improves the execution time. If the form limits are not known, you can train the tool with a sufficiently large number of samples. However, in this case you cannot eliminate that code forms outside the trained range are skipped.

Result tables

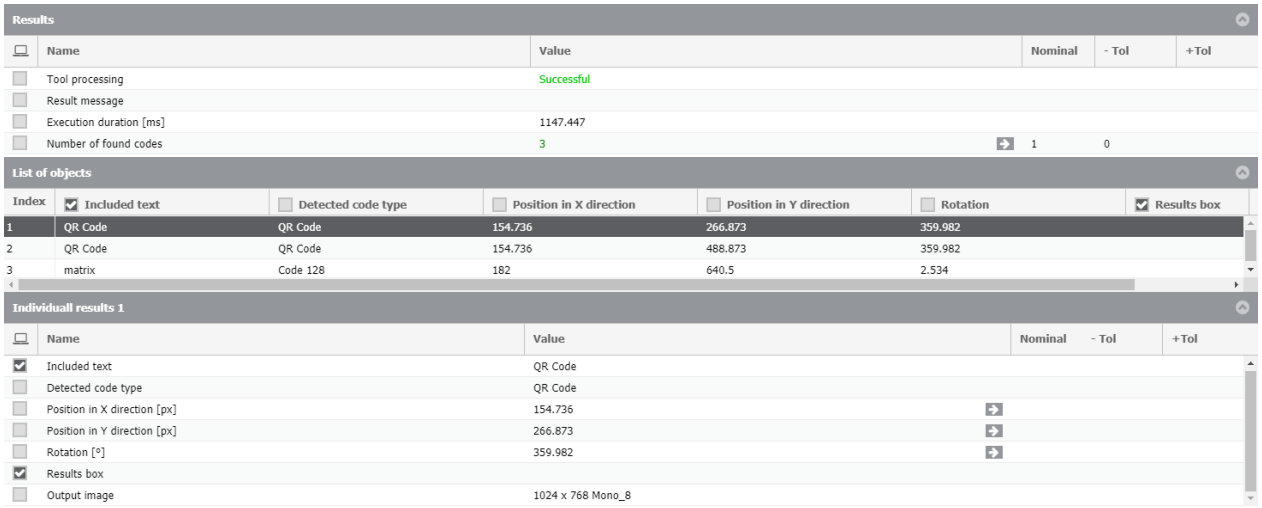


Table 1: Results

This table contains all general results belonging to the tool.

Result	Description	Data type	Default value	Tolerance adjustable
Tool processing	• Successful: no errors occurred. • Failed: the number of found codes is outside the specified tolerance range or a general error occurred.			✗
Results report	Text description of the cause of the error or warning	String		✗
Number of found codes	The number of found codes. The number is limited by the parameter *Expected number of codes.*	Int	0	✓
Execution duration [ms]	The time this single tool needs for execution	Double	0	✗

Table 2: List of all found objects

This table contains the list of all found objects. Is shown as soon as "Expected number of codes" is above 1.

Result	Description
Code_1	Represents a found code with its number. The properties of each code are displayed columnwise in this table. When a code is selected in this table, its properties are displayed in detail in the third table.

Table 3: Properties of the selected object

This table contains the properties of the result object that has been selected in table 2. Is shown as soon as "Expected number of codes" is above 1.

Result	Description	Data type	Default value	Tolerance adjustable
Included text	The text displayed by the code	String	Blank text	✗
Detected code type	The code type detected	String	Blank text	✗
Position in X direction	Horizontal distance from image origin at upper left corner to center of code. Position will be returned in world coordinate system if a calibration was performed in **"Set up camera"**.	Double	0	✓
Position in Y direction	Vertical distance from image origin at upper left corner to center of code. Position will be returned in world coordinate system if a calibration was performed in **"Set up camera"**.	Double	0	✓
Rotation	Rotation angle of the code. Reference orientation is described in table below. Rotation angle will be returned in world coordinate system if a calibration was performed in **"Set up camera"**.	Double	0	✓
Results box	The box in which the code was found, defined by its center position x and y, width, height and inclination. This box is shown in green when the results are within the tolerance values. The color of the box changes to red when the results are out of tolerance values.	Overlay	Blank space to begin with	✗
Module size	Size of the smalles bar code stripe (for 1D code types) or side length of a module (for 2D code types) in pixels	Double	0	✓
Mirrored code	Information whether a 2D code is mirrored or not. For 1D code types this value is always "false"	Bool	false	✗
Output Image	Rotated output image	Image	Blank image	✗
Code quality	The quality of the code	String	Blank text	✗

NOTE

For each expected code there is an individual result *Results box*, *Included Text*, *Detected code type*, *Position in X direction*, *Position in Y direction*, *Rotation*, *Output Image* and *Code quality.*

Result *Code quality* is only displayed if parameter *Assess quality* is set to "true".

Results *Module size* and *Mirrored code* are only displayed if parameter *View code details* is set to "true".

If *Tool Processing* is expected to be displayed as Failed if an unexpected number of codes was found, that can be reached by setting feasible tolerances. Some exemplary settings are listed in the table.

Description	Nominal	-Tol	+Tol
Tool Processing Failed if no code was found (default)	1	0	empty
Tool Processing Failed if not exactly one code was found	1	0	0
Tool Processing Failed if any code was found	0	empty	0
Tool Processing Successful independent of the number of found codes	empty	empty	empty

Reference Orientation 2D Codes

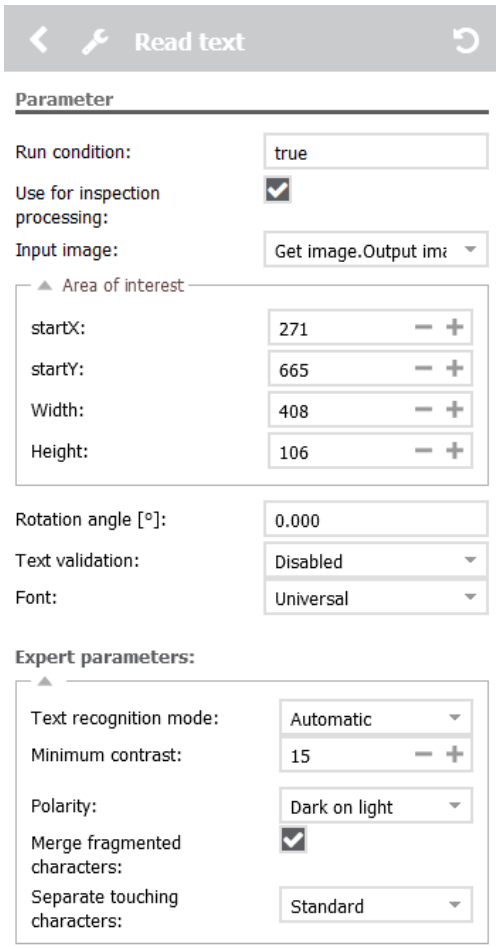
Code type	Aztec Code	Datamatrix ECC 200	Micro QR Code	PDF417	QR Code
Reference orientation					

5.3.6 Read text

Task

The tool reads a text in the image. The text can be aligned horizontally using the parameter rotation angle.

Parameters



NOTE

The tool **"Read text"** is able to read multiple line texts. Each line must have a consistent polarity (dark print on a light background and vice versa). Different lines are allowed to have different polarities. If the text validation is deactivated, every line of a multiple line text will be output. In this case, each line in the output text will be separated with a new line character.

Parameters	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, the outputs of this tool will be set to default values: • Arithmetic types: 0 • Strings: empty string • Images: empty images	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool
Input image	The image in which the text is being searched.	Image
Area of interest (x, y, width, height)	Area of the image in which the text is being searched.	[Double]
Rotation angle	Defines the angle in degrees by which the input image should be rotated.	Double
Text validation	• **Deactivated**: the text read is not validated. • **Regular expression**: an expected text or regular expression is used to verify the result. • **Lexicon**: a check is made to see whether one of the texts specified in a list has been found in the image. The most suitable text in the list is found.	Enum
Valid texts (mode: lexicon)	Specifies the list of valid texts to be searched in the image. The texts are separated by a semi-colon. **NOTE** In a multiple line text, the tool only outputs the line with the best match to the lexicon entry.	String
Text / expression (mode: regular expression)	The expected text. This can consist of a regular expression, to specify the format of the result text and avoid incorrect readings. **NOTE** In a multiple line text, the tool only outputs the line with the best match to the regular expression.	String
Font	The font used for interpreting the text. In most cases, the font "Universal" is suitable. Dialog **Trained font** gets visible as font "Custom" is selected.	Enum
Trained Font	Only visible if either text recognition mode or font is set to "Custom". Clicking on "..." allows to load a font trained in tool **Train font**.	Dialog

NOTE

Using font "Custom" in text recognition mode "Custom" provides both character segmentation (i.e. how to separate a character from background) as well as character classification (mapping of each segmented character to text). If other text recognition modes are selected, only character classification is done as defined in font "Custom".

Combining font "Custom" with text recognition modes "Automatic" and "Manual" is recommended if characters have to be read - which are not defined in default fonts - while, e.g. environmental light is fluctuating.

Setting both text recognition mode and font to "Custom" usually results in best processing time but is mostly bound to a small range of changes in illumination and working distance.

Expert Parameters

Text recognition mode	• **Automatic**: Automatic determination of the possible parameters. • **Manual**: The "Manual" mode allows detailed control of the reading process. • **Custom**: Allows to use a font created with tool "**Train font**". **NOTE** Depending on selected mode, available expert parameters change. The following sections describe of mode individual parameters.	Enum

Text recognition mode Automatic

Parameters	Description	Halcon Procedure	Data type
Minimum contrast	Minimal contrast between letters and their background.	min_contrast	Int
Polarity	• **Dark on Light**: The text to be segmented is darker than its background. • **Light on Dark**: The text to be segmented is lighter than its background.	polarity	Enum
Merge fragmented characters	Fragments such as an i-dot are added to the segmented characters.	add_fragments	Bool
Separate touching characters	Controls the handling of pairs or small groups of neighboring characters that are segmented as one single region. When selecting 'standard' or 'enhanced', such regions are detected and separated into two or more single characters • **Standard**: More inaccurate segmentation, but fast. • **Enhanced**: More accurate segmentation, but slower. • **Deactivated**: Disables the separation of touching characters.	separate_touching_chars	Enum

Text recognition mode Manual

Character features

Parameters	Description	Halcon Procedure	Data type
Character height	Height on an uppercase character in pixel.	manual_char_height	Int
Character width	Width of an uppercase character in pixel.	manual_char_width	Int
Stroke width	Stroke width of a character in pixel.	manual_stroke_width	Int
Polarity	• **Dark on Light**: The text to be segmented is darker than its background. • **Light on Dark**: The text to be segmented is lighter than its background. • **Don't care**: Both types of text are to be segmented.	manual_polarity	Enum
Dot print	Selects if text to be read is dot printed.	manual_is_dotprint	Bool
Imprint characters	Should be used when reading embossed text.	manual_is_imprinted	Bool

Character recognition

Parameters	Description	Halcon Procedure	Data type
Uppercase only	Should be activated if text only contains uppercase characters and digits.	manual_uppercase_only	Bool
Merge fragmented characters	Fragments such as an i-dot are added to the segmented characters.	manual_add_fragments	Bool
Return punctation characters	Punctuation marks as dots or comma are added to the segmented characters.	manual_return_punctuation	Bool
Return separation characters	Separators such as a minus or the equality sign should be added to the segmented characters.	manual_return_separators	Bool

NOTE

Mode Custom does not allow to modify any expert parameters. Instead all parameters used in tool "**Train font"** while creating a new font are shown. In this mode, expert parameters also provide a timestamp, indicating the date and time of creating the font, as well as a build number, which is increased by one every time the font is trained.

Regular expression

A regular expression is a search pattern against which a string is checked. As a result, the first partial string to correspond to the search pattern is returned.

The expression

```
"[A-Z]{3}[0-9]{2}[a-z]"
```

for example, accepts three upper case letters followed by 2 digits and one lower case letter.

Possible parameters are:

Parameters	Description
^	Denotes the start of the string
$	Denotes the end of the string, if necessary, including a new line as the last character.
.	Denotes each character, apart from a new line.
[...]	Denotes each symbol specified in the square brackets. If the first character is '^', the expression is negated. The character '-' can be used to enter value ranges, as in '[A-Z0-9]'. Other characters lose their specific meaning within square brackets, apart from '\'.
*	Permits 0 or more repetitions of the previous character/group.
+	Permits 1 or more repetitions.
?	Permits 0 or 1 repetition.
{n,m}	Permits n to m repetitions.
{n}	Permits exactly n repetitions.
	The foregoing repetition quantifiers normally seek the maximum possible match. With an additional ? the minimum match is sought, e.g. +?.
|	Separates alternative search expressions.
( )	Groups a sub-expression and saves a partial result.
(?:)	Pure grouping (e.g. for repetitions) with no partial result.
\	Escape character, permits the use of a symbol with special significance as a character. There are also some sequences with special significance (where the capitalized variant represents negation in each case): • \d,\D denotes numerals • \w,\W denotes numerals, letters and underscore • \b,\B denotes a word limit

Further examples

```
String: "abba"

Regular expression: "a*b*"            Result: abb
Regular expression: "b*a*"            Result: a
Regular expression: "b+a*"            Result: bba
Regular expression: ".a"              Result: ba
Regular expression: "[ab]*"           Result: abba

String: "img123"

Regular expression: "img(.*)"         Result: 123

String: "mydir/img001.bmp"

Regular expression: "img(.*)\\.(.*)"  Result: '001','bmp'
```

Check whether a **German vehicle registration plate** was recognized
(A **German vehicle registration plate** has following format: *German district* [= one or two or three letters] + underline character + one or two letters + underline character + one or two or three or four digits]; Example: ES_BF_1234)

```
"[A-Z]{1,3}[_][A-Z]{1,2}[_][0-9]{1,4}"

Correct strings would be: ES_BF_1234 | DUW_S_1
```

Check whether a **date** has been recognized
(date format can be one or two characters for the day and month with no plausibility check and also two or four characters for the year. A full-stop is permitted as a separator)

```
"^[0-9]+\.[0-9]+\.[0-9]+$"

Correct strings would be: 25.1.2015 | 1.4.1999 but also 45.23.20045
```

(date format can thus check one or two character entries for the day and month and also two or four character entries for the year between 1900 and 2099. A hyphen or full-stop are all permitted as separators):

```
"^(0?[1-9]|[12][0-9]|3[01])[-/.](0?[1-9]|1[012])[-/.](19|20)?[\d]{2}$"
 |---    Day entry    ---|      |-Month entry -|      |-Year entry -|
 |--- 0-31  or  01-31 ---|      |0-12 or  01-12|      |- 1900-2099 -|

Correct strings would be: 25.1.2015 | 1.4.1999 3-4-2105 | 5/3/1900 but
also 31.02.2015
```

Check a **production code**
(code has five digits, followed by a forward slash, followed by two digits)

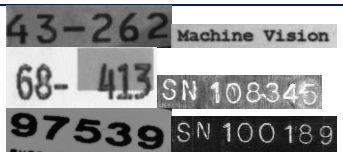
```
"[0-9]{5}[\/][0-9]{2}"

Correct strings would be: 12345/12 but also 1234567/12345
```

Supported fonts

The following fonts can be read:

Fonts	Included characters	Supported special characters	Examples
Document_A-Z+	Upper case letters and special characters	- = + < > . # $ % & ( ) @ *	BALLUFF BALLUFF BALLUFF
Document_0-9A-Z	Numerals and upper case letters		
Document_0-9	Numerals		
Document	Upper and lower case letters, numerals, special characters		
HandWritten_0-9	Numerals	None	0123456789 0123456789 0123456789
Industrial_A-Z+	Upper case letters and special characters	- / + . $ % *	320 38 2 13 544/2 5377479 43-262 Machine Vision 68- 413 SN 108345 97539 SN 100189
Industrial_0-9A-Z	Numerals and upper case letters		
Industrial_0-9	Numerals		
Industrial_0-9+	Numerals and special characters		
Industrial	Upper and lower case letters, numerals, special characters		
OCRA_A-Z	Upper case letters	- ? ! / \{} = + < > . # $ % & ( ) @ *	0123456789 ABCDEFGHIJKLM NOPQRSTUVWXYZ abcdefghijklm nopqrtsuvwxyz -?!/\=+<>.#$%&()@*
OCRA_0-9A-Z	Numerals and upper case letters		
OCRA_0-9	Numerals		
OCRA	Upper and lower case letters, numerals, special characters		
OCRB_A-Z+	Upper case letters and special characters	- ? ! / \{} = + < > . # $ % & ( ) @ *	0123456789 ABCDEFGHIJKLM NOPQRSTUVWXYZ abcdefghijklm nopqrtsuvwxyz -?!/\=+<>.#$%&()@*
OCRB_0-9A-Z	Numerals and upper case letters		
OCRB_0-9	Numerals		
OCRB	Upper and lower case letters, numerals, special characters		
OCRB_passport	Upper and lower case letters, numerals, special characters		
Pharma_0-9A-Z	Numerals and upper case letters	- / . ( ) :	N3 20 50 N2 11/2005 E13004 03/2007 426A06 02/2007 345B11
Pharma_0-9	Numerals		
Pharma_0-9+	Numerals and special characters		
Pharma	Upper and lower case letters, numerals, special characters		
SEMI	Upper and lower case letters, numerals, special characters	- .	ABCDEFGHIJKLMNO PQRSTUVWXYZ- 0123456789.
DotPrint	Upper and lower case letters, numerals, special characters	* - . / :	01.09.06 074104 07123 (A)20:20 12040A 01.04.05 5K2/031I 040704
DotPrint_0-9	Numerals		
DotPrint_0-9+	Numerals and special characters		
DotPrint_A-Z+	Upper case letters and special characters		
DotPrint_0-9A-Z	Numerals and upper case letters		
Universal	Upper and lower case letters, numerals, special characters	- / = + : < > . # $ % & ( ) @ *	320 38 2 13 544/2 5377479

Universal_0-9	Numerals		
Universal_0-9+	Numerals and special characters		
Universal_A-Z+	Upper case letters and special characters		
Universal_0-9A-Z	Upper case letters and numerals		
Universal_0-9A-Z+	Upper case letters, numerals, special characters		
Custom	Arbitrary (see note)		

NOTE

Font "Custom" can only be used if a font, which was trained in tool "**Train font**", has already been loaded using dialog **Trained font**.

NOTE

Space characters are generally ignored and are not read. They should not be used in a regular expression.

In order to avoid misreadings a text will only be read if it consists of at least two characters.

In order to get reliable results the font height should not be too small:

- At least 30 pixels for reading text (OCR)
- At least 10 pixels for validating text (OCV)

If there are any difficulties reading dot print fonts, it is advised to use UniversalDotPrint. This font is able to recognize dot print data which vary to the dot print standard but will take more time.

Results

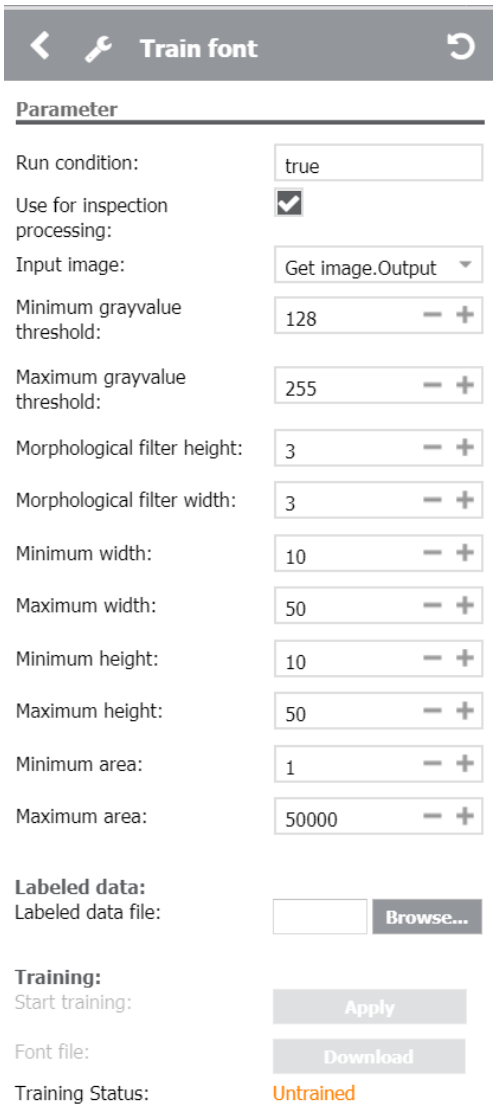
Result	Description	Data type	Default value	Tolerance adjustable
Tool processing	• Successful: no errors occurred. • Failed: ○ No text found in the area of analysis. ○ Text found. However, it deviates excessively from the expected text ○ General error			✗
Results report	Text description of the cause of the error or warning	String		✗
Execution time [ms]	The time this single tool needed for execution.	Double	0	✗
Output text	The text that was read in the image.	String		✗
Degree of match [%]	The degree to which the found text matches the expected text.	Double	60	✓
Output image	The output image is the input image rotated by the value of Rotation angle.	Image		
Area of interest	Image section has been verified. If the calculated focus value is within the tolerance range then this area is displayed green, else in red	Overlay	Identical to the defined area on interest	✗
Width/Height of segmentend characters	Information about minium, maximum and median value of currently segmented characters.	String		✗

5.3.7 Font training

Task

Train a custom font using this tool. Useful to read fonts that are not directly recognizable by the tool "**Read text**".

Parameter



Parameter	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, the outputs of this tool will be set to default values: • Arithmetic types: 0 • Strings: empty string • Images: empty images	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool
Input image	The image in which the text is being searched.	Image
Minimum grayvalue threshold	Lower gray value of the pixels for segmentation.	uint16
Maximum grayvalue threshold	Upper gray value of the pixels for segmentation.	uint16
Morphological filter height	Filter height for blob pre-evaluation. • for values <= -0.5 a closing filter is applied (connects close blobs) • for values >= 0.5 an opening filter is applied (separates blobs with a thin connection) • between -0.5 and 0.5 no filter is applied.	uint16
Morphological filter width	Filter width for blob pre-evaluation. • for values <= -0.5 a closing filter is applied (connects close blobs) • for values >= 0.5 an opening filter is applied (separates blobs with a thin connection) • between -0.5 and 0.5 no filter is applied.	uint16
Minimum width	Sets the minimal width of a blob in pixels. Smaller blobs are discarded.	uint16
Maximum width	Sets the maximum width of a blob in pixels. Wider blobs are discarded.	uint16
Minimum height	Sets the minimal height of a blob in pixels. Smaller blobs are discarded.	uint16
Maximum height	Sets the maximum height of a blob in pixels. Larger blobs are discarded.	uint16
Minimum area	Sets the minimum size of a blob (pixel count). Smaller blobs are discarded .	uint16
Maximum area	Sets the maximum size of a blob (pixel count). Larger blobs are discarded.	uint16
Labeled data file	Upload the file containing the characters and their respective labels. For creating labeled data, *MVTec's Deep Learning Tool* or the *Universal Data Tool* can be used. The required file format of the labeled data file is either MVTec's dictionary format (.HDICT) or the Universal Data Tool JSON format (.UDT) as described in https://github.com/UniversalDataTool/udt-format. **NOTE** A detailed step-by-step guide of how MVTec's Deep Learning Tool can be used for labeled data creation is accessed by the integrated help system of the BVS Cockpit client.	File
Start training	Start training the font based on the labelled file. Training can be started only if a valid label file has been uploaded.	
Font file	Download the trained font. Once the training is completed, the training font is available for download.	
Training Status	Shows the training status. • Trained • Untrained • Modified	

Sequence of the training

1. In the tool "**Set up camera**" select "**File Device**" as active camera.
2. Import all images used for font labeling via "**Import images...**" to the image batch.
3. Select the images via "**Select images...**".
4. Add the tool "**Train font**" and make sure that all the necessary images are present by clicking "**Run once**".
5. Perform segmentation by selecting a suitable upper and lower gray values.
   This roughly segments the font region from the rest of the image.
6. Fine tune the segmented font regions (blobs) by performing morphological operation.
7. Filter the segmented blobs based on the following parameters
   a. Height
   b. Width
   c. Area
      After filtering, it is expected that the font regions representing the font to be trained is almost the same as the labelled font.
8. Upload the file via "**Labeled data file**" containing the font labels.
9. Initiate training by clicking "**Apply**" next to "**Start training**".
10. Once the training has successfully completed, the trained font can be downloaded.

Results

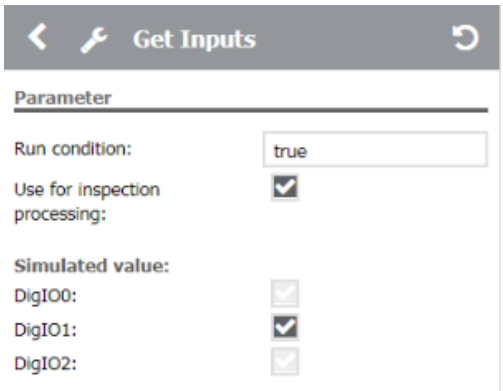
Result	Description	Data type	Default value	Tolerance adjustable
Tool processing	• Successful: no errors occurred. • Failed: ○ No text found in the area of analysis. ○ Text found. However, it deviates excessively from the expected text ○ General error			✗
Results report	Text description of the cause of the error or warning.	String		✗
Execution time [ms]	The time this single tool needed for execution.	Double	0	✗
Font region	The region obtained after blob segmentation.	Region		✗

5.4 Interface tools

5.4.1 Get inputs

Task

Used to read digital camera inputs. This provides a way to receive signals from other devices. The inputs are read in the moment the tool is being processed.



Parameters	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, the outputs of this tool will be set to default values: • Arithmetic types: 0 • Strings: empty string • Images: empty images	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool
IO 0	Simulated value for input 0.	Bool
IO 1	Simulated value for input 1.	Bool
...		

NOTE

The input parameters for the simulated values are only accessible in simulation mode.

Results

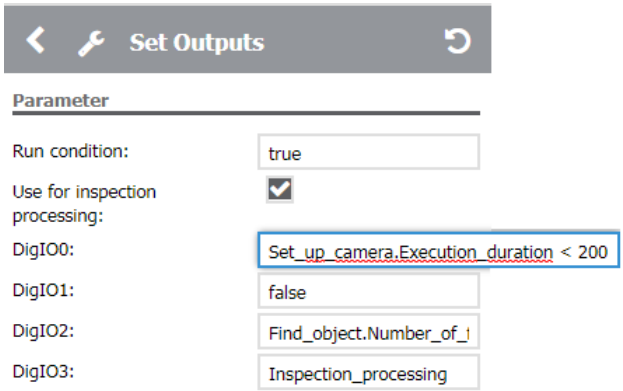
Result	Description	Data type	Default value	Tolerance adjustable
Tool processing	• Successful: No errors occurred. • Failed: A general error occurred			✗
Results report	Text description of the cause of the error	String		✗
Execution time [ms]	The time this single tool needs for execution	Double	0	✗
IO 0	The value entered on the input 0.	Bool		✗
IO 1	The value entered on the input 1.	Bool		✗
...				

5.4.2 Set outputs

Task

Used to manually switch digital camera outputs or link with results of other tools. This enables the result of an inspection program to further serve other connected devices.

Parameters



Parameters	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, the digital output remains at the value of the last tool execution.	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool
IO 0	The value entered on the output 0. You can use constant values (*true* to activate the output, *false* to deactivate it) or links to results of other tools.	Bool
IO 1	The value entered on the output 1. You can use constant values (*true* to activate the output, *false* to deactivate it) or links to results of other tools.	Bool
...		

NOTE

In order to invert a digital output use the keyword *NOT*. The expression *NOT Find_object.Tool_processing* sets the output only if the tool **"Find object"** has an error.

NOTE

Only compatible outputs for the selected camera are shown. If an output is already used by other functions, such as flash release or trigger, this is not shown and cannot be changed.

Outputs are not working as long as the camera simulation is active.

NOTE

If no value is specified for an output, its state remains unchanged.

Results

Result	Description	Data type	Default value	Tolerance adjustable
Tool processing	• Successful: No errors occurred. • Failed: A general error occurred			✗
Results report	Text description of the cause of the error	String		✗
Execution time [ms]	The time this single tool needs for execution	Double	0	✗
IO 0	Value of output 0. The value of the complex input expression.	Bool		✗
IO 1	Value of output 1. The value of the complex input expression.	Bool		✗
...				

5.4.3 Receive data

Task

Used to receive input data via data interface. New input data can be added by selecting the data type.

Compatible data types are:

Data type	Length	Length adjustable
string	20 characters	✓
int16	2 Byte	✗
int32	4 Byte	✗
float32	4 Byte	✗
bool	1 Byte	✗

How data packages must be formatted to be recieved by this tool, is described in chapter "Connection to the customer control system".

NOTE

The tool does not wait until data is being received. You can ensure, that received data is actually available when the tool is executed, by placing this tool after the tool "Get image" within the inspection program, but send the data package before triggering. This is only reliable, if data package and trigger are sent via the same communication channel; a mix of data packages via UDP/TCP/fieldbus with a trigger via digital input is therefore not recommended.

Parameters

Parameters	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, the outputs of this tool will be set to default values: • Arithmetic types: 0 • Strings: empty string • Images: empty images	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool
Position	Position in bytes where the input is expected in the received data packet.	Uint
Type	The type of an input	String
Size	The size, in bytes, or the number of characters in strings	Uint
Name	The name of an input. The input data can be accessed from other tools with this name. The name must contain characters limited to A-Z, a-z ,numbers, underscore and space . Apart from underscore and space characters, the name cannot contain any other special characters.	String
Simulated value	Value of the specified type which is used in simulation mode.	Appropriated to the input data type
No multi read	When this option is checked, received input data is read only once. The tool enters an error state on further reading attempts and its outputs are set to default values. In config mode, only a warning is issued.	Bool

NOTE

The input parameter **Simulated value** is only accessible in simulation mode.

NOTE

Before creating a new row, the row's position can be set by using the two arrow buttons right to the type dropdown.

Results

Result	Description	Data type	Default value	Tolerance adjustable
Tool processing	• Successful: No errors occurred. • Failed: A general error occurred			✗
Results report	Text description of the cause of the error	String		✗
Execution time [ms]	The time this single tool needs for execution	Double	0	✗

For each entry in parameters there is an entry in this table.

NOTE

If the specified number of characters in strings does not match the actual length of the received string, *Tool Processing* is displayed as Failed. In simulation mode, the simulated value is adapted to the specified number of characters, however.

5.4.4 Send results

Task

Used to send inspection program results. Global settings specify whether the result is sent to the PLC via the fieldbus or via the Ethernet interface (TCP/IP or UDP). New results are added by selecting the data type.

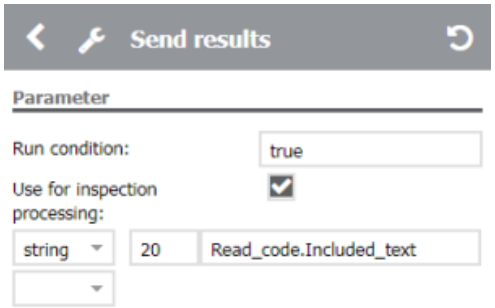
Compatible data types are:

Data type	Length	Length adjustable
String		✓
int16	2 Byte	✗
int32	4 Byte	✗
float32	4 Byte	✗
bool	1 Byte	✗

NOTE

After every change of data to be sent, a result with the current configuration will be generated and sent, whether or not loop mode is active.

Parameters



Parameters	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, no result will be sent to the PLC.	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool
Position	Position of the result in bytes in the sent data packet.	Uint
Type	The type of result	String
Size	The size, in bytes, or the number of characters in strings	Uint
Name	The name of result. The name must contain characters limited to A-Z, a-z ,numbers, underscore and space . Apart from underscore and space characters, the name cannot contain any other special characters.	String
Content	The content to be send. The content may consist of a constant expression, a link to a result of another tool or a formula. If the content is an expression, it must contain only standard ASCII-conformant characters.	Depends on the chosen type

NOTE

It is possible to assign a floating-point number to an integer. This causes the floating-point number to be rounded.

NOTE

Before creating a new row, the row's position can be set by using the two arrow buttons right to the type dropdown.

Results

Result	Description	Data type	Default value	Tolerance adjustable
Tool processing	• Successful: No errors occurred. • Failed: A general error occurred			✗
Results report	Text description of the cause of the error	String		✗
Execution time [ms]	The time this single tool needs for execution	Double	0	✗
Results	All the added result properties and their values are displayed			✗

Interpretation of the sent data

The **"Send results"** tool sends the data via the communication channel defined in the system settings. In order to interpret the data properly its structure, as it is described in chapter "Connection to the customer control system"., has to be taken into account.

NOTE

More information about its configuration can be found in "System menu" chapter of the manual.

NOTE

The "Use cases" chapter contains examples for the usage of the **"Send results"** tool.

NOTE

Some barcode scanner devices send *End of Line* at the end of the message to finalize the message. In order to adopt this behavior you have to configure a parameter **int16** at the end of the message with a constant value of **3338** (0x0D0A, corresponds to CR/LF).

5.4.5 Calibrate robot

Task

This tool is used to calibrate a robot. Using such a calibration allows you to convert result coordinates of tools such as "**Find object**" or "**Check blobs**" by a robot controller or an intermediate application - subsequently called "remote station" - into motion commands for controlling a robot.

NOTE

This tool provides a wizard guided calibration. No further functionality is provided beside this, hence there is no need to keep this tool in any inspection.

Requirements

To use this wizard, following requirements need to be met:

- The robot's interactive calibration process allows to set origin and rotational offset of an 2D coordinate system.
- The camera is either mounted fix or is attached to the robot and is taken into the very same 6D pose as the calibration was conducted.
- All objects to be measured / picked have to be in a 2D plane. The robot TCP pose while picking objects or performing any other tasks on this plane will stay constant.
- The dimension of the calibration plate suits the working environment for the desired robot application.
- The calibration plate can be placed either flat and sound on the working plane itself or with an constant distance to it.
- Complete calibration plate as well as all possible object positions are fully viewable by the camera.
- A particular developed software module residing on either PLC or robot control itself ("remote station") is available, which parses all data received from the camera system into a format suited for the robot.

NOTE

To allow a proper mapping of the object to be found and robot TCP, it is crucial not to change the plate's position during camera calibration in tool "**Set up camera**" and this robot calibration. If the plate's position is changed, you have to repeat the camera calibration in tool "**Set up camera**".

DANGER

Uncontrolled system movement

This tool exclusively provides a supporting functionality. It is in your personal responsibility to ensure a hazard-free and safe employment of the robot and additional periphery equipment for both man and machine.

→ Persons must keep away from the system’s hazardous zones.

→ Startup must be performed only by trained technical personnel.

Wizard

Page	Explanatory text	Button	Comments
1 - Introduction	In the next steps ... interactively.		
2 - Choose calibration plate	Select the calibration plate also used in "**Set up camera**". Ensure calibration plate has not been moved. If possible, secure the calibration plate using dowel pins or provide a precise milled cavity allowing for a repeatable placement of the calibration plate.	**320 mm** **640 mm** **1000 mm**	Choose the same type of calibration plate that you selected in "**Set up camera**". If you decide to choose a different type, you have to repeat the camera calibration in "**Set up camera**". Beside the known calibration patterns for the camera calibration, the calibration plate includes the needed targets for robot calibration. **NOTE** For fault-free function of this wizard it is required that remote station is compatible with the used data format. Therefore it is advised to consider your robot manufacturers documentation. Usually, you can also find information about which calibration targets can be used and which order they need to be visited. If there are no requierements on any particular order, it is advised to select targets most distant from each other to maintain a small calibration error (Use targets B, D, F and G).
3 - Start calibration process	Start calibration on remote station. Continue wizard, as soon as calibration is ready at remote station.		After starting the calibration software on the remote station, it will expect to receive multiple coordinates in the camera's coordinate system. The remote station acquires the robot's pose - which you aligned to the corresponding target - and maps it to the received coordinates. As soon as enough calibration targets are visited - usually at least 3 - the remote station can compute the transformation parameters. Using this particular transformation parameters, the remote station is able to translate results of tools like "**Find object**" or "**Check blobs**" into motion control commands suitable for the used robot.
4 - Approach calibration targets	Visit all required targets with robot TCP. Direct the robot TCP to the center of the selected calibration target. Select each visited target via this dropdown menu und confirm by clicking on "**Send coordinates**". Repeat these steps for all other necessary calibration targets. Click on "**Complete calibration**" after all target points have been completed.	**Target A** **....** **Target H** **Send coordinates** **Complete calibration**	Select calibration targets suitable for the remote station. **NOTE** Usually, the complete pose of the robo TCP is acquired by a remote station. Hence, it is advised to orientate the robot TCP into the proper alignment - typically, perpendicular to the calibration plate. **NOTE** Changing the calibration plate's position during calibration results in restarting the robot calibration and as well as re-run the camera calibration.

5 - Verify calibration	Please verify if calibration has been successfully finished on remote station.		From the camera's perspective, calibration is completed. Complete the calibration process on remote station as required. Please verify the robot's proper function in consideration of all necessary safety precautions.

Used data format

The wizard of tool "**Calibrate robot**" transmits the selected target position using the communication port as defined in system settings using a result container. For certain protocols, it is necessary to first establish connection by sending a "*Connect*" message before communicating the selected target position to the remote station. Hence, to ensure proper functionality, remote station is required to precisely parse all data as defined in chapter "Connection to the customer control system".

NOTE

Information on configuring communication port can be found in hardware manual chapter "Connection to the Customer Control System". To review any issues on compatibility also consult the robot's documentation on calibration.

Usage of a calibration

After completing the calibration, you can use coordinate results of tools as "**Find object"** or "**Find blob**" via "**Send result**" to remote station:

1. Enter "**Send results**" tool's settings page.
2. Add two entries to result container.
3. Select the correct data type appropriate for your remote station.
4. Connect results container's both empty fields to property "*Find_object_Displacement_in_X*" and "*Find_object_Displacement_in_Y*" of tool "**Find object**".

On remote site - details may vary depending on used model - can be brought into service:

1. At robot controller, activate simulation mode.
2. Place an object on working plane and execute the inspection.
3. Verifiy plausibility of robot control's calculated poses
4. If poses seem to be valid, deactivate simulation mode on robot controller.

NOTE

Further information on using tools "**Find object**", "**Find blob**" and "**Send results**" can be found on corresponding chapters in the manual.

Expert knowledge

The camera system can be used, to detect objects in the image, Then the object's coordinates are sent to a robot system to implement Pick & Place applications.

Example

The robot calibration process will look like as follows.

Configuration

The image acquisition will always take place at the same position (camera mounted on a robot looking to a work item, or a stationary camera above a robot looking to a work item).

Required components

- Camera system (**SMART***CAMERA*, Balluff *Camera* at a Balluff SmartVision Controller)
- Robot (scara, six-axis, portal, ...)
- Calibratable robot control with interface to the camera

Process

1. Initial operation of the system
    - Create inspection with following tools
        - "**Set up camera**",
        - "**Get image**",
        - "**Find object**",
        - "**Send results**".
    - Check image acquisition and adjust it.
    - Check robot communication via "**Send results**".
2. Camera calibration
    - Select and print suitable calibration plate (A3, A1, A0) with robot calibration patterns which you can find in the wizard of the tool "**Set up camera**".
    - Fix the calibration plane flat on the working plane, where the work item will be placed.
    - If the camera is mounted on the robot's arm, move the robot to the test position via the robot control.
    - Enter the wizard of the tool "**Set up camera**", chose option "with calibration" and execute it.
3. Robot calibration
    - According to the used robot control you can use different targets on the calibration plate.
    - If there are no limits by the robot control, you should use the targets B, D, F, H at the borders.
    - Move the robot TCP via the robot control to the first used target.

      NOTE

      Please be aware, that the robot TCP is oriented the right way to the working plane.
    - Enter the coordinates of the current target into the calibration assistant of the robot control.
    - Repeat the last two steps for the needed targets and finalize the calibration of the robot control.
4. Detect object
    - Execute the wizard of the tool "**Find object**" and train the work items.

NOTE

The work item has to be in the working plane.

5. Communication with the robot control
   - Go to the settings of the tool "**Send results**".
   - Add to entries to the result package.
   - Chose the right data type for the used robot control.
   - Connect both empty fields of the result package with the outputs of the tool "**Find object**" with "*Find_object_Displacement_in_X*" and "*Find_object_Displacement_in_Y*".

NOTE

Please make sure that the order of the data and the data type meet the requirements of the robot control.

6. Startup
   - Set the robot control to simulation.
   - Place an item on to the working plane and execute the inspection program.
   - Check if the robot control calculates logical poses.
   - Deactivate the robot's simulation mode if the calculated poses make sense.

5.5 Extended Tools

5.5.1 HALCON Script

NOTE

Only included in the HDevelop SmartCamera version.

Each product firmware was realized with following HALCON versions:

Product	HALCON version	Required HDevelop version for HALCON programs
BVS SC **SMART***CAMERA*	19.11.0.0	≥ 19.11.0
BAE PD SmartVision Controller	19.11.0.0	≥ 19.11.0
BAI BVS-CA BVS Cockpit Windows	19.11.0.1	≥ 19.11.0

Supported functions groups of HALCON version 19.11.0:

Function group	Description
Foundation	Arithmetic, bit, edges, noise, smoothing, texture and further filters, FFT, optical flow, classification, segmentation, morphology, blob analysis, color image processing, serial and socket communication, acquiring, reading and writing images, visualization, and window handling. Automatic Operator Parallelization (AOP). GPU acceleration. Processing of images larger than 32k x 32k.
Calibration	3D camera calibration for line and area scan cameras handling even complex lens distortions. Multi-view camera calibration mixed camera setups for 3D reconstruction. Hand-eye calibration for robot vision. Gray-value calibration for linearizing a camera response.
1D Metrology	Subpixel precise projection of gray values to perform 1D measurements along rectangular or circular structures.
2D Metrology	Subpixel precise filters and thresholding operations for obtaining the entire 2D contour of objects, easy measurement of standard shapes.
3D Metrology	Stereo vision and sheet of light for subpixel precise 3D measurements, localization of 3D objects in 3D by using a single camera, surface-based 3D matching, multi-view stereo for robust 3D reconstruction, fitting of 3D primitives, 3D surface comparison, 3D object processing, generating 3D models from views or file in dxf format.
Bar Code	Robustes und schnelles Lesen von Barcodes, automatische Ortung und Dekodierung der am häufigsten verwendeten Barcodes wie GS1 DataBar und Überprüfung der Druckqualität.
Data Code	Robustes und schnelles Lesen von ECC200, QR, Micro-QR, Aztec und PDF417 Codes, automatische Ortung und Dekodierung der Data-Codes einschließlich der Überprüfung der Druckqualität.
OCR / OCV	Train and classify characters reliably for performing efficient OCR/OCV. Industrial-proven pre-trained fonts included. Syntactic and lexicon-based autocorrection. Automatic segmentation and reading of text.
Matching	Subpixel precise matching based on shape, gray-values, or descriptors of the object, automatic detection of the variation of the appearance of objects, component-based matching, local deformable matching, image stitching (mosaicking), generating models from dxf format, sample-based identification, easy-to-use texture inspection.

Task

Executes HALCON programs. I.e., new and existing HALCON programs can be integrated. For this, the program sequence has to be moved to the procedure called "*HalconRun()*". This offers the possibility to exchange images and data. You only have to adapt the input and output to the appropriate postfixes, which is described as follows.

Parameters

Parameter	Description	Data type
Run condition	Condition that has to be met in order for the tool to run. You can use constant values (*true* to run the tool, *false* to skip it) or links to results of other tools. If the tool did not run, the outputs of this tool will be set to default values: • Arithmetic types: 0 • Strings: empty string • Images: empty images	Bool
Use for inspection processing	Determines whether the tool result has an effect on the inspection result.	Bool
Upload	Input field to select and upload HALCON programs and data. The program files must have the ending "**.hdev**" and the file size limit is **1 MB**.	

The tool executes the program's "*main()*" during initialization (when a script is loaded) and a local procedure called "*HalconRun()*" while executing the tool. For this reason, the execution of "*main()*" must not fail. Furthermore,

- parameters must be assigned according to the naming scheme (see expert knowledge),
- it is possible to store and use other local procedures in the HALCON program, and
- it is possible to use HALCON encryption for the program and for the procedures.

Results

Result	Description	Data type	Default value	Tolerance adjustable
Tool processing	• Successful: No errors occurred. • Failed: An error occurred during the execution of the program.			
Results report	Text description of the cause of the error	String		
Execution time [ms]	The time this single tool needs for execution.	Double	0	

Expert knowledge

Summary of the HALCON Parameters

Types	Iconic / Control	Postfix	Input	Output (Result)	Default Value[1]	Default Color[1]	Color	Transformation	Transformation Link
Image	I	*_Img*	✓	✓				✓	
AOI	I	*_Aoi*	✓		✓	✓	✓	✓	✓
Region	I	*_Region*	✓	✓		✓	✓	✓	✓
Contour (=XLD[2])	I	*_Contour*		✓		✓	✓	✓	✓
Integer	C	*_Int*	✓	✓	✓				
MinMax Integer	C	*_MInt*	✓		✓				
Real	C	*_Real*	✓	✓	✓				
MinMax Real	C	*_MReal*	✓		✓				
String	C	*_String*	✓	✓	✓				
Boolean	C	*_Bool*	✓	✓	✓				
Checkbox	C	*_MBool*	✓		✓				

Enum	C	*_Enum*	✓		✓				
Rectangle	C	*_Rect*	✓	✓	✓	✓	✓	✓	✓
Virtual Image	C	*_VImg*		✓					
Non Transformed Image	I	*_NImg*		✓					
Object[3]	I	*_Object*	✓	✓	✓				
Tuple[3]	C	*_Tuple*	✓	✓	✓				
Communication[3]	C	*_Com*	✓	✓					
Serialized Object[4,3]	C	*_Serialized*	✓	✓					
Mask	I	*Mask*	✓			✓	✓	✓	✓
ArrayCount	C	*ArrayCount*	✓						
ToolResult	C	*ToolResult*		✓					
RunMode[3]	C	*RunMode*	✓						
System Information[3]	C	*SystemInfo*	✓						

[1] Default values can be set in the "*main()*" of the HALCON program and are set only once when a new script is loaded in config mode.

[2] XLD stands for "eXtended Line Description" and comprises all Contour and Polygon based data.

[3] These types are displayed neither as text nor as graphical element.

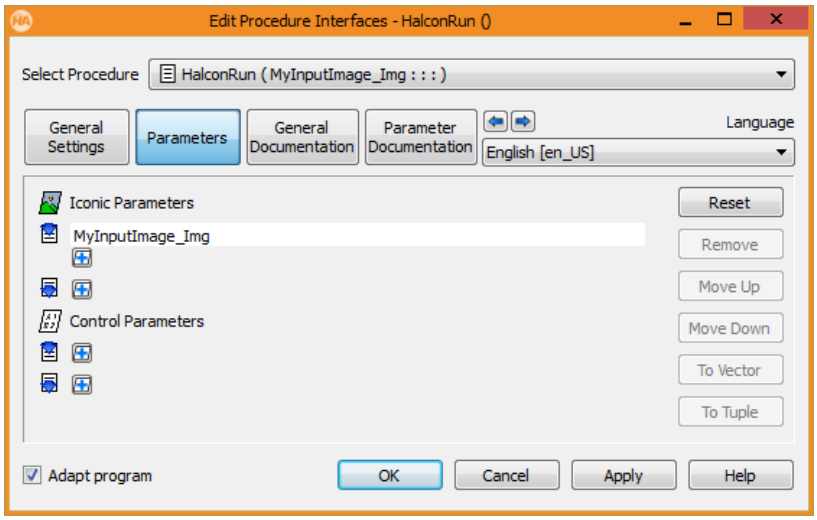
[4] Serialized objects include exclusively tuples and iconic objects, serialized models can't be used.

Rules for HALCON parameters

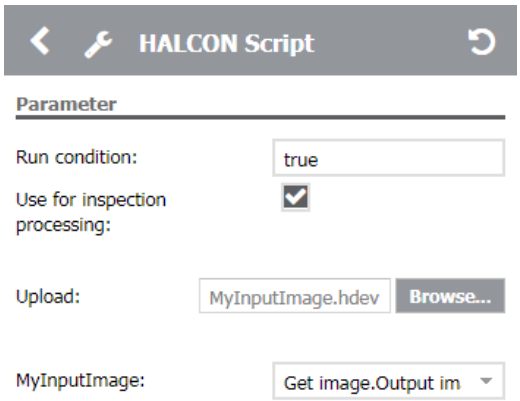
In order to work with the parameters properly, you have to follow some rules:

- Parameters in Halcon must be named like *<name>_<postfix>*. The <name> part is used as parameter/result name in the tool.
  Exceptions of this scheme are described in the following. Some parameters have fixed names (e.g. *Mask*).
- Input or output (parameter or result) is defined in HALCON.
- Adding an _ at the end of the name inverts the direction of a variable (see inverted parameters)
- Parameter count per procedure is limited by HALCON.
  - Control: 20 (In/Out each)
  - Iconic: 9 (In/Out each)
- Parameters with the same name (normal / inverted) are the same variable
- Input strings should not be empty.

To define an input image, for example, add an iconic input parameter with the name "**MyInputImage_Img**" in HDevelop:



This will generate an input image parameter:



While loading a HALCON program,

- parameters are added in the order they are defined inside the parameter settings of the "*HalconRun()*" procedure. This is followed by parameters from further procedures ("*HalconInit()*", "*HalconFinalize()*", Callbacks).
- When loading another script, parameters that match are reused.
  Reused parameters stay at their position, new parameters can be added before, after or between reused parameters, but reused parameters can't switch positions.
- Reused parameters keep their value while a new program is loaded, except there are overwritten in the "*main()*" by a default value.

Working with HALCON parameters

Integer, Real, String, Boolean

Variables	*Postfix*	*Example*

		`* default value in main()`
		`MyIntValue_Int := 123`
		
		`* set value in HalconRun()`
		`MyIntValue_Int := 456`

• Booleans contain 1 for true and 0 for false		

MinMax Integer and Real

Variables	*Postfix*	*Example*

	_MInt_MinMax
	_MReal_MinMax
		`* set minimum and maximum in main()`
		`Test_MInt_MinMax := [-10, 20]`
		`* default value in main()`
		`Test_MInt := 3`
		
		`* set value in HalconRun()`
		`Test_MInt := 5`

• In the example, *Test_MInt* must be available.		

AOI

Variables	*Postfix*	*Example*

	_Values	`* default values in main()`
		`MyAOI_Aoi_Values := [100, 200, 300, 400.567]`
		`* set default color in main()`
		`MyAOI_Aoi_Color := [255, 255, 0, 255]`
		
		`* usage in HalconRun()`
		`reduce_domain(Image, MyAOI_Aoi, ImageReduced)`

• *AOIs* are passed to "*HalconRun()*" as regions. • Default values are set as *Tuple* [x, y, width, height] in "*main()*". x, y describes the *AOI*'s left upper corner.		

Rectangle

Variables	*Post-fix*	*Example*

Tuple [x,y,width,height,angle]	*_Values*	`* default values in main()`
		`MyRectangle_Rect_Values := [100, 200, 300, 400, 55.55]`
		`* set default Color in main()`
		`MyRectangle_Rect_Color := [255, 255, 0, 255]`
		
		`* set value in HalconRun()`
		`MyRectangle_Rect := [101, 202, 303, 404, 66.66]`

• *Rectangles* can be used as input, output and inverted. • The pecification of an angle is optional, so *Rectangles* can be set with either four or five values. • x, y, w, h are defined in the camera's coordinate system. x, y describes the *AOI*'s left upper corner. • The angle is specified in degree, rotation is around (x,y). • *Rectangles* can have *Transformation* and *Color* as additional parameters. • A *Color* can be set as default parameter.		

Enum

Variables	*Postfix*	*Example*

Tuple [index,string]	*_Enum*
	_Enum_Values
		`* define enum values in main()`
		`color_Enum_Values := ['red', 'green', 'blue']`
		`* set default value in main()`
		`color_Enum := 'green'`
		
		`* usage in HalconRun()`
		`color_Enum[1]`

• *Enum* parameters can only be input. • For invalid values a -1 is passed as index.		

Tuple

Variables	*Postfix*	*Example*
	_Tuple	`test_Tuple := [123, 1.23, 'this is a string inside a tuple']`
• *Tuple* can be used as input, output and inverted. • *Tuple* are not displayed in the HALCON tool. • *Tuple* can hold multiple values and handles (serialization handles, model handles, ...). • The value of a handle within a *Tuple* is not stored in an inspection program and is not copied when the HALCON tool is duplicated. It is the user's responsibility to store the model or object that is referenced by the handle, to generate a new handle during load of the model or object, and to assign this new handle to the *Tuple*.		

Object

Variables	*Postfix*	*Example*
	_Object	`gen_circle (test_Object, 200, 200, 100.5)`
• *Objects* can be used as input, output and inverted. • *Objects* are not displayed in the HALCON tool. • *Objects* can hold anything that can be assigned to an iconic variable (*Images*, *Regions*, ...).		

Color

Variables	*Postfix*	*Example*

Tuple [R,G,B] oder [R,G,B,A]	*_Color*	`* parameter MyAOI_Aoi`
		
		`* default value in main()`
		`MyAOI_Aoi_Color := [255, 0, 0, 255]`
		
		`* usage in HalconRun()`
		`MyAOI_Aoi_Color := [255, 0, 0, 255]`

• For some parameters (*AOIs*, *Regions, etc.),* you can specify an additional *Color* parameter which you can use to change or the read the *Color* with every run. • In the "*main()*", you can also add a *Color*, which will be applied once after loading the script. • Input *Regions* get their color from the linked *Region.* In this case, the *Color* cannot be set. • For inverted parameters you can also read and change the *Color*. The *Color* parameter must be inverted, too (look at inverted parameters).		

Transformation

Variables	*Postfix*	*Example*

Tuple (6D)	*_Trafo*	`* parameter output_Region`
		`* transformation parameter output_Region_Trafo`
		
		`* init identity trafo`
		`hom_mat2d_identity (HomMat2DIdentity)`
		`* set rotation`
		`hom_mat2d_rotate (HomMat2DIdentity, 0.78, 0, 0, output_Region_Trafo)`

- For some parameters (e.g. *Images*, *AOIs*, etc.), you can specify additional *Transformation* parameters, which you can use to change or the read the related *Transformation*.
- *Transformations* are passed as 6D Tuple (look at HALCON documentation for *hom_mat2d_**).
- When passing a *Transformation* parameter to an input parameter, in it you will pass the *Transformation* parameters from, for example, the "**Find object**" tool.
- For inverted parameters you can also read and change the *Transformation*. The *Transformation* parameter must be inverted, too (look at inverted parameters).

Calibration

Variables	*Postfix*	*Example*
Tuple	*_Calib*	`* image parameter MyInput_Img` `* calibration parameter name MyInput_Img_Calib`

- For *Image* parameters, you can specify additional *Calibration* parameters, which you can use to change or the read the related *Calibration*.
- *Calibration* data contain either
- no value: The image has no *Calibration*.
- 15 values: The image has *Calibration* transformation ("**Set up camera**" use calibration for: result or image), which can be used as follows:

```
 extract camera parameters
amPar := MyIn-
  put_Img_Calib[0:7]
 extract pose
ose := MyIn-
  put_Img_Calib[8:14]

 sample image coordinates
  in pixel
ow := 123
ol := 456

 transform to millimeter
mage_points_to_world_plane
  (camPar, pose, row, col,
  'mm', X_in_mm, Y_in_mm)
```

NOTE

Incorrect values (e.g. all 0) may lead to failing other tools, which use the calibration.

Transformation Link

Variables	*Postfix*	*Example*
	_TrafoLink	`* set transformation in main()` `input_aoi_Aoi_TrafoLink := 'input_image_Img'`

- *Transformations* of some parameters can be linked to other parameters.
- For example an *AOI* can be linked to an image: *<name_of_destination_parameter>_TrafoLink := '<name_of_source_parameter>'*.
- The parameters of a link must be used in the same procedure.
- Links do not have an effect across several procedures.
- A link applies to all procedures, which uses both parameters.

Array / ArrayCount

Variables	*Postfix*	*Example*

	_<type>Array	`* set default in main()`
		`ArrayCount := 3`
		`ArrayCount_MinMax := [3, 8]`
		
		`* set value in HalconRun()`
		`Test_IntArray := [0, 1, 2, 3, 4, 5]`

- *Arrays* can be input or output and used as inverted parameter.
- *Arrays* are possible in combination with *Int*, *Real*, *String*, *Bool*, and *MInt*.
- *ArrayCount* sets the number of elements of all *Arrays*.
- For each *Array* single elements with *<name> <number>* are created in the tool.
  - *ArrayCount* can be used as input parameter in "*HalconRun()*".
- MinMax of *ArrayCount*
  - has a maximum length of 32, and
  - *ArrayCount* is invisible, as soon as Min and Max are equal.
- Additional values are discarded when the number of values is higher than the number specified in *ArrayCount*.
- Missing values are filled with default values (zero / empty string).

Options

Variables	*Postfix*	*Example*
	_Options	`* set options in main()` `output_int_Int_Options := ['UnitMM']`

- It is possible to set the following *Options*: *UnitMM*, *UnitDEG*, *UnitPercent*, *UnitPixel*, *Invisible*, *Disabled, UseAsResult.*
- Only numerical output values (including arrays) support units. A parameter can only have on unit.
- Several options can be set, as long as they are supported by the respective parameter type.
- For Input parameters it is possible to set the option *UseAsResult*. Input parameter with this option will be displayed in the result list and can be selected in the action menu **Monitoring** or used as Input in subsequent tools. This option is not supported by *Image* parameters, *Array* parameters and invisible parameters (e.g. *Tuples*).

Virtual Image

Variables	*Postfix*	*Example*

	<out_name>_<in_name>_VImg	`* name of input image parameter example_input_Img`
		
		`* name of virtual image parameter example_ref_example_input_VImg`

• The input image with the name *<in_name>_Img* must be existent. • This resembles the behavior of the tools "**Find object**" and "**Read code**".		

Virtual Image Array

Variables	*Postfix*	*Example*

	<out_name>_<in_name>_VImgArray	`* name of input image parameter example_input_Img`
		
		`* name of virtual image parameter array example_ref_example_input_VImgArray`

• Works like *Virtual Image*, but as an array. • 6D *Transformation* tuples requires 6 x *ArrayCount* values. • Additional *Transformations* are discarded. • Missing *Transformations* are set to identity transformation (no rotation and no translation). • Incomplete *Transformations* (less than six values) are set to identity transformation, too.		

Non Transformed Image

Variables	*Postfix*	*Example*
	_NImg	
• Is used like normal *Image*, but as output only. • Can be used as input for "**Filter image**" tool. • Can't have a *Transformation*.		

ReferenceImage

Variables	*Postfix*	*Example*

	<name>_<name_of_input_image>_RImg	`* name of input image parameter example_input_Img`
		
		`* name of reference image parameter example_ref_example_input_RImg`

• *ReferenceImage* can only be input. • A button is displayed in the tool and if you click this button, a new reference image will be taken.		

Inverted Parameters

Variables	*Postfix*	*Example*

	_	`* Standard Input Parameter`
		`inputInt_Int`
		
		`* Output for HALCON, Input for HALCON tool`
		`inputInt_Int_`

- With inverted parameters you can write to an input parameter or read an output parameter (intended for Callbacks, "*HalconInit()*", and "*HalconFinalize()*" or to build a pass-through parameter).
- A parameter can be used as normal and inverted in the same procedure, this way a parameter can be used as "pass-through" parameter, the old value can be used to generate a new value.

Communication Parameters

Variables	*Postfix*	*Example*

Tuple	*_Com*
	_Com_Format
		`* set communication main()`
		`testCommunication_Com_Format := ['i', 'r', 's32']`
		
		`* usage in HalconRun()`
		`testCommunication_Com := [123, 456.789, 'teststringvalue']`

- It is possible to send and receive data via communication plugin similar to the "**Send results**" and "**Receive data**" tools.
- Can't be used inverted.
- Output: Is sent via communication plugin to PLC.
- Input: Current input from communication plugin (from PLC) is read.
- Format tuple in "*main()*":
  - 'i' => int32
  - 'r' => real32
  - 's####' => string with length (string is cut at length or filled with '\0')
- If the parameter tuple has more/less items then the format tuple, values are discarded or filled with default values.
- For output values the inspection ID (uint16) is added as first item.
- The resulting output length must not exceed about 64kB.

Serialized Parameters

Variables	*Postfix*	*Example*

	_Serialized	`* Serialization, parameter used as output inverted`
		`serialize_object (Region, region_Serialized_)`
		
		`* Deserialization: parameter used as input`
		`deserialize_object(Region, region_Serialized)`

- Can be used to store serialized HALCON tuples and icon objects in the tool.
- Does not support serialized models, e.g. shape models.
- Serialized parameters are control variables that contain serialization handles.
- The serialization/deserialization is done in the HALCON script, the actual saving and loading to flash is done automatically by the tool.
- When a serialized handle is passed as parameter it must not be cleaned in HALCON script.
- When something can't be deserialized the tuple is empty.
- Multiple items can be used with a single serialized parameter.
- It is possible to upload serialized data via the Upload button.

ToolResult

Variables	*Post-fix*	*Example*
Tuple [string] or [string,string]		`ToolResult := ['warn', 'Could not find peak']`
• The first string sets the tool processing (valid values are "**ok**", "**warn**", "**error**", and "**critical**"). • The second string is optional and sets the result message.		

ToolName

Variables	*Postfix*	*Example*
		`* set tool name in main()` `ToolName := 'MyFirstHalconScriptTool'`
• Sets the tool name when loading a script.		

InfoString

Variables	*Postfix*	*Example*
		`* set info string in main()` `InfoString := 'This is my first info string'`
• Sets the an information string in the input area of the tool.		

ScriptVersion

Variables	*Postfix*	*Example*
Tuple		`* set the script version in main()` `ScriptVersion := 1`
• Sets the version of the HALCON script. The HALCON tool checks this version while loading the HALCON program and creates a warning, if the version of the program does not match with the version of the tool. Versions will also be check during an update of the camera software. • The current script version of the HALCON tool is 1. • 1 will be default for HALCON programs, where the script version is not stated.		

Callback

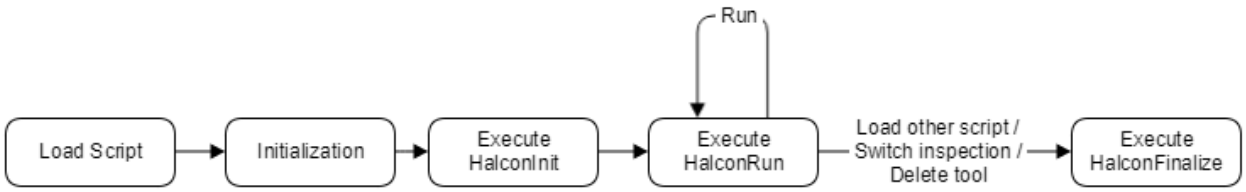
Variables	*Post-fix*	*Example*
Tuple [string,...]		`* list names of local procedures to be used as callbacks in main()` `HalconCallbacks := ['Test1', 'Test2']`
• Adds HALCON procedures which appear as Buttons in the tool. • These procedures are called when the button is pressed. • Can also be used to initialize variables. • Parameters of these procedures are named the same way they are named in "*HalconRun()*". • When a parameter has the same name in a different procedure, it refers to the same variable. • If input parameters are changed in the Callback, the tool will be updated and "*HalconRun()*" will be run once.		

Mask

Variables	*Postfix*	*Example*

- Is a *Region* parameter, that is handled as mask like in the "**Find object**" tool.
- *Mask* can be input (or used inverted), is drawable, and the name must be **Mask**.
- Only one *Mask* is possible.
- When a HALCON script with a mask parameter is loaded, the wizard icon appears .

Init/Finalize



Variables	*Postfix*	*Example*

- You can define HALCON procedures "*HalconInit()*" and "*HalconFinalize()*" to make initializations as soon as an inspection program is loaded and cleanups when a program is unloaded.
- "*HalconInit()*" will be called after starting of the camera, when changing an inspection program, and after a new inspection program was loaded in the action menu **Configuration**.
- "*HalconFinalize()*" will be called before changing an inspection program and before a new inspection program was loaded in the action menu **Configuration**.
- Parameters of init and finalize are named the same way they are named in "*HalconRun()*".
- When a variable has the same name in init, finalize or run, it refers to the same variable.

RunMode

Variables	*Name*	*Example*
	RunMode	

- Indicates the script is running in **Monitoring** (value = 1) or in other modes (Configuration, Initialization; value = 0).

System Information

Variables	*Name*	*Example*
Tuple	*SystemInfo*	

- Contains following information about the camera and the inspection:
  a. ID of the inspection program
  b. ID of the inspection result
  c. Fix string "obsolete", exists to maintain compatibility (string)
  d. Color support of the camera (1 = color camera (or PC version), 0 = gray scale camera)
  e. Device information (string)
  f. Information about I/O extensions (string)
- It is possible to load color images via the File Device on gray scale camera, too.

Loading (model) data

Beside HALCON programs, it is also possible to upload model data like fonts or classificatiors. Please note the following:

- Any kind of serialized HALCON data can be loaded in a serialized parameter.
- The data file name must start with the name of the serialized parameter.
- If the serialized file contains more then one item, a tuple with the numer of items must be serialized before the other items.
- File names can have any extension, except "**.hdev**" and "**.zip**".
- The data must be in the serialized HALCON format.
- After uploading serialized data, the tool will be reinitialized, i.e. e.g. *"HalconInit()"* be called again.

Example, how to generate a serialized region:

```
gen_circle (Circle, 500, 500, 100.5)
serialize_object (Circle, SerializedItemHandle)
open_file ('test_Serialized.bin', 'output_binary', FileHandle)
length := 1
serialize_tuple (length, SerializedItemHandle1)
fwrite_serialized_item (FileHandle, SerializedItemHandle1)
fwrite_serialized_item (FileHandle, SerializedItemHandle)
close_file (FileHandle)
clear_serialized_item (SerializedItemHandle1)
clear_serialized_item (SerializedItemHandle)
```

HALCON Debug Server

You can start the debug server of HDevEngine on the camera, HDevelop can connect with. Thus you can debug the script code with HDevelop on the camera. The following must be considered:

- The HALCON tool must be added to your inspection program.
- In the HALCON tool, the HALCON program you want to debug, must be loaded.
- Afterwards, you can click on the button *HDevEngine Debug Server:* ***Start*** in the System settings → category *Developer*. The debug server starts at its default port *57786*.
- Finally, you can connect to the debug server on the camera via the main menu *Execute* → *Attach To Process...* in HDevelop.

Limitations:

- You can only debug one HALCON tool, given that HDevelop cannot deal with more than one procedures with the same name (but all programs use "HalconRun()").
- It is not possible to debug "*HalconInit()*" or "*main()*", given that the program, you want to debug, must be loaded before the connection between camera and HDevelop.
- The version of HDevelop must be compliant to the HALCON version of the camera. The camera uses 19.X currently.
- Since the debug server is always started at the default port, it can only run on one instance.

You can stop the debug server using the button *HDevEngine Debug Server:* ***Stop*** on the camera.

Tips and tricks

General

- The tool runs a bit slower in action menu **Configuration** than in **Monitoring**.
- The camera software uses a different coordinate system as HALCON.
- You can find detailed error descriptions in the system log.
- Do not modify input *Images*, *Regions*, etc. This will lead to undesired results, especially in the action menu **Configuration**.
- Global variables in HALCON programs are global in all tools. However, the usage of global variables is not a good programming style.
- Part of programs in "*main()*", which are thought for working on a PC and not on the camera, can be put in a try/catch block. Thus, the same program will run on the camera and under HDevelop.
- You can download some sample programs via the online help of the tool.

Working with Handles

- You have to cleanup halcon handles (*object_model*, etc.) otherwise this will cause memory leaks. **Exception**: serialization handles for serialized parameters.
- You have to cleanup any handles that are allocated in the main procedure.
- HALCON handles must be stored in tuple variables when passed as parameters.

Usage of files

- You have to cleanup file handles.

Special case SMARTCAMERA

Try not to access files at all, use "*../images*" if you have to. This is the shared images directory.

ATTENTION

Risk of equipment damage

Do not change the current working directory. This can cause the configuration of the camera to contain errors and a reset of the complete configuration can be necessary.

ATTENTION

Loss of data

Do not change the current working directory. This can cause the configuration of the camera to contain errors and a reset of the complete configuration can be necessary.

6 Connection to the customer control system

6.1 Communication via UDP and TCP

It is possible to control BVS Cockpit using so-called UDP sockets or TCP sockets via the Ethernet interface of the host system. For this, the port 36701 is used. Using further BVS Cockpit instances in the same host system, the port numbers are incremented (instance 2 = port 36702, instance 3 = port 36703, etc.).

NOTE

To communicate via UDP, you have to set the *Mode* in the system menu ("*System settings -> communication*") to "**UDP**". To communicate via TCP, this *Mode* must be set to "**TCP**".

TCP and UDP sockets are essentially different. With a TCP socket first a connection between BVS Cockpit and the controlling system is opened. With a UDP socket this is not necessary, since individual packets are simply sent. In both cases BVS Cockpit expects a Connect message, then it starts to send out status updates and result data. If the controlling system does not want any more updates, it sends Disconnect. Likewise a Disconnect should be sent before the socket is closed. BVS Cockpit supports only one active client.

BVS Cockpit responds to a Connect message with a Connect response; in addition, it then sends a status message to the controlling system so that the latter is informed about the current status of BVS Cockpit.

Since TCP is based on data streams, multiple messages sent by BVS Cockpit may share a common TCP-header. It is therefore recommended to parse the user data length, which is sent with every message, to divide individual messages on the receiver side. To minimize latencies, Nagle's algorithm is deactivated in BVS Cockpit (option TCP_NODELAY). It is recommended to set option TCP_NODELAY in the controlling system also. In many cases this also prevents that multiple messsages share a common TCP header.

Since UDP is based on data packages, every message of BVS Cockpit corresponds exactly to one UDP message. On the other hand, UDP provides no guarantee, that sent messages are actually delivered. If transmission security is of great importance, use of TCP sockets is therefore recommended.

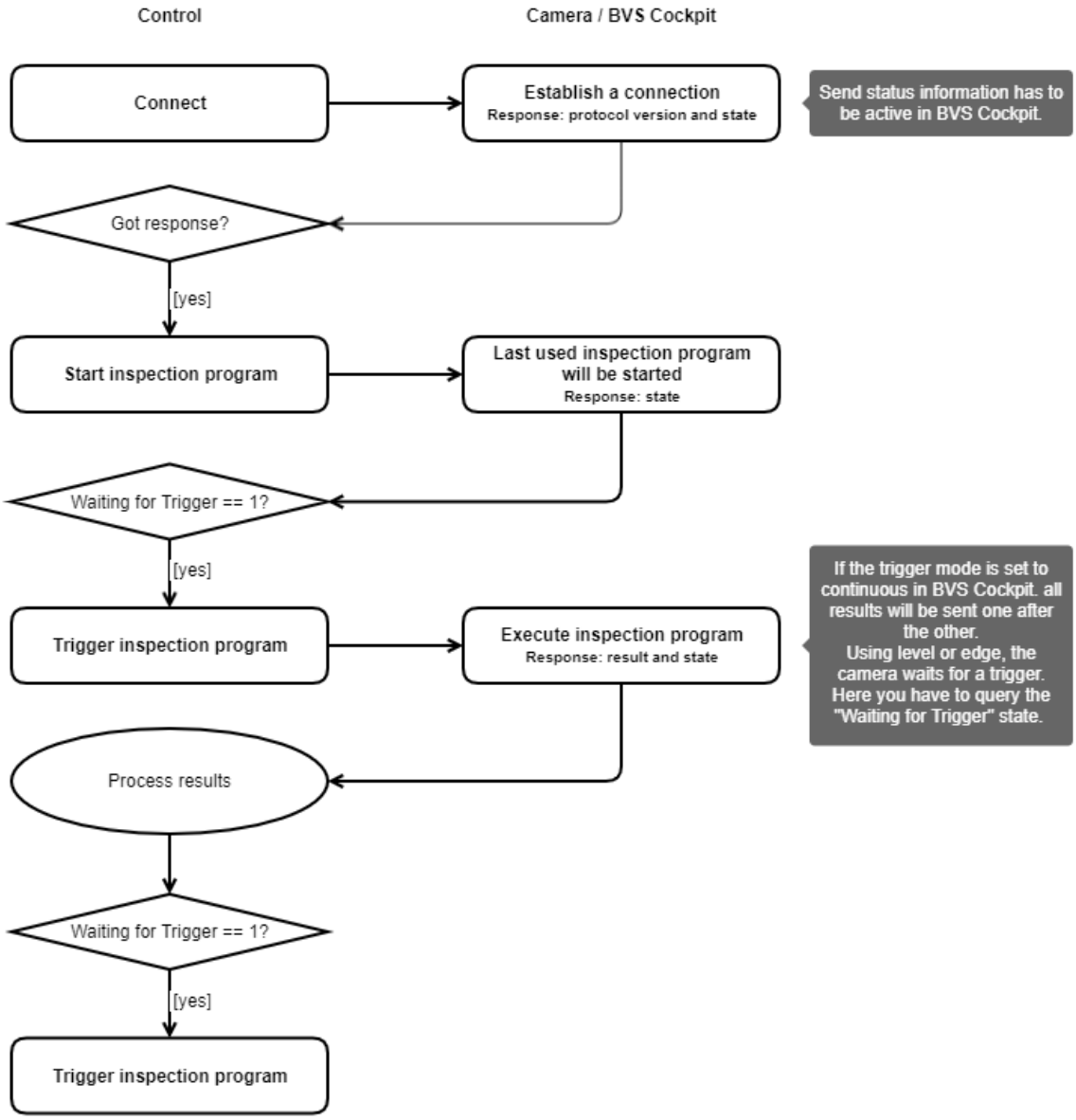
Invalid messages are ignored by BVS Cockpit.

NOTE

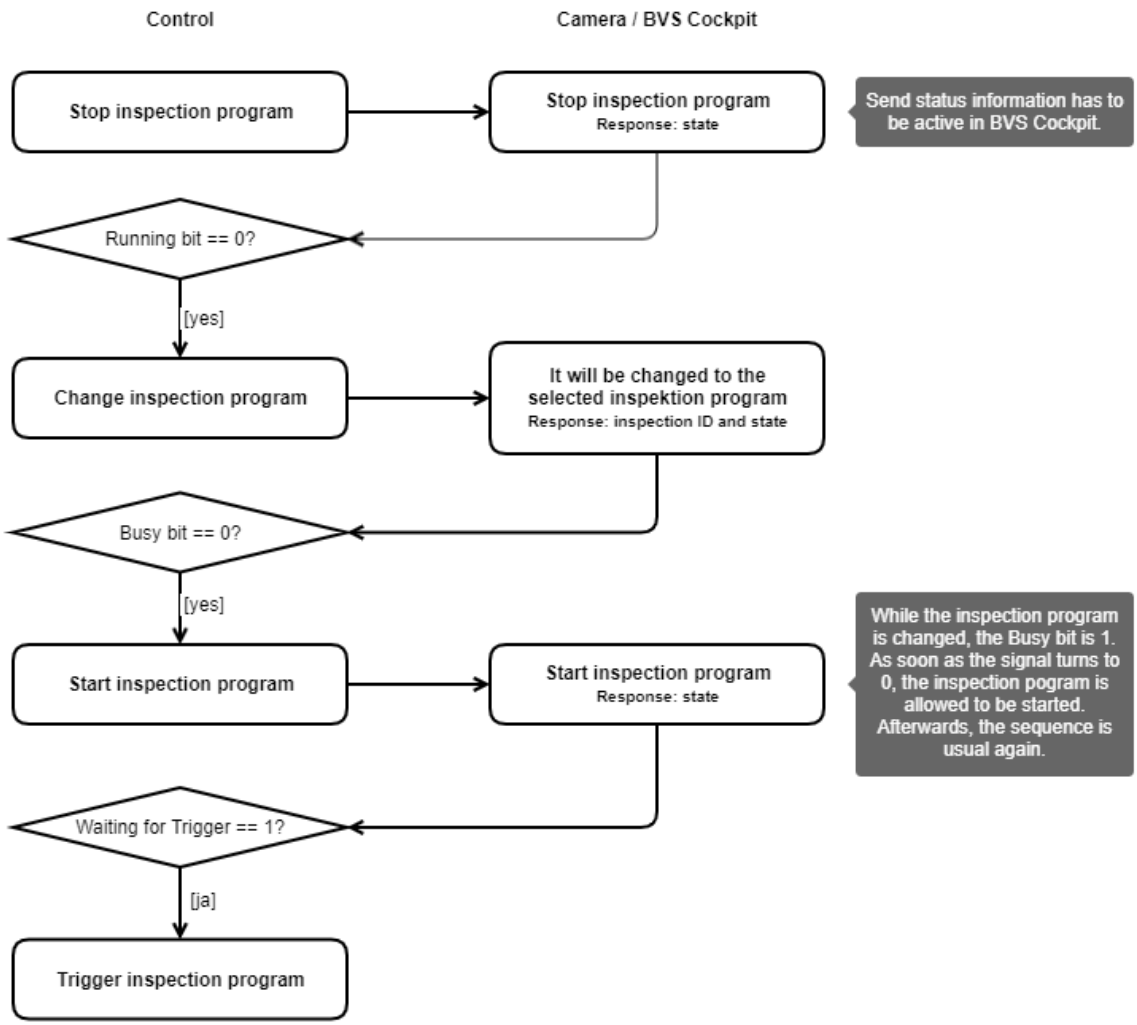
Messages from BVS Cockpit are sent asynchronously.

6.1.1 Communication sequence

Example 1: Start and trigger inspection program



Example 2: Change inspection program



6.1.2 Message structure

Most message elements are of type UINT16 and consist of 2 bytes. But over Ethernet one byte after the other is sent. The messages are transferred using the little-endian format. This means the lowest value byte is transferred first.

Each message begins with the message ID.

There are three different message categories:

1. *Messages from BVS Cockpit*
    a. Status message
    b. Result container
2. *Messages to BVS Cockpit without answer*
    a. Disconnect
    b. Set timestamp
    c. Set input data
    d. Set result number
    e. Trigger inspection program
    f. Start inspection program
    g. Stop inspection program

h. Restart camera

i. Write log message

3. *Messages to BVS Cockpit with an answer after an action*

a. Connect

b. Change inspection program

c. Get inspection program ID

d. Get timestamp

All messages have the same structure:

Length in bytes	Structure	Description
2	UINT16	Message ID
2	UINT16	User data length in bytes
		User data

Messages are ignored by BVS Cockpit which do not comply with the given format (wrong length, unknown message ID).

Every message has a unique ID:

ID	Message
01_{hex}	Connect
02_{hex}	Disconnect
12_{hex}	Status message
20_{hex}	Result container
30_{hex}	Get inspection program ID
31_{hex}	Change inspection program
32_{hex}	Get timestamp
33_{hex}	Set timestamp
34_{hex}	Set input data
35_{hex}	Set inspection result number
40_{hex}	Trigger inspection program
41_{hex}	Start inspection program
42_{hex}	Stop inspection program
43_{hex}	Restart
$F2_{hex}$	Write log message

NOTE

It is not guaranteed that an interrupted network connection via TCP is detected promptly. For this reason, a watchdog functionality is recommended, which checks the connection in a defined time interval. For this, you can use the *Connect* message which responds with a state message. You can also use "Get inspection program ID", which responds with the ID of the current inspection.

Connect

The *Connect* message creates a connection to BVS Cockpit.

Length in bytes	Structure	Value	Description
2	UINT16	01_{hex}	Message ID
2	UINT16	00_{hex}	User data length in bytes

BVS Cockpit answers with a *Connect* message.

Length in bytes	Structure	Value	Description
2	UINT16	01_{hex}	Message ID
2	UINT16	02_{hex}	User data length in bytes
2	UINT16	02_{hex}	Version of the protocol

The following examples show how the connect message is structured and sent:

The messages ID for Connect is 01_{hex}, as UINT16, little-endian: 01_{hex} 00_{hex}

Then the user data length follows. The Connect message has no user data and so the length is Null. As UINT16 in little-endian: 00_{hex} 00_{hex}

There are no user data, the message is now complete.

The entire Connect command is thus:

00_{hex} 01_{hex}
00_{hex} 00_{hex}

The message is coded in C language:

```
unsigned char[ ] connect =
{
0 x 01, 0 x 00,
0 x 00, 0 x 00
};
```

Therefore:

Message from the client to BVS Cockpit: 01 00 00 00

BVS Cockpit responds to it with: 01 00 02 00 02 00

Disconnect

With the *Disconnect* message the connection will be disconnected. Afterwards, no more result containers or status messages will be transmitted; messages to BVS Cockpit will be ignored.

Length in bytes	Structure	Value	Description
2	UINT16	02_{hex}	Message ID
2	UINT16	00_{hex}	User data length in bytes

Example:

Message from the client to BVS Cockpit: 02 00 00 00

Status message

For each status change BVS Cockpit sends a status message. If multiple status changes occur during a short period of time, they will be summarized in a common status message.

Length in bytes	Structure	Value	Description
2	UINT16	12_{hex}	Message ID
2	UINT16	02_{hex}	User data length in bytes
2	UINT16		Status word. A set bit indicates that the status is active (see the following table). In contrast to the fieldbus interface, the status bits do not have to be reset actively. If a status is no longer active, a new status message is sent.

Structure of status word:

Value (dez)	Value (hex)	Bit	Name	Description
1	01 00	0	Overheat	The connected device exceeded the permissible temperature. This could lead to a drop in quality of the inspection results. • Check ambient temperature • Check installation conditions
2	02 00	1	System Error	Fatal system error. • The inspection program could not run • The connection to the camera is lost or the camera is defective
4	04 00	2	Input Status	It is set for input container overflow (more input containers than inspection results, so that one input container was discarded) or underflow (more inspection results than input containers, so that one input container was used multiple times).
16	10 00	4	Busy	This bit signals that BVS Cockpit is currently busy. It is set while the camera performs an administrative task, that prevents it from processing messages. The bit is particularly set while the Manage inspection programs dialog or the System Settings dialog is opened, the user resides in the Configuration mode, during the change of the inspection program, export, import, or duplication of an inspection program and during creation or loading of a backup. Furthermore it is set during the startup of BVS Cockpit including the import of the last selected inspection program and while the inspection program is being automatically saved due to its modification in

				configuration mode. BVS Cockpit is not considered to be busy during execution of an inspection program. For sequence control during an inspection run, the bits *Running* and *Waiting For Trigger* are provided.
32	20 00	5	Report	This bit becomes active as soon as a report file could not be transferred and got lost.
64	40 00	6	Simulation Mode	This bit signals that either the File Device or the Simulation mode in the Communication Settings of the System Settings is active.
128	80 00	7	Running	This bit signals that BVS Cockpit is operating in Run mode.
256	00 01	8	Overtriggered	This bit signals that multiple trigger signals were issued before processing was complete. This suggests that image processing is slower than image acquisition and there is a risk that images may be lost.
512	00 02	9	Waiting For Trigger	This bit signals that BVS Cockpit is waiting for a trigger signal.

Example 1:

Status message from BVS Cockpit to the client: 12 00 02 00 C0 00

The status word C0 00 can be interpreted as: 0b 1100 0000 0000 0000. The 6th and the 7th bit are high.

Therefore this status message shows that BVS Cockpit is operating in the Run mode and either the file device or the simulation mode is active.

Example 2:

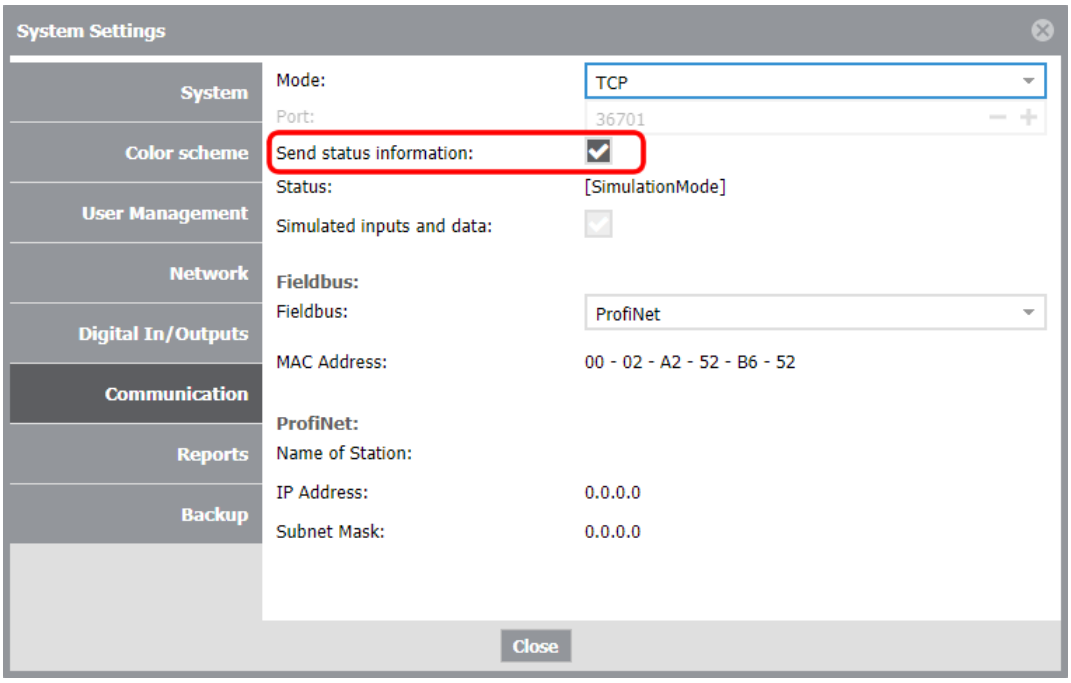
Status message from BVS Cockpit to the client: 12 00 02 00 80 02

The status word 80 02 can be interpreted as: 0b 1000 0000 0000 0010. The 7th of the low byte and the 2nd bit of the high byte are high.

Therefore this status message shows that BVS Cockpit is operating in the Run mode and "Waiting for trigger" is active.

NOTE

In BVS Cockpit the option *Send status information* should be activated. Afterwards, every action will send a state and a regular communication sequence is guaranteed.



Result container

If an inspection program uses a "**Send results**" tool, the inspection program will send the result data in this message after processing the inspection program.

Length in bytes	Struc-ture	Value	Description
2	UINT16	20_{hex}	Message ID
2	UINT16		User data length in bytes (the inspection program ID is considered as user data)
2	UINT16		Inspection program ID. BVS Cockpit can manage various inspection programs. Each of this program has a unique inspection program ID. This section indicates the inspection program, which created the result container.
			Result data as specified tool "**Send results**". Bool value "false" is sent as 0x00, "value" true as 0xFF.

Example of how a result container is formed:

After each completed inspection BVS Cockpit sends a results container to the controlling system. The structure of the user data depends on the configuration in BVS Cockpit. In this example it is assumed that an inspection was configured which provides a string of 20 bytes (= 14_{hex}). BVS Cockpit reads the text “Hello World” and sends it to the controlling system.

The coding of the text is in ASCII format. The text that is read (11 bytes long) is shorter therefore than the area provided in the message (20 bytes). BVS Cockpit therefore fills the remaining bytes with zeros. One byte chain is simply sent after the other. Little-endian has no meaning here since these are individual bytes.

The entire message is thus:

20_{hex} 00_{hex} (Message ID)
16_{hex} 00_{hex} (user data length = 2 bytes for inspection program ID + 20 bytes result data)
15_{hex} 00_{hex} (inspection program ID: 21)
48 65 6C 6C 6F 20 57 6F 72 6C
64 00 00 00 00 00 00 00 00 00
(ASCII: 'H', 'e', 'l', 'l' 'o', space, 'W', 'o', 'r', ‚l‘, 'd', 9× fill-bytes)

Get inspection program ID

Gets the program number of the active inspection program.

Length in bytes	Structure	Value	Description
2	UINT16	30_{hex}	Message ID
2	UINT16	00_{hex}	User data length in bytes

BVS Cockpit answers with a response message containing the program ID.

Length in bytes	Structure	Value	Description
2	UINT16	30_{hex}	Message ID
2	UINT16	02_{hex}	User data length in bytes
2	UINT16		Inspection program ID

Example:

Message from the client to BVS Cockpit: 30 00 00 00

BVS Cockpit responds to it with: 30 00 02 00 15 00

The last 2 bytes representing the applcation ID (15 00) is interpreted as 21 in decimal system.

NOTE

Users with the appropriate rights can change the inspection programs using the BVS Cockpit user interface.. It cannot be excluded that the control does not recognize a change. For this reason, you should check periodically if the right inspection program is loaded,

Change Inspection program

A change is possible, when the following conditions are fulfilled.

- BVS Cockpit is not in the config mode.
- The current inspection program is stopped.
  Running bit is 0.
- The number of the inspection program which should be loaded is not zero.
  You can check and set the ID using BVS Cockpit.
- The inspection program which should be loaded is available.
  You can check the ID using BVS Cockpit.
- No other inspection program is currently loaded or duplicated.
  The Busy bit hast to be 0 before.

Length in bytes	Structure	Value	Description
2	UINT16	31_{hex}	Message ID
2	UINT16	02_{hex}	User data length in bytes
2	UINT16		Inspection program ID

BVS Cockpit answers with a response message containing the program number.

Length in bytes	Structure	Value	Description
2	UINT16	31_{hex}	Message ID
2	UINT16	04_{hex}	User data length in bytes
2	UINT16		Error flag
2	UINT16		Inspection pogram ID

The error flag has a value of 00_{hex} if the mentioned conditions are fulfilled and the inspection program gets changed. In this case the program ID of the response message is equal to the number in the change request. If one of the conditions is not fulfilled the inspection program is not changed. The message then contains an error flag of 01_{hex} as well as the program ID of the program that is still loaded. The response message may be sent before the inspection program is completely loaded. Therefore, the response provides the information, that a change of the inspection program is possible, but not if the loading can be completed successfully.

NOTE

To check if loading is completed, the Busy bit of the status message has to be evaluated: When loading starts, it changes to 1, when loading is completed, it changes back to 0.

Additionally, one should check if a status message with System Error bit is sent during load. If the inspection program has got a defect that causes loading to be aborted, the system error bit will be set to 1. Instead a new standard inspection program will be loaded. After loading of the standard inspection program is completed, the bit remains set until a valid inspection program is loaded.

Example:

Let the desired inspection program ID be 1500_{hex} or 21 in decimal.

Message from the client to BVS Cockpit: 31 00 02 00 15 00

If BVS Cockpit responds to it with: 31 00 04 00 00 00 15 00, the error flag (00_{hex}) indicates that the inspection program can be changed and the inspection program with ID 15_{hex} will be loaded.

If BVS Cockpit responds to it with: 31 00 04 00 01 00 36 00, the error flag (01_{hex}) indicates that the inspection program cannot be changed and the old program with ID 36_{hex} will stay loaded.

If BVS Cockpit responds to it with: 31 00 04 00 01 00 36 00 and is followed by a status message with active system error bit (e.g. 12 00 02 00 12 00), the error flag (00_{hex}) indicates that the inspection program can be changed , but the inspection program with ID 15_{hex} has got a defect and can thus not be loaded. Instead a new default inspection will be loaded.

Get timestamp

Get the current date and time.

Length in bytes	Structure	Value	Description
2	UINT16	32_{hex}	Message ID
2	UINT16	00_{hex}	User data length in bytes

BVS Cockpit answers with the current date and time.

Length in bytes	Structure	Value	Description
2	UINT16	32_{hex}	Message ID
2	UINT16	$0C_{hex}$	User data length in bytes
2	UINT16		Year
2	UINT16		Month
2	UINT16		Day
2	UINT16		Hour
2	UINT16		Minute
2	UINT16		Second

Example:

Message from the client to BVS Cockpit: 32 00 00 00

BVS Cockpit responds to it with: 32000C00E007060019000A000B000C00, which means that the date and time is 2016-06-25 10:11:12

Set timestamp

Set date and time.

Length in bytes	Structure	Value	Description
2	UINT16	33_{hex}	Message ID
2	UINT16	$0C_{hex}$	User data length in bytes
2	UINT16		Year
2	UINT16		Month
2	UINT16		Day
2	UINT16		Hour
2	UINT16		Minute
2	UINT16		Second

There is no response to this message. Invalid values are rounded to the next valid value.

To set date and time to 2016-06-25 10:11:12 send the following data (hex): 3300 0C00 E007 0600 1900 0A00 0B00 0C00

NOTE

If an NTP server is specified, the device automatically obtains the time from the server. Use of the Set timestamp message is then not necessary. The manually set time is overwritten by that of the NTP server.

Set input data

Some inspection programs require input data which are compared with the acquired data of BVS Cockpit. These data can be sent to BVS Cockpit using this message. The tool "**Receive data**" interprets the input data.

Length in bytes	Structure	Value	Description
2	UINT16	34_{hex}	Message ID
2	UINT16		User data length in bytes
			Input data for the inspection program as specified in BVS Cockpit. If a bool value is expected, 0x00 is interpreted as "false", all other values as "true".

Example:

Message from the client to BVS Cockpit: 34 00 0B 00 48 65 6C 6C 6F 20 57 6F 72 6C 64

Converting the message to its ASCII equivalent gives: Hello World

Trigger inspection program

It is sent from the client to start an inspection pass. The behavior depends on the setting in the BVS Cockpit. If an inspection is untriggered, this message is being ignored. To check whether the device is ready to receive a trigger, the Waiting For Trigger bit of the status message may be evaluated.

Length in bytes	Structure	Value	Description
2	UINT16	40_{hex}	Message ID
2	UINT16	00_{hex}	User data length in bytes

Example:

Message from the client to BVS Cockpit: 40 00 00 00

NOTE

The *trigger inspection program* command will capture a single image only if the *trigger mode* is configured to edge in the tool "**Setup camera**".

Start inspection program

Is sent to BVS Cockpit to start the inspection program. The behavior depends on the settings. Untriggered inspection programs will start and will transmit results continuously while triggered inspection programs will wait for a trigger signal. A user which is in the restricted area **configuration** is relocated to the **monitor** section, after the *start inspection program* command is received. There is no response message to this request; to get the point in time at which the inspection starts, the change of the Running bit in the status message from 0 to 1 can be awaited.

Length in bytes	Structure	Value	Description
2	UINT16	41_{hex}	Message ID
2	UINT16	00_{hex}	User data length in bytes

Example:

Message from the client to BVS Cockpit: 41 00 00 00

Status message from BVS Cockpit: 12 00 02 00 80 00, which represents that BVS Cockpit is running.

NOTE

Start inspection program should check the Busy bit of the status message before. The Busy bit should be 0.

Stop inspection program

Is sent to BVS Cockpit to stop an inspection program. Untriggered inspection programs will stop and will not send results anymore while triggered inspection programs will ignore further trigger signals. There is no response message to this request; to get the point in time at which the inspection stops, the change of the Running bit in the status message from 1 to 0 can be awaited.

Length in bytes	Structure	Value	Description
2	UINT16	42_{hex}	Message ID
2	UINT16	00_{hex}	User data length in bytes

Example:

Status message from BVS Cockpit before"Stop inspection pogram " message: 12 00 02 00 80 00

Message from the client to BVS Cockpit to stop inspection: 42 00 00 00

Status message from BVS Cockpit after "Stop inspection pogram " message: 12 00 02 00 00 00

Restart

Restarts the used device. After this message, the network connection will be lost and will be established after the restart. The restart will take approximately 30 seconds.

Length in bytes	Structure	Value	Description
2	UINT16	43_{hex}	Message ID
2	UINT16	00_{hex}	User data length in bytes

Example:

Message from the client to BVS Cockpit: 43 00 00 00

Set inspection result number

Every inspection pass generates a result that contains a unique sequence number. This sequence number is displayed in the BVS Cockpit. After the restart, the sequence number starts with zero. This message allows the client to set the sequence number to a defined value. The next inspection pass will then receive this sequence number. With every additional inspection pass, the sequence number is being incremented.

Length in bytes	Structure	Value	Description
2	UINT16	35_{hex}	Message ID
2	UINT16	04_{hex}	User data length in bytes
4	UINT32		New inspection result number

Example:

To set the sequence number to $560783366_{decimal}$ / 03 57 B0 00_{hex}

Message from the client to BVS Cockpit: 35 00 04 00 03 57 B0 00

NOTE

Contrary to other elements, the inspection result number is encoded using 4 bytes.

Write log message

Writes a log message into the log file "user.log". The string has to be in ASCII.

Length in bytes	Structure	Value	Description
2	UINT16	$F2_{hex}$	Message ID
2	UINT16		User data length in bytes (length of the string)
			Log message to be written in ASCII

Example:

Message from the client to BVS Cockpit: F2 00 0B 00 48 65 6C 6C 6F 20 57 6F 72 6C 64 (represents 'Hello World' in ASCII)

6.1.3 Testing the communication

Under Linux

For testing the UDP communication you can use the tool **Netcat** under Linux. The following console prompts will open a connection, start and stop the current inspection program, and close the connection.
In this example the IP address is 192.168.1.230. The port 60001 is used to get results.

```
# Connect
echo -n -e '\x01\x00\x00\x00' | nc -uw 1 -p 60001 192.168.1.230 36701
# Start inspection program:
echo -n -e '\x41\x00\x00\x00' | nc -uw 1 -p 60001 192.168.1.230 36701
# Stop inspection program:
echo -n -e '\x42\x00\x00\x00' | nc -uw 1 -p 60001 192.168.1.230 36701
# Disconnect:
echo -n -e '\x02\x00\x00\x00' | nc -uw 1 -p 60001 192.168.1.230 36701
```

6.2 Communication via fieldbus

NOTE

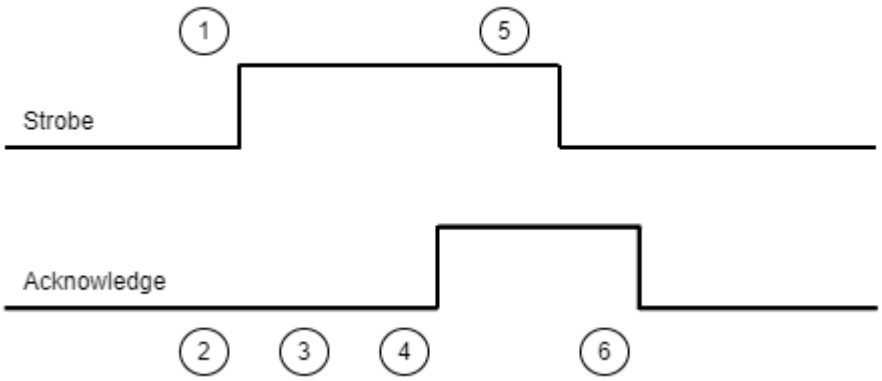
The communication via fieldbus is only available with the appropriate hardware.

Two buffers are needed to exchange data and commands between BVS Cockpit and the fieldbus controller (input buffer and output buffer). The buffer contents are exchanged using cyclical polling. When writing to the buffer, the transmitted data from the previous cycle is overwritten. If some bytes are not being used in a cycle, then they retain the last value.

Most of the commands are executed as command-response pair via a handshake mechanism. The PLC "sends" a command to the Balluff SmartVision Controller and the Balluff SmartVision Controller "responds" with a result. This process is synchronized with two bits in the cyclical memory and referred to as handshake. The
handshake bits are called strobe and acknowledge bits.

6.2.1 Handshake mechanism

Basic sequence of the PLC data exchange

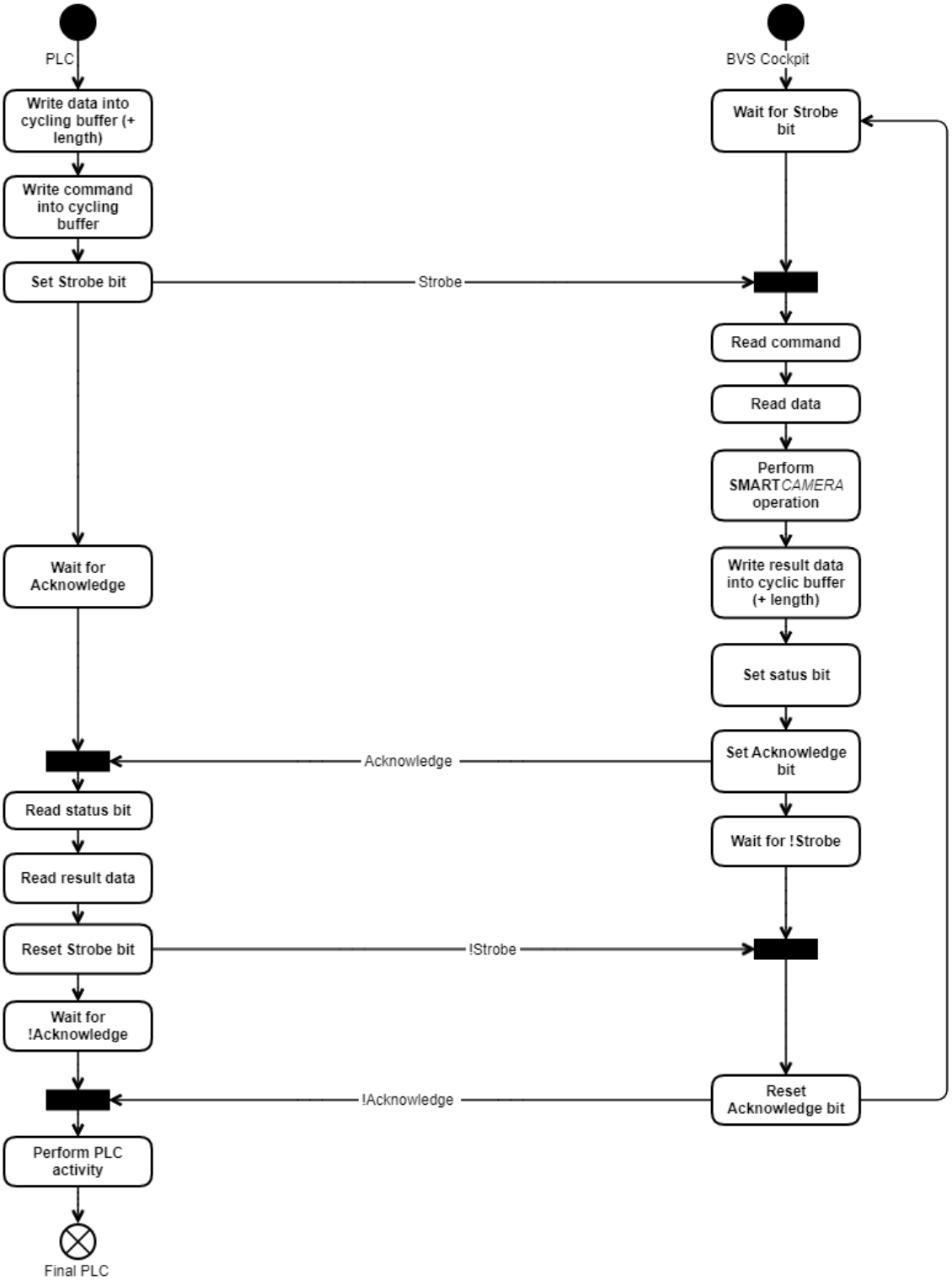


1. The PLC sets command ID, data length and data in the output buffer and then the strobe bit.
2. The strobe bit tells BVS Cockpit that a job is being transferred and the data to be transmitted are valid.
3. BVS Cockpit accepts the command and executes it.
4. BVS Cockpit correctly executed the command and sets command status, data length and data in the input buffer and then the acknowledge bit.
5. The PLC sees the acknowledge bit, reads the data from BVS Cockpit and then resets the strobe bit.
6. BVS Cockpit sees the reset strobe bit and resets the acknowledge bit. BVS Cockpit is ready for the next command.

NOTE

Please note the double bit string chapter in "Process Data Buffer".

Handshake flow chart



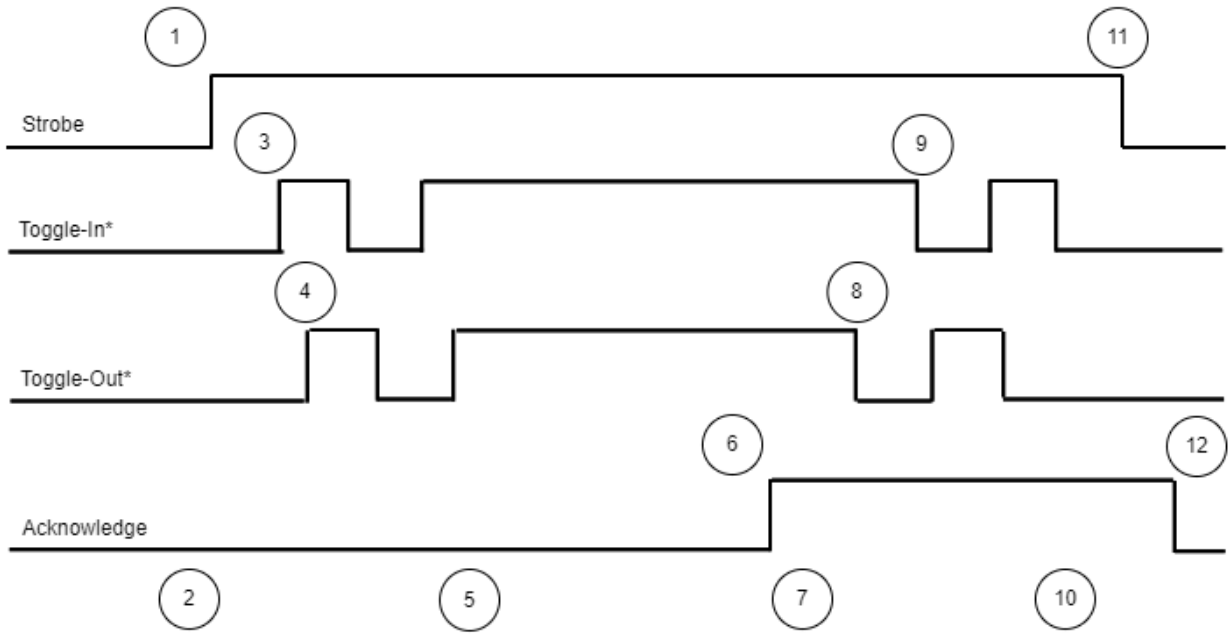
6.2.2 Toggle mechanism

If more data is to be transferred with a command or with a response than the cyclical memory area can hold, a toggle mechanism within the handshake is being used. For this purpose, two additional bits (Toggle-In and Toggle-Out) are used in the cyclical memory.

NOTE

Please note the double bit string chapter in "Process Data Buffer".

Process



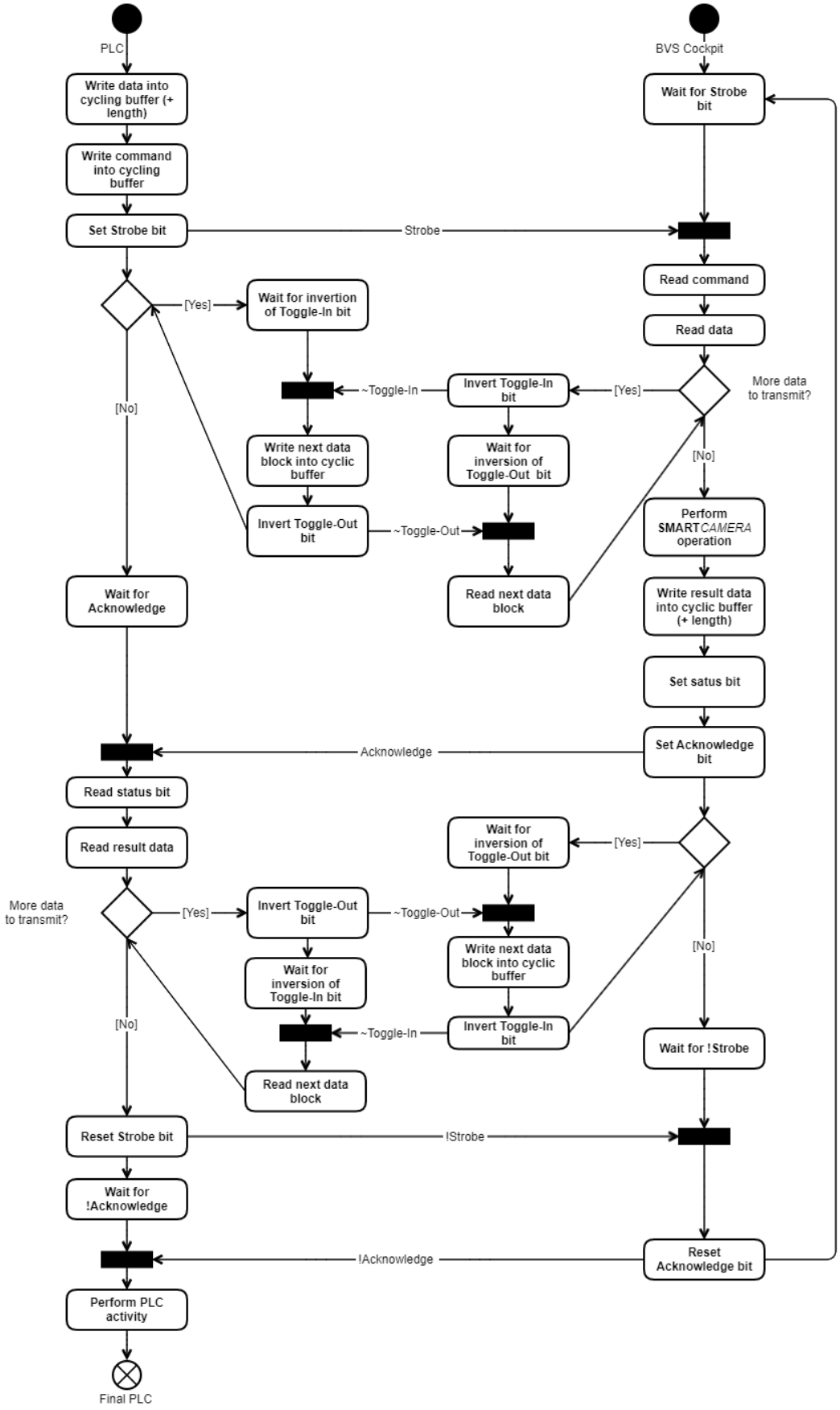
* In the example, Toggle-In/Out start at 0, the actual start value does not matter.

1. The PLC sets command ID, data length and the first part of the data in the output buffer. The data length is the length of the entire data to be transferred. Afterwards, the PLC sets the strobe bit.
2. BVS Cockpit reads the strobe bit and then command ID, data length and (partial) data. Basedon the data length and size of the cyclical buffer, BVS Cockpit recognizes that additional data still have to be transferred.
3. BVS Cockpit inverts the Toggle-In bit and waits for the inverted Toggle-Out bit.
4. The PLC reads the inverted Toggle-In bit, places the next partial data in the output buffer and then inverts the Toggle-Out bit.
5. BVS Cockpit reads the inverted Toggle-Out bit and accepts the next partial data. If additional partial data must be transferred, continue with 3, otherwise with 6.
6. BVS Cockpit accepts the command and executes it and sets command status, data length and (partial) data in the input buffer and then the acknowledge bit.
7. The PLC reads the acknowledge bit and then command status, data length and (partial) data. Based on the data length and size of the cyclical buffer, the PLC recognizes that additional data still have to be transferred.
8. The PLC inverts the Toggle-Out bit and waits for the inverted Toggle-In bit.
9. BVS Cockpit reads the inverted Toggle-Out bit, places the next partial data in the input buffer and inverts the Toggle-In bit.
10. The PLC reads the inverted Toggle-In bit and accepts the next partial data. If additional partial data must be transferred, continue with 8, otherwise with 11.
11. The PLC resets the strobe bit.
12. BVS Cockpit reads the reset strobe bit, withdraws the acknowledge bit and is ready for the next command.

NOTE

Wait processes are assigned a timeout of 250 milliseconds (Balluff SmartVision Controller) or 500 milliseconds (**SMART***CAMERA*) that can be switched off. (see chapter "Toggle Mechanism", step 4(3), 9 and 12) If no response occurs during this timeout, the handshake will be canceled and it will be returned to the initial status. The PLC should also monitor a corresponding timeout.

Handshake flow chart with toggle



6.2.3 Process Data Buffer

Double bit string

To ensure the complete transmission of all the data in the data buffer, the bytes are transferred with the control bits (bit strings) at the start and end of the data buffer for the process data and compared. If both bit strings are the same, then the data between these bytes were updated completely and can be taken over. This means that the data for each R/W head is only valid if both bit headers are the same. The host control system must also compare the bits in the bit headers.

Endianness

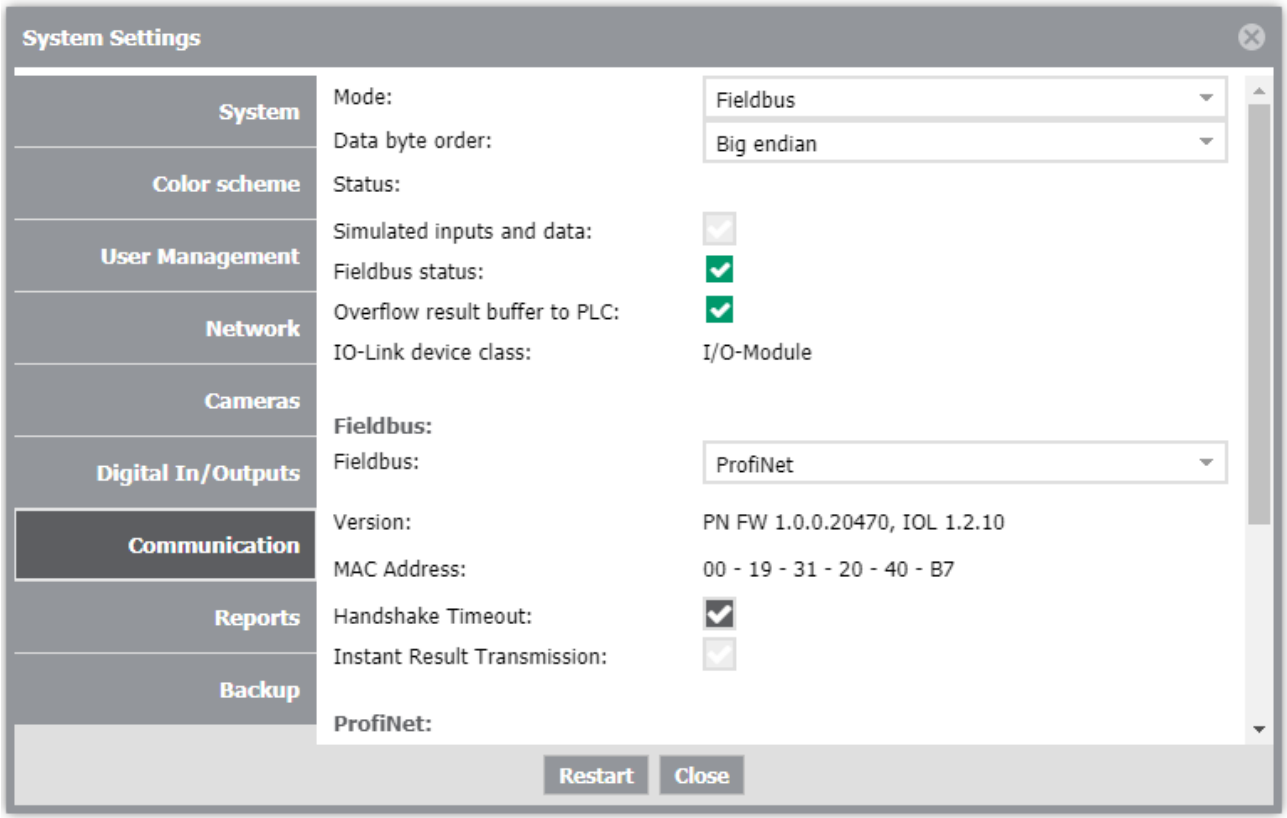
Endianness refers to the memory organization for simple numeric values, primarily the storage of integers in the working memory.

- For big endian, the highest value byte is stored first, i.e. at the lowest memory address.
- For little endian, on the other hand, the lowest value byte is stored at the start address.

For the values transmitted via fieldbus interface, the endianness can be set in the BVS Cockpit.

Instant Result Transmission

If the "**Instant Result Transmission**" mode is activated the result data will be sent via the fieldbus by BVS Cockpit immediately after they are available. No "Get Results" command to the camera is required. The "Result Ready" bit indicates that result data are available. If the results are not read, it can be possible that they will be overwritten by new result data. Then the "Buffer Error" bit is set and BVS Cockpit will show the state "Result Overflow". A segmentation of the data is not possible and the max. data length depends on the fieldbus configuration (e. g. input module length with ProfiNet-IO).



NOTE

After activation or deactivation of the "**Instant Result Transmission**" mode on a SmartVision Controller, a reboot of the SmartVision Controller is necessary. After the reboot, the chosen setting is valid for all instances of BVS Cockpit.

Output buffer

The control commands are transferred via the output buffer. Control commands are triggered either as individual bit or via handshake. It will be responded to commands transferred via handshake with a status and possibly result data.

Subaddress	
00_{hex}	Control bits
01_{hex}	Control bits (double bit string)
02_{hex}	Command byte - command code
03_{hex}	Data length byte 0 [1)]
04_{hex}	Data length byte 1 [1)]
05_{hex}	Data byte 0
…	Data byte n
Last byte = byte 1	Control bits (double bit string)

[1)] Byte 0 and byte 1 of the data length are processed as a whole-number 16 bit value.

NOTE

The sub address 00hex corresponds to the lowest address byte in the hardware configuration of the PLC.

Structure byte 0 (PLC → Camera) Control bits

BVS Cockpit responds to 0-1 transitions of the individual command bits. The bits should be set until the corresponding acknowledgment arrives.

Subaddress	Bit number	Name	Function description
00_{hex}	0	Trigger bit	Initiates a trigger It is acknowledged by resetting the WaitingForTrigger bit.
	4	Reset warning bit	Resets warning messages (warning bits). Is cleared by resetting the corresponding warning bit as long as the indicated status is no longer active.
	5	Reset bit	Triggers a reset which takes approx. 30 seconds. It is acknowledged by resetting the Ready bit.
	6	Start run mode bit	It activates the Run mode[1)] of BVS Cockpit. It is acknowledged by setting the Running bit.
	7	Stop run mode bit	It deactivates the Run mode[1)]of BVS Cockpit. It is acknowledged by resetting the Running bit.

[1)] If the Start and Stop run modus bits are set from 0 to 1 at the same time, then the status of the Run mode is not defined (random).

Structure byte 1 (PLC → Camera) Check bits to control the handshake

Subaddress	Bit number	Name	Function description
01_{hex}	0	Handshake strobe bit	The PLC uses this bit to signal that a command is to be transferred and that valid data is in the command byte, data length bytes and data bytes.
	1	Toggle-Out bit	The PLC uses this bit to signal that additional data can be read or written.

NOTE

Please note the double bit string. You have to set the same check bits in the last byte of the buffer.

Commands

The PLC writes command codes in byte 2 of the output buffer. The following codes are allowed:

Name	Command code	Meaning
Switch Application	01_{hex}	Switches the inspection.
Get Application ID	02_{hex}	Gets the ID of the currently active inspection.
Get Results	03_{hex}	Gets the result container.
Set Inputs	04_{hex}	Sets the input data.
Get Camera Info	05_{hex}	Gets the camera information.
Get Date Time	06_{hex}	Gets the timestamp.
Set Date Time	07_{hex}	Sets the timestamp.
Set Sequence Number	08_{hex}	Sets the sequence number.

Input buffer

The input buffer is used to transfer responses (status information and results) to the PLC.

Subaddress	
00_{hex}	Status bits
01_{hex}	Warn bits
02_{hex}	Control bits (double bit string)
03_{hex}	Command status byte
04_{hex}	Data length byte 0 [1)]
05_{hex}	Data length byte 1 [1)]
06_{hex}	Data byte 0
…	Data byte n
Last byte = byte 2	Status/control bits (double bit string)

[1)] Byte 0 and byte 1 of the data length are processed as a whole-number 16 bit value.

Structure byte 0 (Camera → PLC) Control bits

Subad-dress	Bit num-ber	Name	Function description
00_{hex}	0	Waiting for Trig-ger	This bit signals that BVS Cockpit is wating for a trigger signal.
	1	Result Ready	This bit signals that a result is ready to be picked up.
	2	Instant Result	This bit signals that the mode "Instant Result Transmission" is on.
	3	-	
	4	-	

	5	Busy	This bit signals that BVS Cockpit is currently busy. It is set while the camera performs an administrative task, that prevents it from processing messages. This is the case, when
			• BVS Cockpit is booted including the import of the last selected in-spection program during startup
			• the Manage inspection programs dialog or the System Settings dia-log is opened
			• the user resides in the Configuration mode
			• the inspection program is
			  ○ changed,
			  ○ exported,
			  ○ imported,
			  ○ duplicated, or
			  ○ automatically saved (due to its modification in configuration mode) and also during
			• a backup is
			  ○ created or
			  ○ loaded.
			BVS Cockpit is not considered to be busy during execution of an inspection program. For sequence control during an inspection run, the bits *Running* and *Waiting For Trigger* are provided.
			Some commands are rejected in this condition with the status NOK Busy.

	6	Running	This bit signals that BVS Cockpit is operating in Run mode.
	7	Device Ready	This bit is active after a restart only if the internal camera module has been completely started.

Structure byte 1 (Camera → PLC) Information bits

Subad-dress	Bit num-ber	Name	Function description
01_{hex}	0	Overtrig-gered [1)]	A trigger was discarded by BVS Cockpit since it was still busy with the analy-sis of the previous image.
	1	Buffer Error bit [1)]	This bit becomes active as soon as a result container was not picked up by the PLC and overwritten by a new result.
	2	Report [1)]	This bit becomes active as soon as a report file could not be transferred and got lost. **NOTE** A succesfull file transfer to the FTP server must be made prior to clearing the bit.
	3	Simula-tion Mode	This bit signals that either the File Device or the Simulation mode in the Com-munication Settings of the System Settings is active.
	4	Input Con-tainer [1)]	It is set for input container overflow (more input containers than inspections, so that one input container was discarded) or input container underflow (more inspections than input containers, so that one input container was used multi-ple times).
	5	-	
	6	Overheat [1)]	The permissible temperature was exceeded. This could lead to a drop in qual-ity of the inspections. • Check ambient temperature • Check installation conditions
	7	System Error [1)]	Fatal system error. At least one reset is required, a part of the camera could be defective.

[1)] These bits remain set even if the status is no longer present. They can be reset using the control bit "Reset warning".

Structure byte 2 (Camera → PLC) Check bits to control the handshake

Subad-dress	Bit number	Name	Function description
02_{hex}	0	Handshake Acknowledge	BVS Cockpit uses this bit to signal that an incoming command was read and processed and valid result data are present.
	1	Toggle-In	BVS Cockpit uses this bit to signal that additional data can be read or written.
	7	Error	An error occurred during the handshake or the execution of a command. The Error bit is reset with a reset of the Strobe bit.

NOTE

Please note the double bit string. You have to set the same check bits in the last byte of the buffer.

Structure byte 3 (Camera → PLC) Responses to commands / status

Name	Status code	Meaning
OK	00_{hex}	Command was successfully executed.
OK Application ID	02_{hex}	Response to the command Get Application ID; transfer of the inspection program ID.
OK Results	03_{hex}	Response to the command Get Results; transfer of results.
OK Date Time	06_{hex}	Response to the command Get Date Time; transfer of timestamp.
NOK Error	10_{hex}	An error occurred during the last transfer of a command.
NOK Communication Abort	11_{hex}	Communication was canceled by the PLC by resetting the Strobe bit.
NOK Communication Error	12_{hex}	A communication error occurred (e.g. handshake timeout violation). .
NOK Invalid Command	13_{hex}	An invalid command code was transferred.
NOK Invalid Application ID	14_{hex}	At the last command an invalid inspection program ID was transmitted or the camera was not ready to switch insepction because an insepction was still running.
NOK Results	15_{hex}	Response to the command Get Results; no results present.
NOK Busy	16_{hex}	Command could not be executed since the camera is still processing another command or is not yet ready after a restart.

6.2.4 Commands and Responses

The following responses are possible for all commands:

- NOK Error: general error (e.g. command format error, error in internal communication, …)
- NOK Communication Error: communication error (e.g. handshake timeout violation, Toggle bit timeout violation …)
- NOK Communication Abort: Strobe bit was aborted prematurely (abort by PLC)

When setting an invalid command code, it will be responded with NOK Invalid Command.

Command	Responses	Meaning
Switch Application	OK	Switching insepction successful.
	NOK Invalid Application	Invalid inspection program ID requested.
	NOK Busy	Device not ready, command currently not possible.
Get Application ID	OK Application ID	Response with the currently selected inspection program ID.
	NOK Busy	Device not ready, command currently not possible.
Get Results	OK	Response with the last calculated result data.
	NOK Results	No result data available.
	NOK Busy	Device not ready, command currently not possible.
Set Inputs	OK	Setting the input data was successful.
	NOK Busy	Device not ready, command currently not possible.
Get Date Time	OK Date Time	Response with the current timestamp.
	NOK Busy	Device not ready, command currently not possible.
Set Sequence Number	OK	Setting the sequence number was successful.
	NOK Busy	Device not ready, command currently not possible.

6.2.5 Command Structure

Byte 00_{hex} of the output buffer contains the command bits and is meaningless for commands that are transferred via handshake.

NOTE

All values are represented with the little-endian setting.

The structure of different commands are explained in the following.

Command structure (PLC → Camera): Switch Application

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}	Control bits	
02_{hex}	Command code	01_{hex}: Switch Application
03_{hex}	Data length byte 0	01_{hex}: Length 1 byte
04_{hex}	Data length byte 1	00_{hex}
05_{hex}	Data byte 0	Inspection program ID
06_{hex}	Data byte 1	No meaning
…	…	…
Last byte = byte 1	Double bit string	

The inspection program ID Null is not allowed since multiple inspections can be given this ID.

Command structure (PLC → Camera): Getting the inspection program ID - Get Application ID

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}	Control bits	
02_{hex}	Command code	02_{hex}: Get Application ID
03_{hex}	Data length byte 0	00_{hex}
04_{hex}	Data length byte 1	00_{hex}
05_{hex}	Data byte 0	No meaning
…	…	…
Last byte = byte 1	Double bit string	

Command structure (PLC → Camera): Getting the results - Get Results

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}	Control bits	
02_{hex}	Command code	03_{hex}: Get Results
03_{hex}	Data length byte 0	00_{hex}
04_{hex}	Data length byte 1	00_{hex}
05_{hex}	Data byte 0	No meaning
...	…	…
Last byte = byte 1	Double bit string	

Command structure (PLC → Camera): Getting the timestamp - Get Date Time

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}	Control bits	
02_{hex}	Command code	06_{hex}: Get Date Time
03_{hex}	Data length byte 0	00_{hex}
04_{hex}	Data length byte 1	00_{hex}
05_{hex}	Data byte 0	No meaning
...	…	…
Last byte = byte 1	Double bit string	

Command structure (PLC → Camera): Setting the timestamp - Set Date Time (only Balluff SMARTCAMERA)

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}	Control bits	
02_{hex}	Command code	07_{hex}: Set Date Time
03_{hex}	Data length byte 0	07_{hex} Long timestamp
04_{hex}	Data length byte 1	00_{hex}
05_{hex}	Data byte 0	Decade
06_{hex}	Data byte 1	Years
07_{hex}	Data byte 2	Months
08_{hex}	Data byte 3	HoursDays
09_{hex}	Data byte 4	Days
$0A_{hex}$	Data byte 5	Minutes
$0B_{hex}$	Data byte 6	Seconds
...	...	…
Last byte = byte 1	Double bit string	

All elements of the timestamp are binary coded (e.g. month: 0Bhex = 11 = November). Example: August 31, 2015 12:34:56 = 14_{hex} $0F_{hex}$ 08_{hex} $1F_{hex}$ $0C_{hex}$ 22_{hex} 38_{hex}

NOTE

If an NTP server is specified in the BVS Cockpit, the time is automatically obtained. Use of the Set Date Time message is then not necessary. The manually set time is overwritten by that of the NTP server.

Command structure (PLC → Camera): Setting the sequence number - Set Sequence Number

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}	Control bits	
02_{hex}	Command code	08_{hex}: Set Sequence Number
03_{hex}	Data length byte 0	04_{hex} Length 4 bytes
04_{hex}	Data length byte 1	00_{hex}
05_{hex}	Data byte 0	Sequence number byte 0
06_{hex}	Data byte 1	Sequence number byte 1
07_{hex}	Data byte 2	Sequence number byte 2
08_{hex}	Data byte 3	Sequence number byte 3
…	…	No meaning
Last byte = byte 1	Double bit string	

Every inspection result features a sequence number which is being incremented with every new result. After sending this command, the numbering starts with the number transferred here. The sequence number is coded as UINT32.

Command structure (PLC → Camera): Setting input data - Set Inputs

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}	Control bits	
02_{hex}	Command code	04_{hex}: Set Inputs
03_{hex}	Data length byte 0	Length input data byte 0
04_{hex}	Data length byte 1	Length input data byte 1
05_{hex}	Data byte 0	Input data byte 0
…	…	…
###	Data byte n	Input data byte n
…	…	…
Last byte = byte 1	Double bit string	

Content and format of the input data are interpreted with the tool "**Receive data**".

Command structure (PLC → Camera): Getting camera information – Get Camera Info (only Balluff SMARTCAM-ERA)

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}	Control bits	
02_{hex}	Command code	05_{hex}: Get Camera Info
03_{hex}	Data length byte 0	00_{hex}
04_{hex}	Data length byte 1	00_{hex}
05_{hex}	Data byte 0	No meaning
…	…	…
Last byte = byte 1	Double bit string	

6.2.6 Structure of Responses

Bytes 00_{hex} and 01_{hex} contain warning and error bits and have no direct meaning for the handshake.

NOTE

All values are represented with the little-endian setting.

Response (Camera → PLC): Command successful - OK

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}		
02_{hex}	Status/Control bits	
03_{hex}	Command status byte	00_{hex}: OK
04_{hex}	Data length byte 0	00_{hex}
05_{hex}	Data length byte 1	00_{hex}
06_{hex}	Daten Byte 0	No meaning
…	…	…
Last byte = byte 2	Double bit string	

Response (Camera → PLC): Application ID - OK Application ID

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}		
02_{hex}	Status/Control bits	
03_{hex}	Command status byte	00_{hex}: OK Application ID
04_{hex}	Data length byte 0	$01_{hex:}$ Length 1 byte
05_{hex}	Data length byte 1	00_{hex}
06_{hex}	Data byte 0	Application ID
07_{hex}	Data byte 1	No meaning
…	…	…
Last byte = byte 2	Double bit string	

Response (Camera → PLC): Result data - OK Results

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}		
02_{hex}	Status/Control bits	
03_{hex}	Command status byte	00_{hex}: OK Results
04_{hex}	Data length byte 0	Length of the following data byte 0
05_{hex}	Data length byte 1	Length of the following data byte 1
06_{hex}	Data byte 0	Application ID
07_{hex}	Data byte 1	Result data byte 0
…	…	…
###	Data byte n	Result data byte m
…	…	…
Last byte = byte 2	Double bit string	

Content and format of the result data is configured with the tool "**Send results**". Consequently, different result data can also define formats for different insepctions. While the result data are being picked up, a maximum of one additional result data record of the next analysis is buffered in the background. If the bus communication or the used device restarts, the result data is discarded.

Response (Camera → PLC): Camera information - OK Camera Information (only Balluff SMARTCAMERA)

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}		
02_{hex}	Status/Control bits	
03_{hex}	Command code	05_{hex}: OK Camera Information
04_{hex}	Data length byte 0	$0E_{hex}$: Length 14 bytes
05_{hex}	Data length byte 1	00_{hex}
06_{hex}	Data byte 0	Camera data byte 0
...	...	...
13_{hex}	Data byte 13	Camera data byte 13
…	…	...
Last byte = byte 2	Double bit string	

The data is coded as follows:

1 byte:	Validity (0 = not valid / 1 = valid)
4 bytes:	Firmware version of camera module
	2 bytes UINT16 major version
	2 bytes UINT16 minor version
1 byte:	Camera type (0 = monochrome; 1 = color)
4 bytes:	X-resolution of camera (UINT32)
4 bytes:	Y-resolution of camera (UINT32)

Response (Camera → PLC): Timestamp - OK Date Time

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}		
02_{hex}	Status/Control bits	
03_{hex}	Command code	06_{hex}: OK Date Time
04_{hex}	Data length byte 0	07_{hex}: Length timestamp
05_{hex}	Data length byte 1	00_{hex}
06_{hex}	Data byte 0	Decades
07_{hex}	Data byte 1	Years
08_{hex}	Data byte 2	Months
09_{hex}	Data byte 3	Days
$0A_{hex}$	Data byte 4	Hours
$0B_{hex}$	Data byte 5	Minutes
$0C_{hex}$	Data byte 6	Seconds
…	…	No meaning
Last byte = byte 2	Double bit string	

All elements of the timestamp are binary coded.

Response (Camera → PLC): General error - NOK Error

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}		
02_{hex}	Status/Control bits	
03_{hex}	Command status byte	10_{hex}: NOK Error
04_{hex}	Data length byte 0	00_{hex}
05_{hex}	Data length byte 1	00_{hex}
06_{hex}	Data byte 0	No meaning
…	…	…
Last byte = byte 2	Double bit string	

The error code is coded as UINT32.

Response (Camera → PLC): Communication abort - NOK Communication Abort

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}		
02_{hex}	Status/Control bits	
03_{hex}	Command status byte	11_{hex}: NOK Communication Abort
04_{hex}	Data length byte 0	00_{hex}
05_{hex}	Data length byte 1	00_{hex}
06_{hex}	Data byte 0	No meaning
…	…	…
Last byte = byte 2	Double bit string	

Response (Camera → PLC): Communication error - NOK Communication Error

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}		
02_{hex}	Status/Control bits	
03_{hex}	Command status byte	12_{hex}: NOK Communication Error
04_{hex}	Data length byte 0	00_{hex}
05_{hex}	Data length byte 1	00_{hex}
06_{hex}	Data byte 0	No meaning
…	…	…
Last byte = byte 2	Double bit string	

Response (Camera → PLC): Invalid command code - NOK Invalid Command

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}		
02_{hex}	Status/Control bits	
03_{hex}	Command status byte	13_{hex}: NOK Invalid Command
04_{hex}	Data length byte 0	00_{hex}
05_{hex}	Data length byte 1	00_{hex}
06_{hex}	Data byte 0	No meaning
…	…	…
Last byte = byte 2	Double bit string	

Response (Camera → PLC): Invalid application ID - NOK Invalid Application ID

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}		
02_{hex}	Status/Control bits	
03_{hex}	Command status byte	14_{hex}: NOK Invalid Application ID
04_{hex}	Data length byte 0	00_{hex}
05_{hex}	Data length byte 1	00_{hex}
06_{hex}	Data byte 0	No meaning
…	…	…
Last byte = byte 2	Double bit string	

Response (Camera → PLC): No results - NOK Results

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}		
02_{hex}	Status/Control bits	
03_{hex}	Command status byte	15_{hex}: NOK Results
04_{hex}	Data length byte 0	00_{hex}
05_{hex}	Data length byte 1	00_{hex}
06_{hex}	Data byte 0	No meaning
…	…	…
Last byte = byte 2	Double bit string	

Response (Camera → PLC): Not ready - NOK Busy

Subaddress	Meaning	Content / Function description
00_{hex}		
01_{hex}		
02_{hex}	Status/Control bits	
03_{hex}	Command status byte	16_{hex}: NOK Busy
04_{hex}	Data length byte 0	00_{hex}
05_{hex}	Data length byte 1	00_{hex}
06_{hex}	Data byte 0	No meaning
…	…	…
Last byte = byte 2	Double bit string	

6.2.7 Error Codes

Code		Description	Handling recommendation
00000000_{hex}	NONE	No error occurred.	None
00000005_{hex}	COM-MAND_BUFFER_ OVERFLOW	Overflow internal command buffer.	Internal overload error. Try command again.
00000006_{hex}	COMMAND_EN-CODE_ ERROR	Error coding output data	Check the structure of commands.
00000007_{hex}	COM-MAND_TIMEOUT	Timeout violation while execut-ing a command.	Have program run faster or disable timeout in BVS Cockpit.
00000009_{hex}	FORMAT_ ERROR	Invalid format of data for a com-mand. (incl. wrong data length).	Check the structure of commands.
$0000000A_{hex}$	INCON-SISTENT_DATA_ TIMEOUT	After beginning of a handshake both bit strings were different for too long.	Check consistency of both bit strings. You may have forgotten to set the second bit string.
$0000000B_{hex}$	COMMAND_EXE-CUTE_ FAILED	Executing a command failed (e.g. no response from camera module).	Serious error. Restart required.
FFFFFFFF-$_{hex}$	UNKNOWN_ER-ROR	An unknown error occurred.	None

6.2.8 Function Module

To simplify the communication, Balluff GmbH provides following function blocks:

- For Siemens controllers there is a PROFINET function block.
- For Rockwell controllers there is an EtherNet/IP function block.

Both can be downloaded from the Balluff website (www.balluff.com).

6.2.9 Examples of Fieldbus Communication

- 16 bytes output buffer; 16 bytes input buffer
- Representation for the setting “little-endian” (lowest value byte first)

Notation	Meaning	Logic combination
|$\#\#_{hex}$	Set bits	Value = Value OR $\#\#_{hex}$
~$\#\#_{hex}$	Reset bits	Value = Value AND (NOT($\#\#_{hex}$))
^$\#\#_{hex}$	Invert bits	Value = Value XOR $\#\#_{hex}$

Switch application (application 51 = 33hex)

PLC				BVS Cockpit / Camera		
1. Process output buffer (observe sequence):				2. Process input buffer (observe sequence):		
02_{hex}	Command identifier:	01_{hex}	→	03_{hex}	Command status: (command executed successfully)	00_{hex}
03_{hex}	Data length byte 0:	01_{hex}		04_{hex}	Data length byte 0:	00_{hex}
04_{hex}	Data length byte 1:	00_{hex}		05_{hex}	Data length byte 1:	00_{hex}
05_{hex}	Application ID:	33_{hex}		02_{hex} / $0F_{hex}$	Set Acknowledge bit	|01_{hex}
01_{hex} / $0F_{hex}$	Set Strobe bit	|01_{hex}				
			←			
3. Process output buffer:				4. Process input buffer:		
01_{hex} / $0F_{hex}$	Reset Strobe bit	~01_{hex}	→	02_{hex} / $0F_{hex}$	Reset Acknowledge bit	~01_{hex}

Switch application, problem: invalid application ID (FFhex)

PLC				BVS Cockpit / Camera		
1. Process output buffer (observe sequence):				2. Process input buffer (observe sequence):		
02_{hex}	Command identifier:	01_{hex}	→	03_{hex}	Command status: (NOK Invalid Application ID)	14_{hex}
03_{hex}	Data length byte 0:	01_{hex}		04_{hex}	Data length byte 0:	00_{hex}
04_{hex}	Data length byte 1:	00_{hex}		05_{hex}	Data length byte 1:	00_{hex}
05_{hex}	Application ID:	FF_{hex}		02_{hex} / $0F_{hex}$	Set Acknowledge bit	|81_{hex}
01_{hex} / $0F_{hex}$	Set Strobe bit	|01_{hex}				
			←			
3. Process output buffer:				4. Process input buffer:		
01_{hex} / $0F_{hex}$	Reset Strobe bit	~01_{hex}	→	02_{hex} / $0F_{hex}$	Reset Acknowledge bit and error bit	~81_{hex}

Switch application, problem: camera not yet ready

PLC				BVS Cockpit / Camera		
1. Process output buffer (observe sequence):				2. Process input buffer (observe sequence):		
02_{hex}	Command identifier:	01_{hex}	→	03_{hex}	Command status: (NOK Busy)	16_{hex}
03_{hex}	Data length byte 0:	01_{hex}		04_{hex}	Data length byte 0:	00_{hex}
04_{hex}	Data length byte 1:	00_{hex}		05_{hex}	Data length byte 1:	00_{hex}
05_{hex}	Application ID:	33_{hex}		02_{hex} / $0F_{hex}$	Set Acknowledge bit	|81_{hex}
01_{hex} / $0F_{hex}$	Set Strobe bit	|01_{hex}				
			←			
3. Process output buffer:				4. Process input buffer:		
01_{hex} / $0F_{hex}$	Reset Strobe bit	~01_{hex}	→	02_{hex} / $0F_{hex}$	Reset Acknowledge bit and error bit	~81_{hex}

Switch application, problem: internal camera problem

PLC				BVS Cockpit / Camera		
1. Process output buffer (observe sequence):				2. Process input buffer (observe sequence):		
02_{hex}	Command identifier:	01_{hex}	→	03_{hex}	Command status: (NOK Error)	10_{hex}
03_{hex}	Data length byte 0:	01_{hex}		04_{hex}	Data length byte 0:	00_{hex}
04_{hex}	Data length byte 1:	00_{hex}		05_{hex}	Data length byte 1:	00_{hex}
05_{hex}	Application ID:	33_{hex}		06_{hex}	Data byte 0: Error code byte 0	11_{hex}
01_{hex} / $0F_{hex}$	Set Strobe bit	|01_{hex}		07_{hex}	Data byte 1: Error code byte 1	00_{hex}
				08_{hex}	Data byte 2: Error code byte 2	00_{hex}
				09_{hex}	Data byte 3: Error code byte 3	00_{hex}
				02_{hex} / $0F_{hex}$	Set Acknowledge bit and error bit	|81_{hex}
			←			
3. Process output buffer:				4. Process input buffer:		
01_{hex} / $0F_{hex}$	Reset Strobe bit	~01_{hex}	→	02_{hex} / $0F_{hex}$	Reset Acknowledge bit and error bit	~81_{hex}

Get result container (without toggling) (4 byte result container)

PLC				BVS Cockpit / Camera		
1. Process output buffer (observe sequence):				2. Process input buffer (observe sequence):		
02_{hex}	Command identifier:	03_{hex}	→	03_{hex}	Command status: (OK Results)	03_{hex}
03_{hex}	Data length byte 0:	00_{hex}		04_{hex}	Data length byte 0:	05_{hex}
04_{hex}	Data length byte 1:	00_{hex}		05_{hex}	Data length byte 1:	00_{hex}
01_{hex} / $0F_{hex}$	Set Strobe bit	|01_{hex}		06_{hex}	Application ID	
				07_{hex}	Data byte 0: Result byte 0	
				08_{hex}	Data byte 1: Result byte 1	
				09_{hex}	Data byte 2: Result byte 2	
				$0A_{hex}$	Data byte 3: Result byte 3	
				02_{hex} / $0F_{hex}$	Set Acknowledge bit	|01_{hex}
			←			
3. Read output buffer:						
07_{hex} ... $0A_{hex}$	Copy result bytes					
Process output buffer:				4. Process input buffer:		
01_{hex} / $0F_{hex}$	Reset Strobe bit	~01_{hex}	→	02_{hex} / $0F_{hex}$	Reset Acknowledge bit	~01_{hex}

Get result container (with toggling) (20 byte result container)

PLC				BVS Cockpit / Camera		
1. Process output buffer (observe sequence):				2. Process input buffer (observe sequence):		
02_{hex}	Command identifier:	03_{hex}	→	03_{hex}	Command status: (OK Results)	03_{hex}
03_{hex}	Data length byte 0:	00_{hex}		04_{hex}	Data length byte 0:	15_{hex}
04_{hex}	Data length byte 1:	00_{hex}		05_{hex}	Data length byte 1:	00_{hex}
01_{hex} / $0F_{hex}$	Set Strobe bit	|01_{hex}		06_{hex}	Application ID	
				07_{hex}	Data byte 0: Result byte 0	
				...	...	
				$0D_{hex}$	Result byte 6	
				$0E_{hex}$	Result byte 7	
				02_{hex} / $0F_{hex}$	Set Acknowledge bit	|01_{hex}
			←			
3. Read input buffer:						
06_{hex} ... $0E_{hex}$	Copy first 8 result bytes					
Process output buffer:				4. Process input buffer:		
01_{hex} / $0F_{hex}$	Invert Toggle-Out	~02_{hex}	→	06_{hex}	Result byte 8	
				07_{hex}	Result byte 8	
				...	...	
				$0D_{hex}$	Result byte 15	
				$0E_{hex}$	Result byte 16	
				02_{hex} / $0F_{hex}$	Invert Toggle-In	^02_{hex}
			←			
5. Read input buffer:						
06_{hex} ... $0E_{hex}$	Copy second 9 result bytes					
Process output buffer:				6. Process input buffer:		
01_{hex} / $0F_{hex}$	Invert Toggle-Out	~02_{hex}	→	06_{hex}	Result byte 17	
				07_{hex}	Result byte 18	
				08_{hex}	Result byte 19	
				02_{hex} / $0F_{hex}$	Invert Toggle-In	^02_{hex}
			←			
7. Read input buffer:						
06_{hex} ... $0E_{hex}$	Copy last three result bytes					
Process output buffer:				8. Process input buffer:		
01_{hex} / $0F_{hex}$	Reset Strobe bit	~01_{hex}		02_{hex} / $0F_{hex}$	Reset Acknowledge bit	~01_{hex}

Result data:

Set input data (with toggling) (22 byte input data)

PLC				BVS Cockpit / Camera		
1. Process output buffer (observe sequence):				2. Read output buffer:		
02_{hex}	Command identifier:	04_{hex}	→	06_{hex} / $0E_{hex}$	Copy first 10 input bytes	
03_{hex}	Data length byte 0:	16_{hex}				
04_{hex}	Data length byte 1:	00_{hex}				
05_{hex}	Input byte 0					
06_{hex}	Input byte 1					
...	...					
$0D_{hex}$	Input byte 8					
$0E_{hex}$	Input byte 9					
01_{hex} / $0F_{hex}$	Set Strobe bit	|01_{hex}				
				Process input buffer (observe sequence):		
			←	02_{hex} / $0F_{hex}$	Invert Toggle-In	^02_{hex}
3. Process output buffer:				4. Read output buffer:		
06_{hex}	Input byte 10		→	06_{hex} / $0E_{hex}$	Copy second 10 input bytes	
07_{hex}	Input byte 11					
...	...					
$0D_{hex}$	Input byte 18					
$0E_{hex}$	Input byte 19					
01_{hex} / $0F_{hex}$	Invert Toggle-Out	~02_{hex}				
				Process input buffer (observe sequence):		
			←	02_{hex} / $0F_{hex}$	Invert Toggle-In	^02_{hex}
5. Process output buffer:				6. Read output buffer:		
05_{hex}	Input byte 20		→	06_{hex} / $0E_{hex}$	Copy last two input bytes	
06_{hex}	Input byte 21					
01_{hex} / $0F_{hex}$	Invert Toggle-Out	~02_{hex}				
				Process input buffer (observe sequence):		
				03_{hex}	Command status: OK	00_{hex}
				04_{hex}	Data length byte 0:	00_{hex}
				05_{hex}	Data length byte 1:	00_{hex}
				02_{hex} / $0F_{hex}$	Set Acknowledge bit	|01_{hex}
			←			
7. Process output buffer:				8. Process input buffer		
01_{hex} / $0F_{hex}$	Reset Strobe bit	~01_{hex}	→	02_{hex} / $0F_{hex}$	Reset Acknowledge bit	~01_{hex}

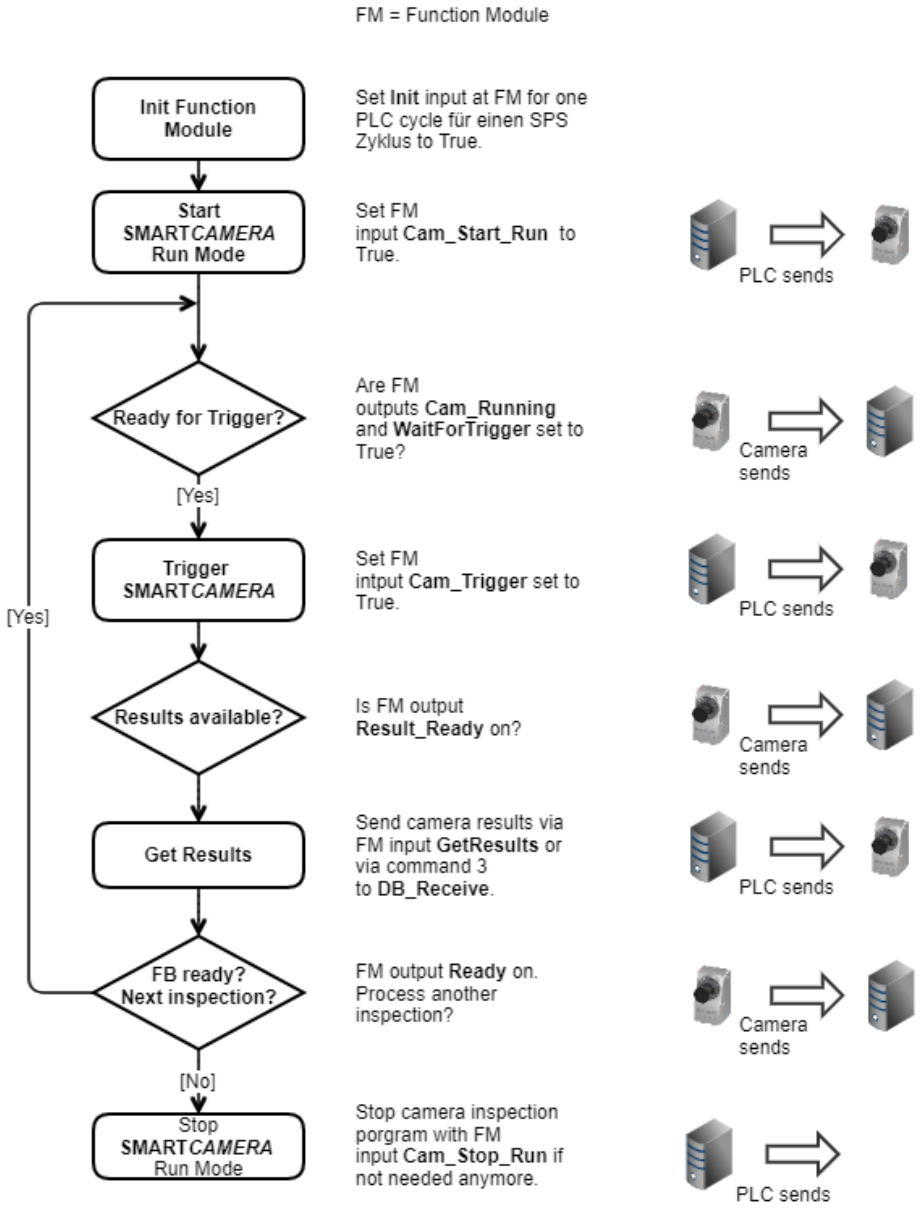
Result data:

Switch application, problem: PLC sets invalid length

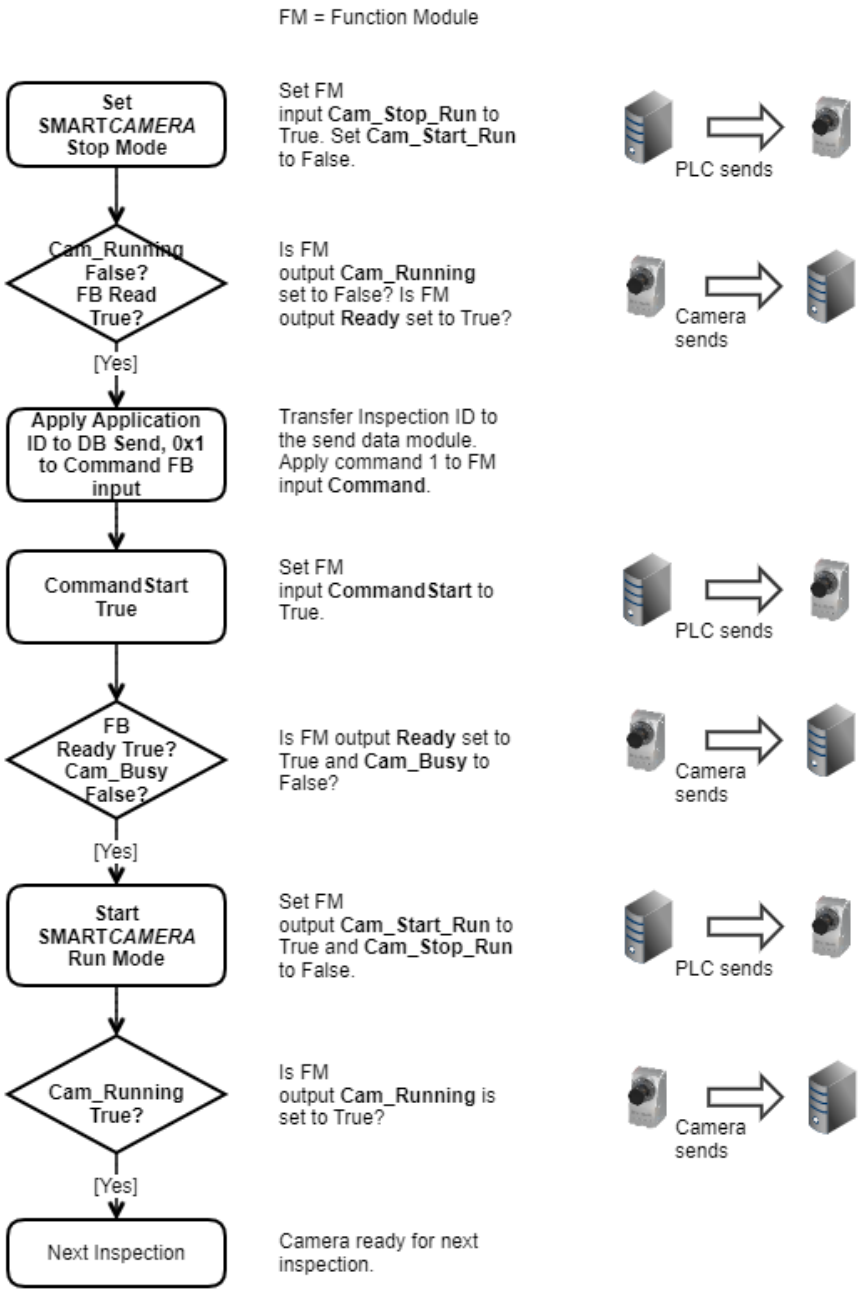
PLC		**BVS Cockpit / Camera**
1. Process output buffer (observe sequence):		2. Process input buffer (observe sequence):
02_{hex} | Command identifier: | 01_{hex} 03_{hex} | Data length byte 0: | FF_{hex} 04_{hex} | Data length byte 1: | 00_{hex} 05_{hex} | Application ID: | 33_{hex} 01_{hex} / $0F_{hex}$ | Set Strobe bit | $\|01_{hex}$ Timeout while waiting for Acknowledge bit ⇒ Cancel handshake	→	02_{hex} / $0F_{hex}$ | Invert Toggle-In | $\^{}02_{hex}$ Timeout while waiting on inverted Toggle-Out ⇒ Cancel handshake
↓		↓
		3. Process input buffer:
		03_{hex} | | 30_{hex} 04_{hex} | | 00_{hex} 05_{hex} | | 00_{hex} 02_{hex} / $0F_{hex}$ | Set error bit | $\|80_{hex}$
4. Process output buffer:		
01_{hex} / $0F_{hex}$ | Reset Strobe bit | $\sim 01_{hex}$		
	→	5. Process input buffer
		02_{hex} / $0F_{hex}$ | Reset error bit | $\sim 80_{hex}$

- PLC sets wrong data length (FF = 255), but wants to transfer only one application ID byte.
- After the start of the handshake additional data is expected and toggle process is started.
- While waiting for the next partial data, a timeout occurs → the process is canceled the process and the command status NOK Communication Error is written to the input buffer and the error bit is set.
- While waiting for the end of the handshake (waiting for Acknowledge bit), the PLC runs into a timeout → the PLC also cancels the process and resets the Strobe bit.
- By reading out the error bit and the value NOK Communication Error in the command status, the PLC recognizes a canceled handshake.
- After canceling the handshake on both sides, both sides are back in the initial state and ready for new commands.

Flow chart of a sample inspection with camera function module BVS_SC



Flow chart of insepction switch with camera function module BVS_SC



6.3 Communication via RS232 (only SMARTCAMERA Lite)

NOTE

To communicate via serial interface, you have to set the *Mode* in the system menu ("*System settings -> communication*") to "**Serial**".

With a suitable cable and a terminal program like *PuTTY* for Windows or *CuteCom* for Linux you can establish a serial connection to the smart camera. Following communication parameters are necessary:

Parameters	Windows	Linux
Interface	COM1	/dev/ttyS0
Speed (Baud)	57600	
Data bits	8	
Stop bits	1	
Handshake	None	
Flow control	None	

6.3.1 Message structure

The message structure is similar to the structure used in communication via UDP. Thus all messages have the structure:

Length in bytes	Structure	Description
2	UINT16	Message ID
2	UINT16	User data length in bytes
	UINT8[]	User data

Just as with UDP, messages are transferred using the little-endian format. This means the lowest value byte is transferred first.

6.3.2 Messages

All messages are equal to the communication via UDP and can be found in the correspondent chapter. *Connect* and *Disconnect* messages are not necessary for communication via serial interface.

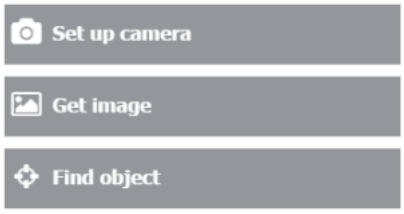
7 Create an initial inspection program

7.1 Specify position and orientation

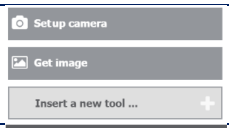
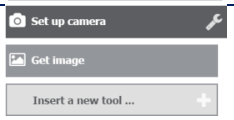
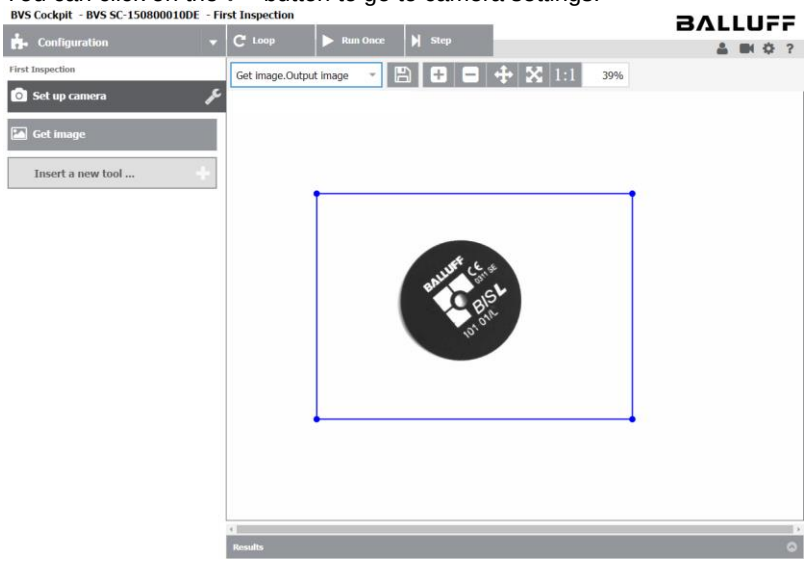
Specifying the object position in the image and its orientation is a basic operation that is often also used as the first step in inspecting an object.

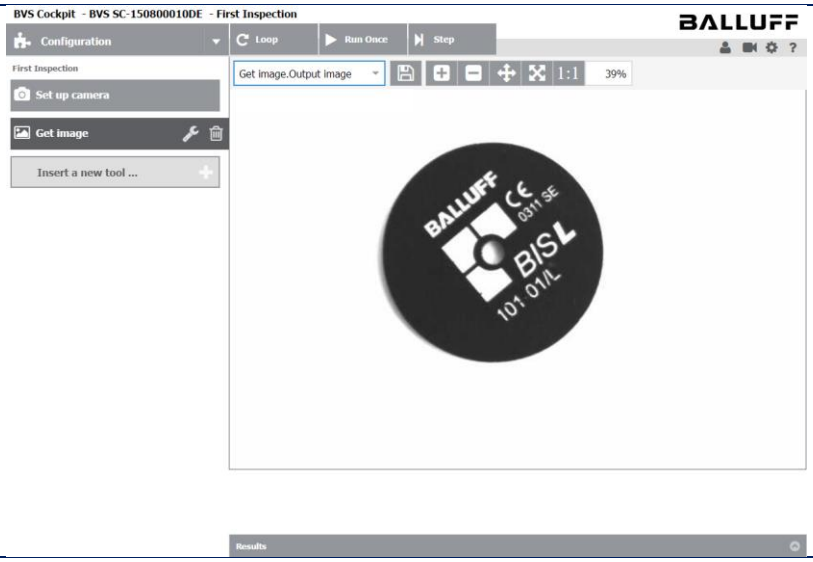
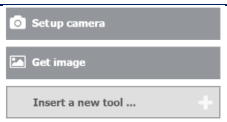
This example is concerned with establishing whether and where an expected object exists in the image.

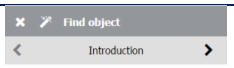
The complete inspection program looks like this:



7.1.1 Step-by-step

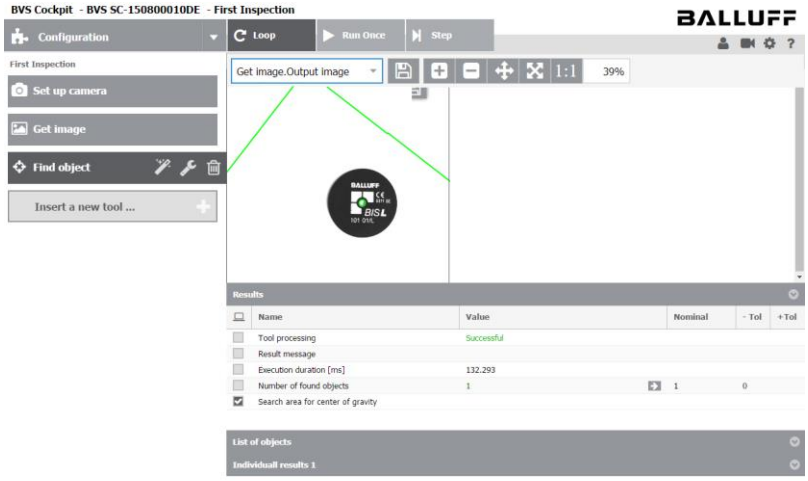
	**(0) Initial situation** When creating a new inspection program, the first two program steps appear automatically: "**Set up camera**" and "**Get image**".
	**(1) Set up camera** By selecting the "**Set up camera**" step, a live image is displayed. This makes it possible to align the mechanical orientation of the camera, such that the object to be inspected can be seen. It also makes it easier to set the focus- and aperture ring of the lens correctly. You can click on the button to go to camera settings.  The image section is selected, such that the object, in this case the RFID tag, is always within the image. The smaller the image section, the quicker the subsequent inspection steps are completed.
	**(2) Get image** The next inspection step acquires an image with the selected parameters and with the image size specified when setting up the camera.

<table>
<tr><td></td><td> 
</td></tr>
<tr><td> 
</td><td>(3) Insert tool
1. Click on the ✚ button.
The "Tool Box" opens.
2. Select the "Find object" 

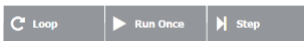
tool and
3. insert it by clicking on the ✚ button.
Parameters that require a link (e.g. the input image) are automatically linked on insertion.</td></tr>
<tr><td> 
</td><td>(4) "Find object" training
When inserting the tool, a wizard opens automatically, providing examples that enable you to set up the inspection program step. The objects are also displayed in the various positions and orientations and example acquisitions are created. The tool uses the examples displayed to learn the possible variances in position and maximum rotations. Finally, the wizard creates a parameter record for this inspection program step, which, if required, can also be numerically modified by using the settings 🔧 .
The benefit of learning via examples is that variable elements of the image (e.g. shadow) can be recognized and excluded from the model. This results in stable recognition in various object positions and rotations.</td></tr>
</table>

Testing the inspection program

You can test the stability and speed of an inspection step by selecting an inspection step and using other example images.



On the right of the configuration page, the image and the results of the inspection program step are shown in the form of a table.

Sequence control 

supports you when testing:

- **Loop**: the program captures images continuously and runs through to the selected inspection program step.
- **Single run**: captures a single new image. In this case, the program also runs through to the selected inspection program step.
- **Continue**: serves to run through all inspection program steps consecutively, one-by-one. A new image is captured at the inspection program step "**Get image**".

Results: the values in the results table for the inspection program step "**Find object**" are i. a.:

- Tool processing: Successful, if the object was found. Error, if it was not found. The inspection program aborts the current run in the event of any error and, depending on the mode, starts another iteration.
- Result message: Text description of the error with detailed information about the reason for the error.
- Displacement in X direction [px]: horizontal movement of the image with respect to the target position.
- Displacement in Y direction [px]: vertical movement of the image with respect to the target position.
- Rotation [°]: rotation of the image with respect to the target position.
- Output Image: image orientation and rotation are correct.

Other examples can be found in the "**Use Cases**" section.

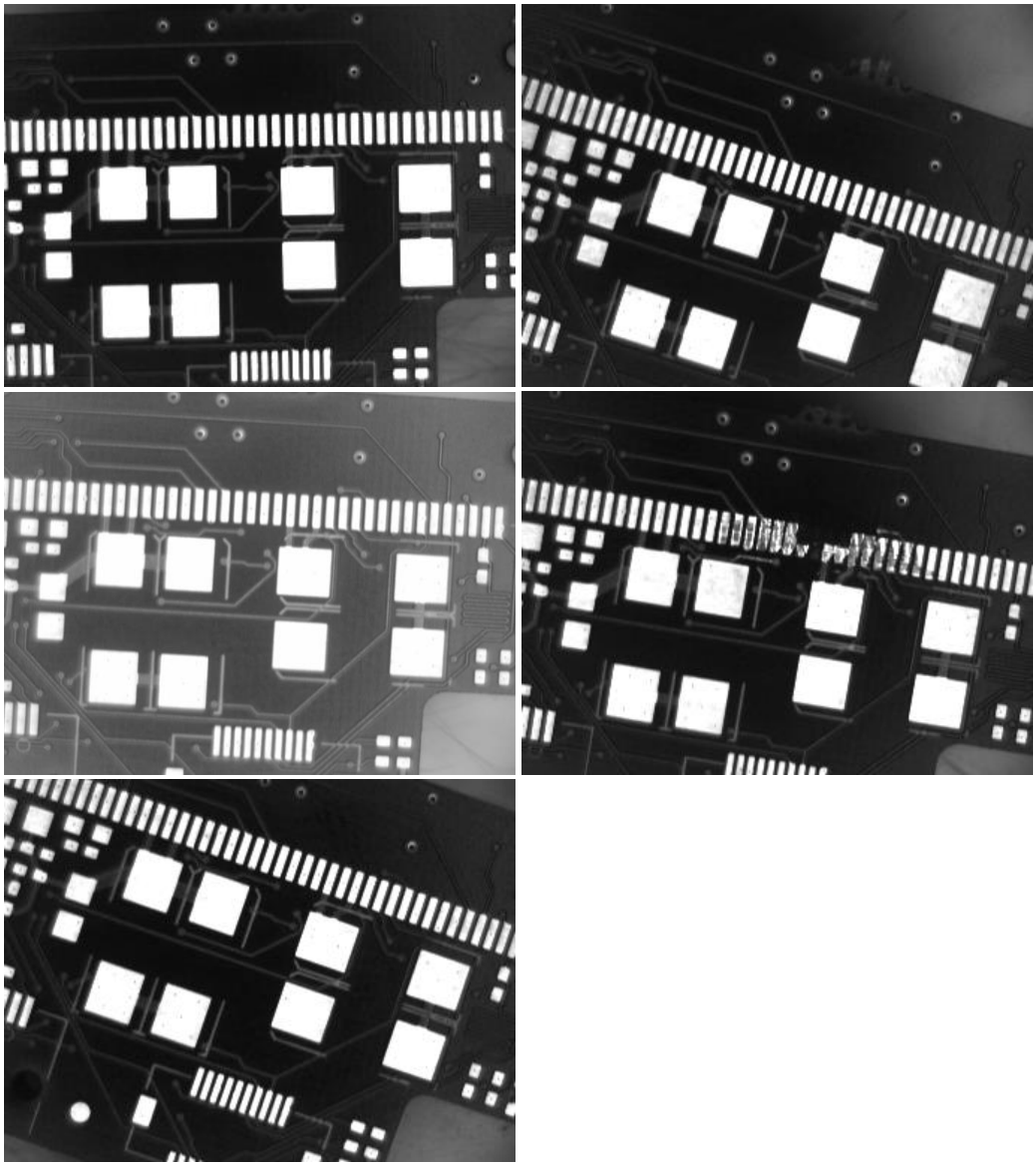
8 Use Cases

8.1 Quality control

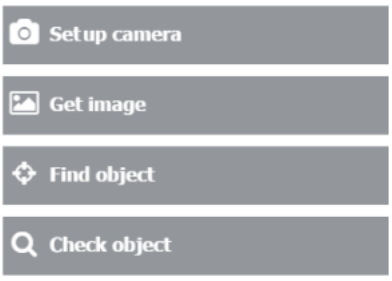
8.1.1 Check printed circuit boards

Checking an object for errors in its contours is a very frequent task in production. Unfortunately, there are mostly only good elements available and errors are difficult to describe.

Example images



The complete inspection program looks like this:



Step-by-step

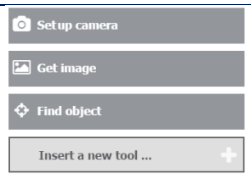
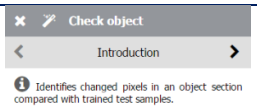
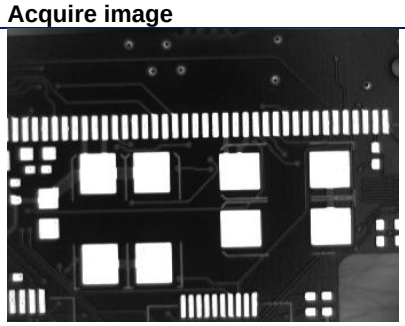
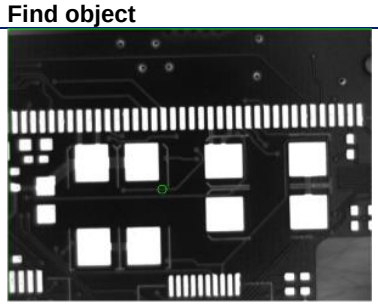
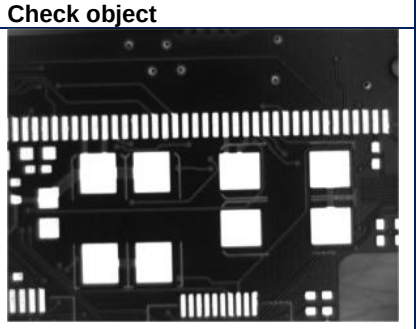
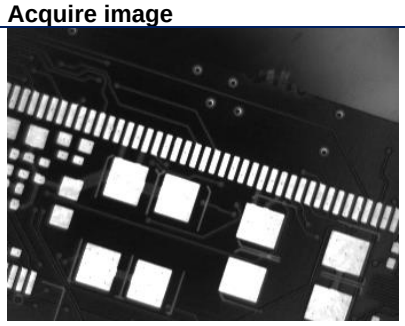
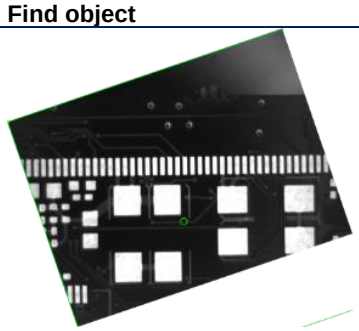
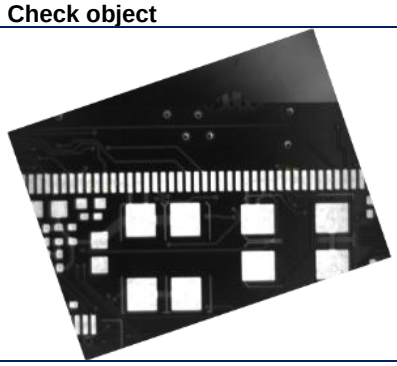
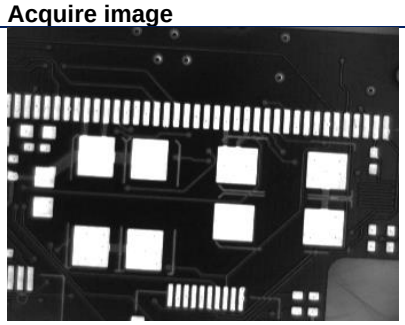
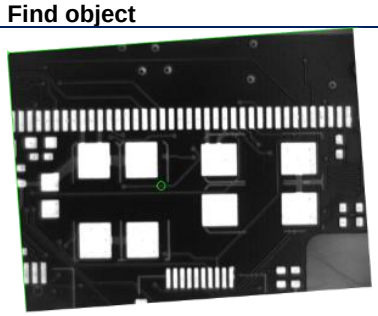
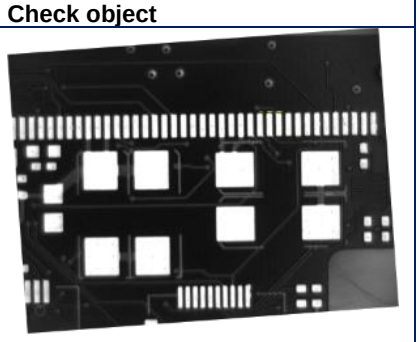
	**(0) Initial situation** The camera is set up according to the example "**Specify position and orientation**" and the object is learned, such that it is reliably found and orientated.
	**(1) Insert tool** 1. Click on the ✚ button. The "**Tool Box**" opens. 2. Select the "**Check object**" Check object ✚ tool and 3. insert it by clicking on the ✚ button. This will automatically take you to the tool's wizard.
	**(2) "Check object" training** To train the tool, a sequence of consecutive good elements is displayed. The tool learns the permitted variations based on the differences between the individual elements. As far as possible, you should also use extreme examples here.

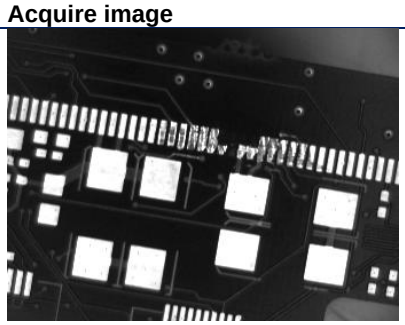
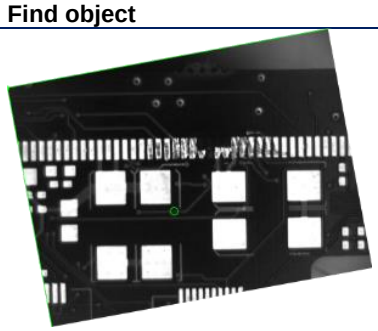
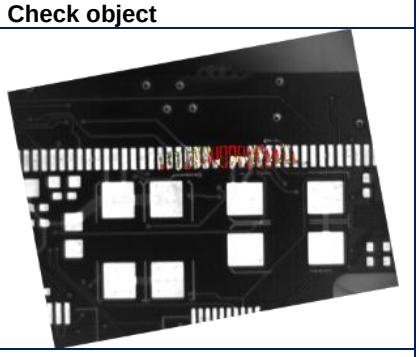
Image examples

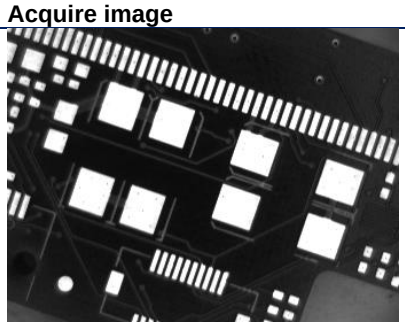
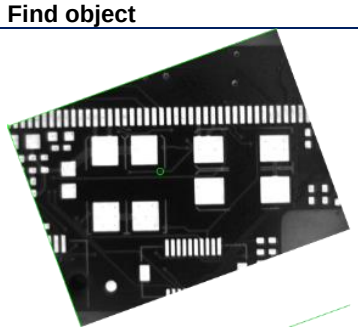
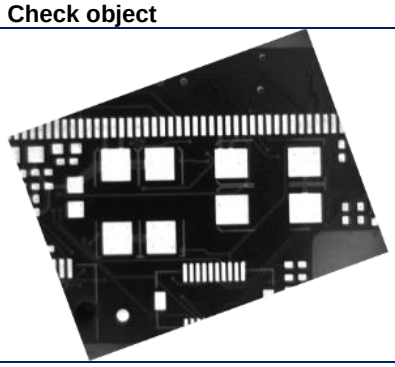
The explanation below uses image examples to show how different input images are processed further by the "**Find object**" and "**Check object**" tools.

Acquire image	Find object	Check object
		
Comment		
The object was correctly rotated. The "**Check object**" tool found no errors.		

Acquire image	Find object	Check object
		
Comment		
The object was correctly rotated. The "**Check object**" tool found no errors.		

Acquire image	Find object	Check object
		
Comment		
The object was correctly rotated. The "**Check object**" tool found no errors.		

Acquire image	Find object	Check object
		
Comment		
The object was correctly rotated. The "**Check object**" tool has **recognized an error and highlighted it in the image**.		

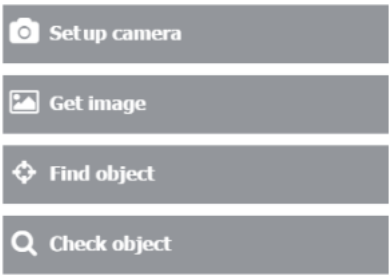
Acquire image	Find object	Check object
		
Comment		
The object was correctly rotated. The "**Check object**" tool found no errors.		

8.1.2 Check completeness

In production, completeness checks are part of everyday life. Small details are often missed.

Example images

The complete inspection program looks like this:



Step-by-step

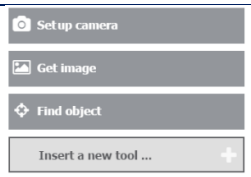
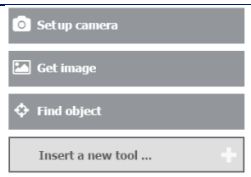
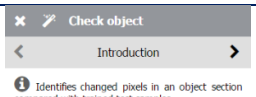
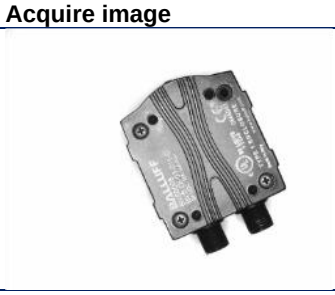
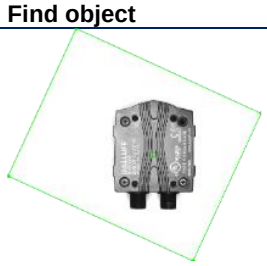
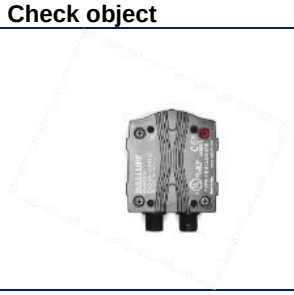
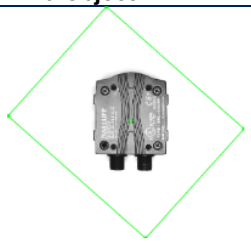
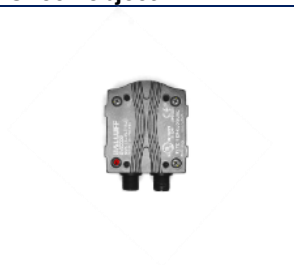
	**(0) Initial situation** The camera is set up according to the example "**Specify position and orientation**" and the object is learned, such that it is reliably found and orientated. **NOTE** The accelerate the inspection program, you can reduce the area of interest.
	**(1) Insert tool** 1. Click on the ✚ button. The "**Tool Box**" opens. 2. Select the "**Find object**"  tool and 3. insert it by clicking on the ✚ button. This will automatically take you to the tool's wizard.
	**(2) "Check object" training** To train the tool, a sequence of consecutive good elements is displayed. The tool learns the permitted variations based on the differences between the individual elements. As far as possible, you should also use extreme examples here.

Image examples

The explanation below uses image examples to show how different input images are processed further by the "**Find object**" and "**Check object**" tools.

Acquire image	Find object	Check object
		
Comment		
The object was correctly rotated. The "**Check object**" tool found an error.		

Acquire image	Find object	Check object
		
Comment		
The object was correctly rotated. The "**Check object**" tool found an error.		

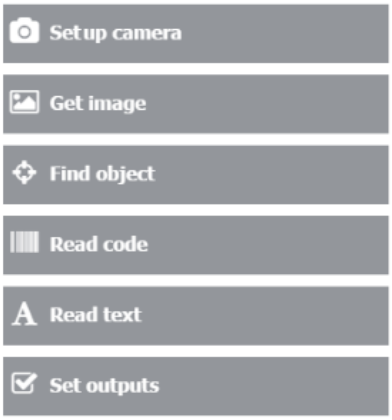
8.2 Identification

8.2.1 Compare bar code with text on label

Identifying bar codes and texts is a fundamental task to trace a production process.

Example images

The complete inspection program looks like this:

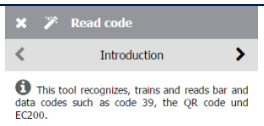


Step-by-step

(0) Initial situation
The camera is set up according to the example "**Specify position and orientation**" and the object is learned, such that it is reliably found and orientated.

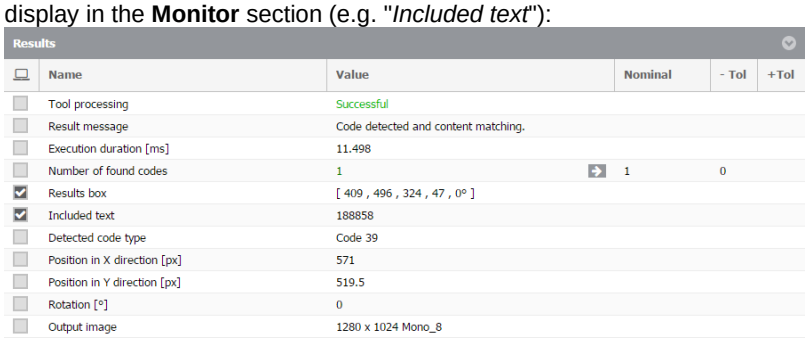
(1) Insert tool

1. Click on the ✚ button.
   The "**Tool Box**" opens.
2. Select the "**Read code**"  tool and
3. insert it by clicking on the ✚ button.
   This will automatically take you to the tool's wizard.



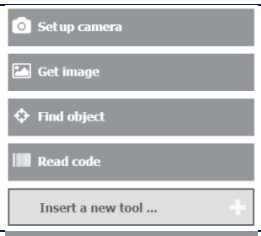
(2) "Read code" set up
Let the wizard find the code. By limiting the search area with the blue box, the tool will recognize the code faster. The blue box will be visible as soon as your mouse is located over the live image:

After finishing the wizard, you can select the results in the results table that you want to display in the **Monitor** section (e.g. "*Included text*"):



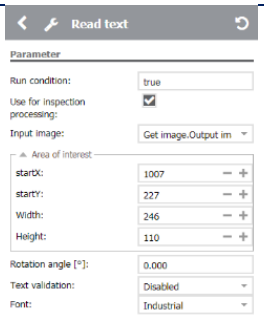
Results

	Name	Value	Nominal	- Tol	+Tol
☐	Tool processing	Successful			
☐	Result message	Code detected and content matching.			
☐	Execution duration [ms]	11.498			
☐	Number of found codes	1	1	0	
☑	Results box	[409 , 496 , 324 , 47 , 0°]			
☑	Included text	188858			
☐	Detected code type	Code 39			
☐	Position in X direction [px]	571			
☐	Position in Y direction [px]	519.5			
☐	Rotation [°]	0			
☐	Output image	1280 x 1024 Mono_8			

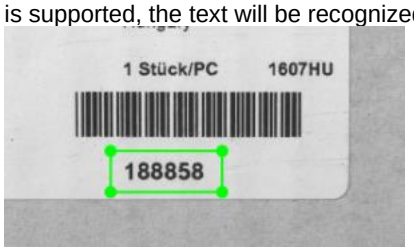


(3) Insert tool

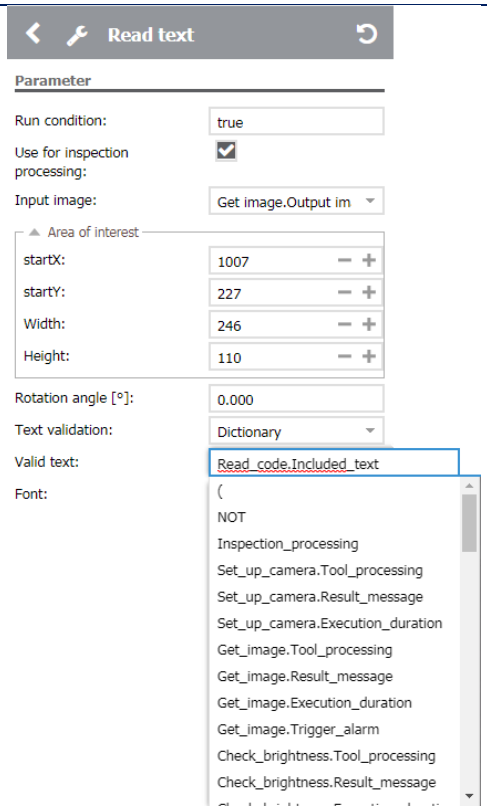
1. Click on the ✚ button.
   The "**Tool Box**" opens.
2. Select the "**Read text**"  tool and
3. insert it by clicking on the ✚ button.

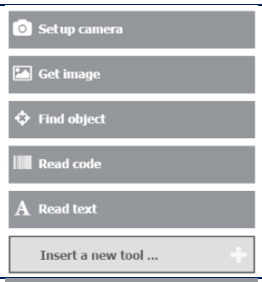


(4) "Read text" set up
Define the search area with the blue box in which the text should be searched. If the font is supported, the text will be recognized automatically.



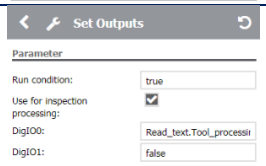
Using the parameter "*Text validation*", you can validate the text. In this example, the text is compared with the bar code text. Choose "**Lexicon**" and as its content the result of the "**Read code**".
Click into the text field and remove all of its contents. A list will appear containing all results of predecessor tools from which you choose "**Read code.Included text**".





(5) Insert tool

1. Click on the ✚ button.
   The "**Tool box**" opens.
2. Select the "**Set outputs**"  tool an
3. insert it by clicking on the ✚ button.



(6) "Set outputs" set up

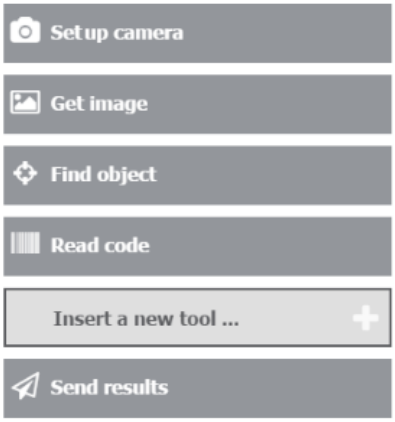
Select the output you want to use for the result of the "**Read text**" tool. Click into the regarding text field, remove its contents and choose "**Read text.Tool processing**" from the appearing list. As soon as the read bar code matches the read text, the Output0 will be active.

8.2.2 Read bar code and send result

Identifying bar codes and texts is a fundamental task to trace a production process.

Example images

The complete inspection program looks like this:

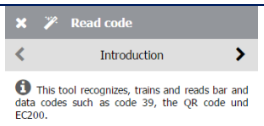


Step-by-step

(0) Initial situation
The camera is set up according to the example "**Specify position and orientation**" and the object is learned, such that it is reliably found and orientated.

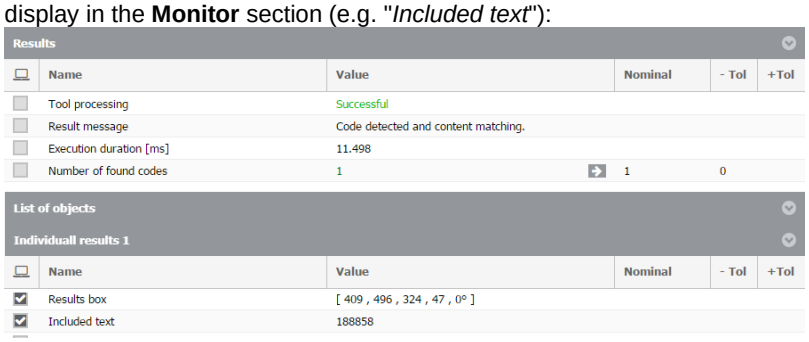
(1) Insert tool

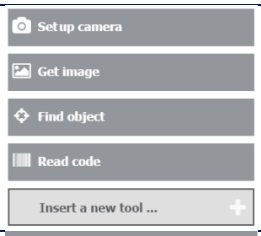
1. Click on the ✚ button.
   The "**Tool Box**" opens.
2. Select the "**Read code**"  tool and
3. insert it by clicking on the ✚ button.
   This will automatically take you to the tool's wizard.



(2) "Read code" set up
Let the wizard find the code. By limiting the search area with the blue box, the tool will recognize the code faster. The blue box will be visible as soon as your mouse is located over the live image:

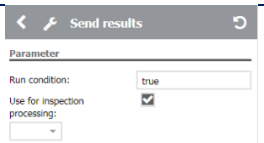
After finishing the wizard, you can select the results in the results table that you want to display in the **Monitor** section (e.g. "*Included text*"):



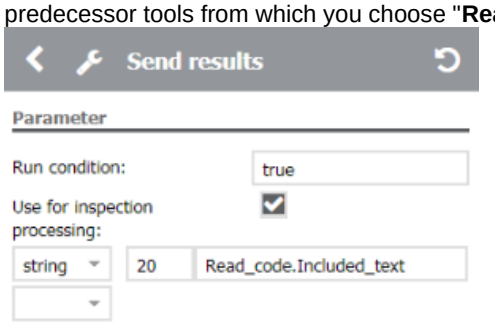


(3) Insert tool

1. Click on the ✚ button.
   The "**Tool Box**" opens.
2. Select the "**Send results**"  tool and
3. insert it by clicking on the ✚ button.



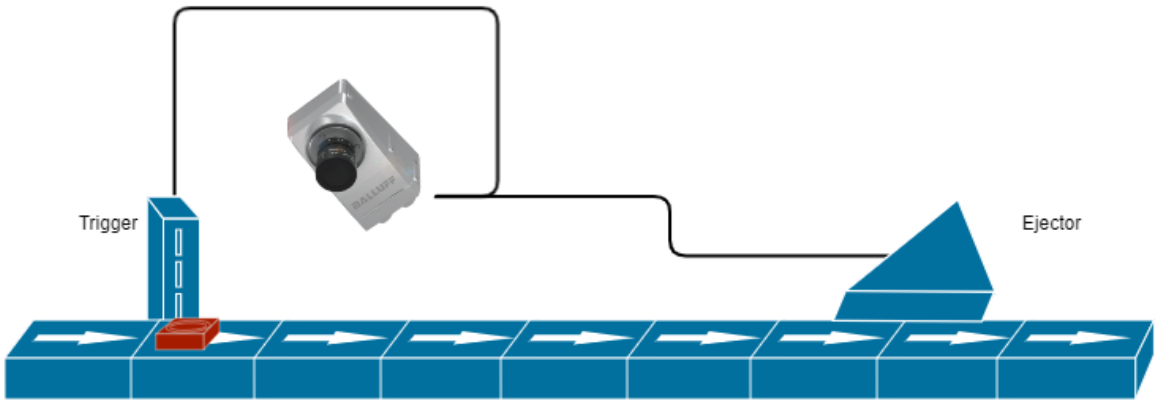
(4) "Send results" set up
Specify the value you want to send. Choose "**string**" as the parameter type and click in the input field next to the length parameter field. A list will appear containing all results of predecessor tools from which you choose "**Read_code.Included_text**".



8.3 Process control

8.3.1 Hardware-based trigger and ejection using pulse and delay

This use case assumes that the connected camera supports hardware-based triggers, pulses, and delays. Is this the case (e.g. with **SMART***CAMERA*) then it is possible to realize reliable inspections in applications with conveyor belts and ejectors. In most cases, the installation will look like as follows:

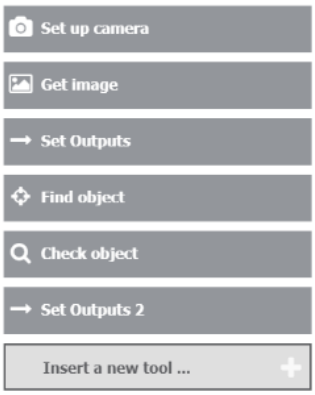


In detail,

1. a hardware trigger triggers the acquisition, afterwards
2. the camera inspects the object, and finally
3. the camera controls the ejector if necessary.

Normally, the processing time of all inspections steps is sufficient to control the ejector in time. If there is the possibility that this is not always the case, you can define an initial state using the "**Set output**" tool at the beginning of the inspection, which indicates, whether in "**Set output 2**" all not checked objects are rejected or not in.

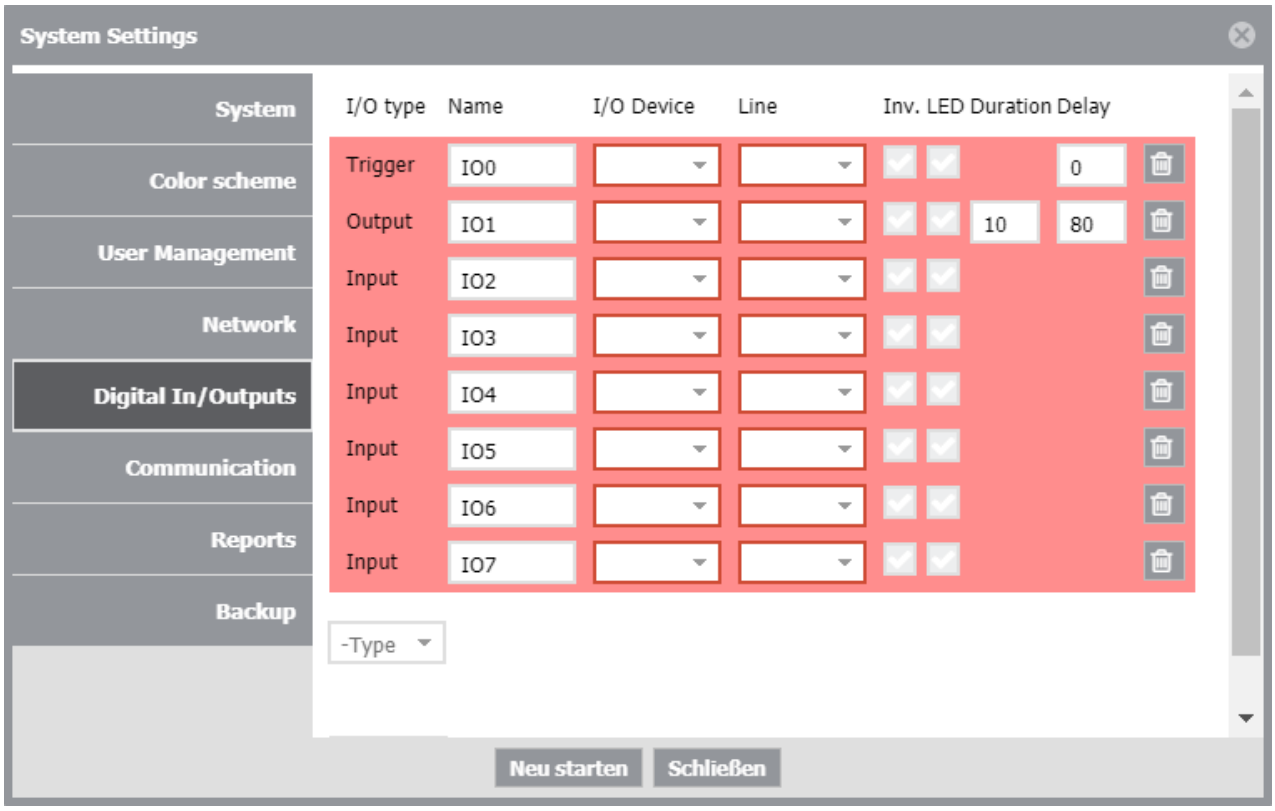
With the following program we will go through the different scenarios:



Following environment variables are given:

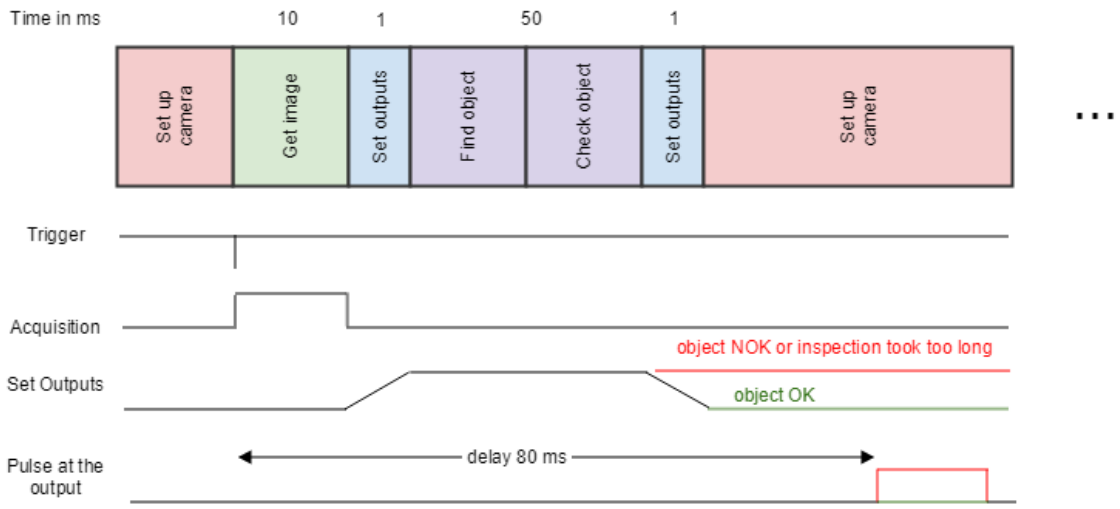
Time (acquisition → ejector)	80 ms
Pulse width for ejector	10 ms
Input (trigger)	IO 0
Output (ejector)	IO 1

Make the appropriate digital inputs and outputs settings in System Settings:



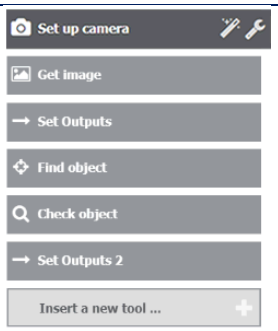
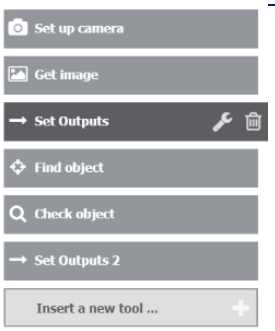
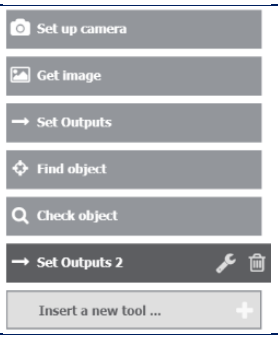
The delay ensures that a signal with a pulse width of 10 ms will be available at the output 80 ms after the trigger (right in time when the object is at the ejector). Whether the output is active or not, this signal is available externally.

Scenario 1: Ejecting NOK objects

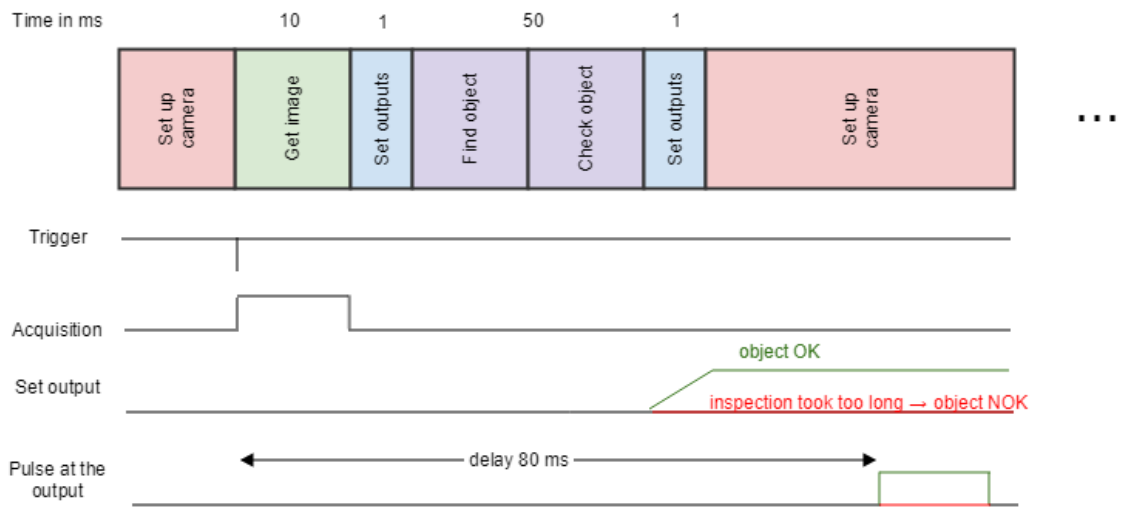


As soon as a trigger signal is at the input, the acquisition will be started. Afterwards the output is set to active. If the object is OK, the output will be set to inactive, in order that the pulse signal is ineffective after 80 ms.
If the inspection took too long or the object was NOK, the output will remain active and the object will be ejected as NOK.

The suitable settings will be as follows.

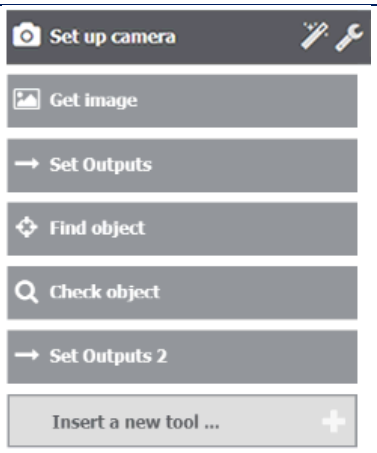
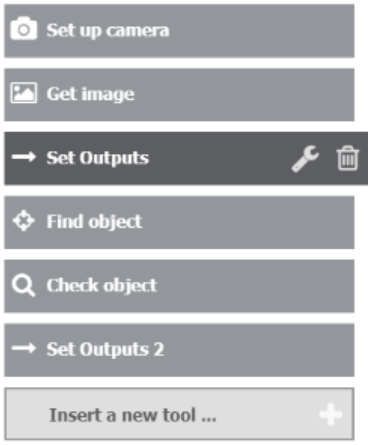
	**(0) Set up camera** 1. Set "**Trigger mode**" to "**edge**".
	**(1) Set output** 1. Set "**IO 1**" to "**true**".
	**(2) Set output 2** 1. Set "**IO 1**" to "**NOT Check_object.Tool_processing**". **If an object is NOK, "Check_object.Tool_processing = false". For this reason the output remains active and the object will be ejected.**

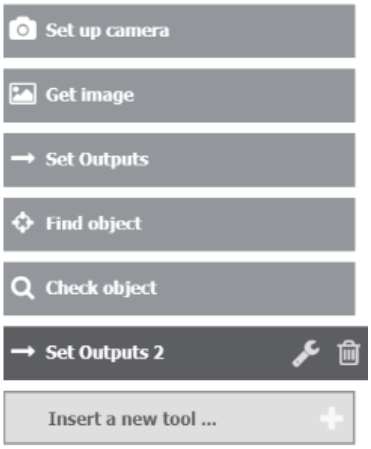
Scenario 2: Active forwarding of OK objects



As soon as a trigger signal is at the input, the acquisition will be started. Afterwards, the output will only be active, if the inspection was processed in time and the object was OK. If the inspection took too long, all objects are considered as OK.

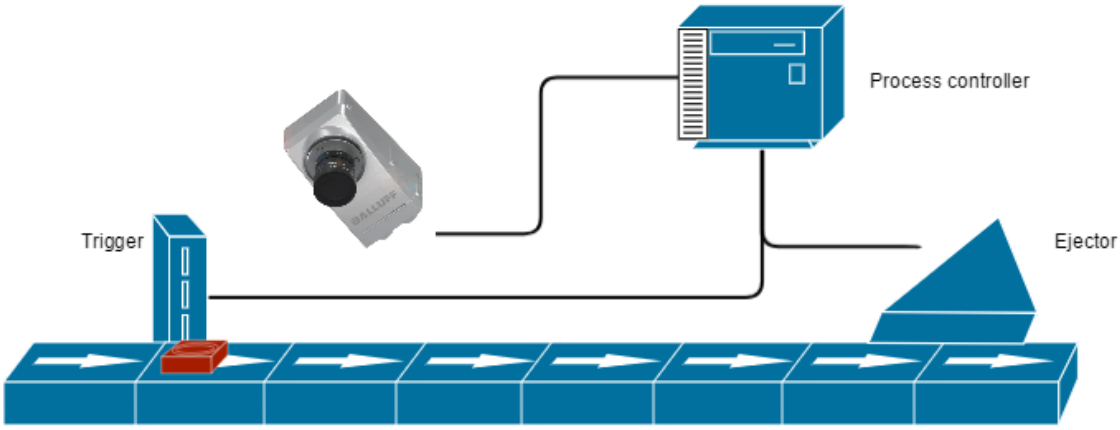
The suitable settings will be as follows.

	**(0) Set up camera** 1. Set "**Trigger mode**" to "**edge**".
	**(1) Set output** 1. Set "**IO 1**" to "**false**".

	**(2) Set output 2** 1. Set "**IO 1**" to "**Check_object.Tool_processing**".

8.3.2 Network based trigger and ejection

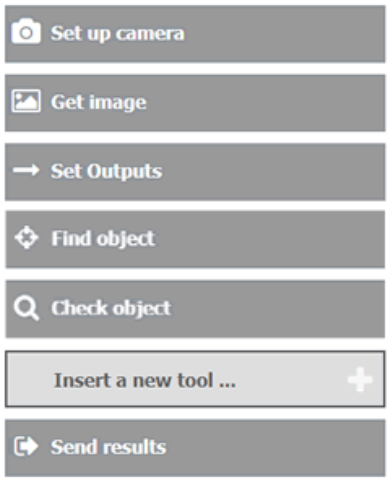
This use case shows how a connected camera gets a trigger from a control unit via UDP, and how the camera inform the control unit, if a part has to be ejected. In most cases, the installation will look like as follows:



In detail,

1. a software trigger will be trigger from a control unit to start an acquisition, afterwards
2. the camera inspects the object and finally,
3. the camera sends the inspection results to the control unit, so that the unit can control the ejector.

The complete inspection program will look like follows:



Activate UDP in System Settings:

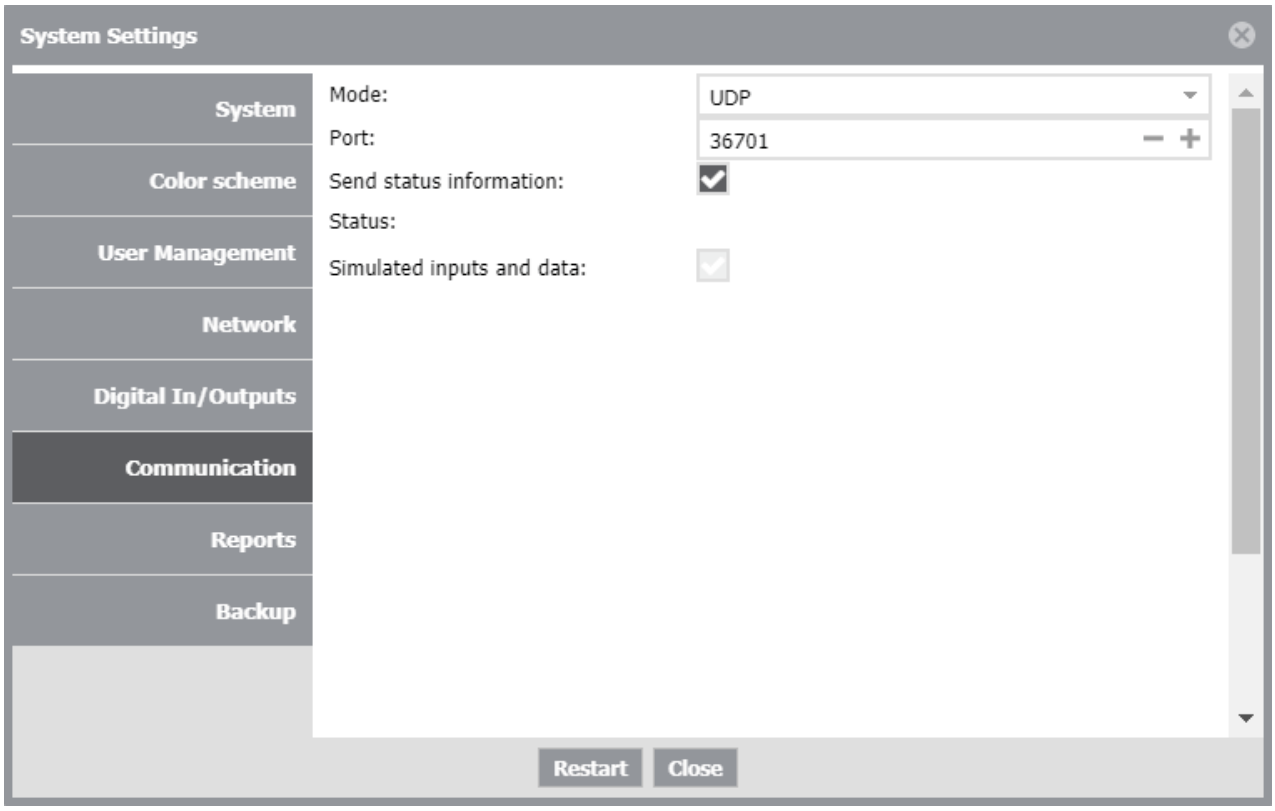
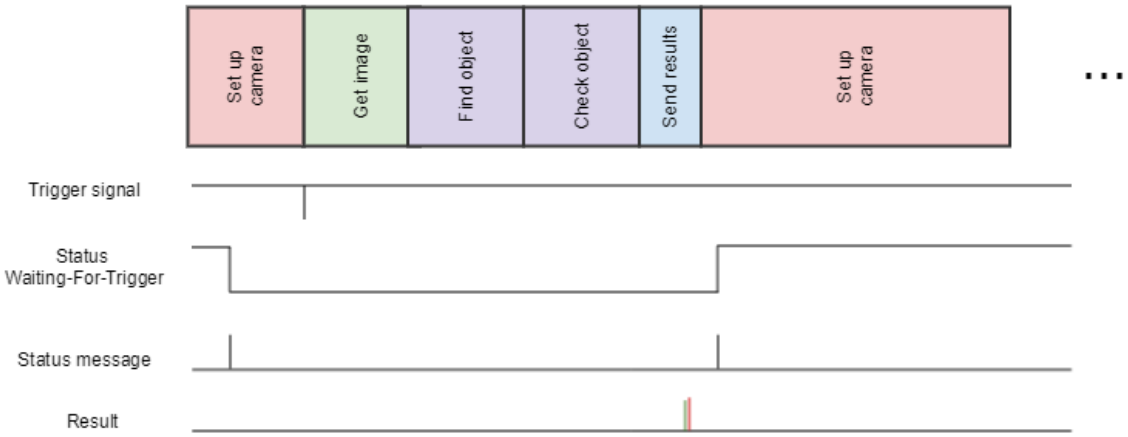


Figure 1 UDP activated

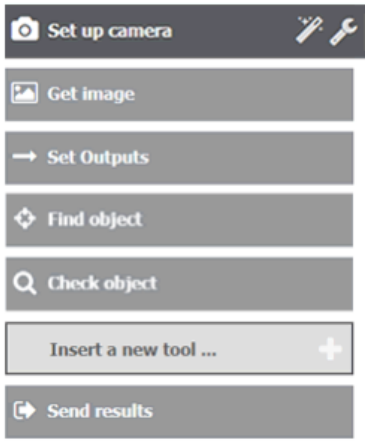
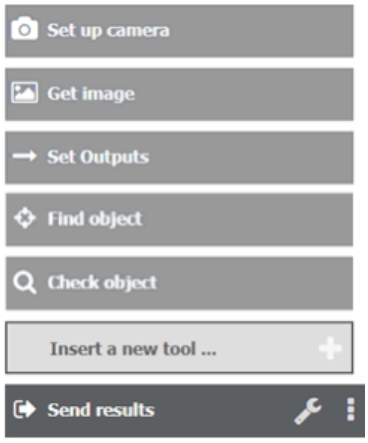
Process sequence



Controller		Camera
	←	Status (Waiting-For-Trigger)
Trigger signal	→	
	←	Status (NOT Waiting-For-Trigger)
Result	→	
	←	Status (Waiting-For-Trigger)

As soon as the control unit sends a trigger signal via UDP, an image acquisition will start. The control unit will receive a status message without a set Waiting-For-Trigger bit, i.e. the camera is busy with the inspection processing and will ignore further trigger signals. After the inspection is completed, a new image can be acquired and for this reason the control unit will receive a status message with a set Waiting-For-Trigger bit. The control unit is responsible to send a new trigger signal only then, when the camera has sent a Waiting-For-Trigger status.

The suitable settings will be as follows.

	**(0) Set up camera** 1. Set "**Trigger mode**" to "**edge**".
	**(1) Sens results** 1. Set "**bool**" as parameter type and "**Check_object.Tool_processing**" as data you want to send.

8.3.3 Using input and output configurations

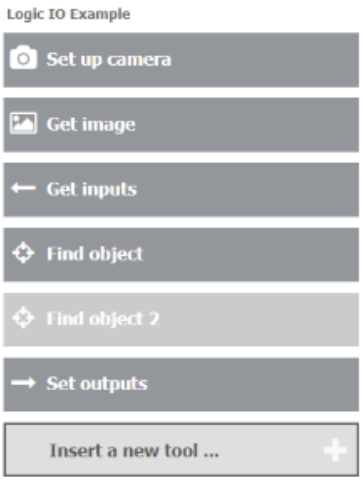
This use case assumes that an inspection program is created on a **SMART***CAMERA*, then exported to another plattform (PC) and optimized there, and finally is operated again on the **SMART***CAMERA*. This scenario can be relevant if bad part images are used for an offline training on a PC. You can also use this approach to exchange inspection programs between different plattforms.

In detail,

1. an inspection program with inputs and outputs is created on the **SMART***CAMERA*.
2. This program is imported to and optimized on a PC.
3. The optimized inspection program will be operated on the **SMART***CAMERA*.

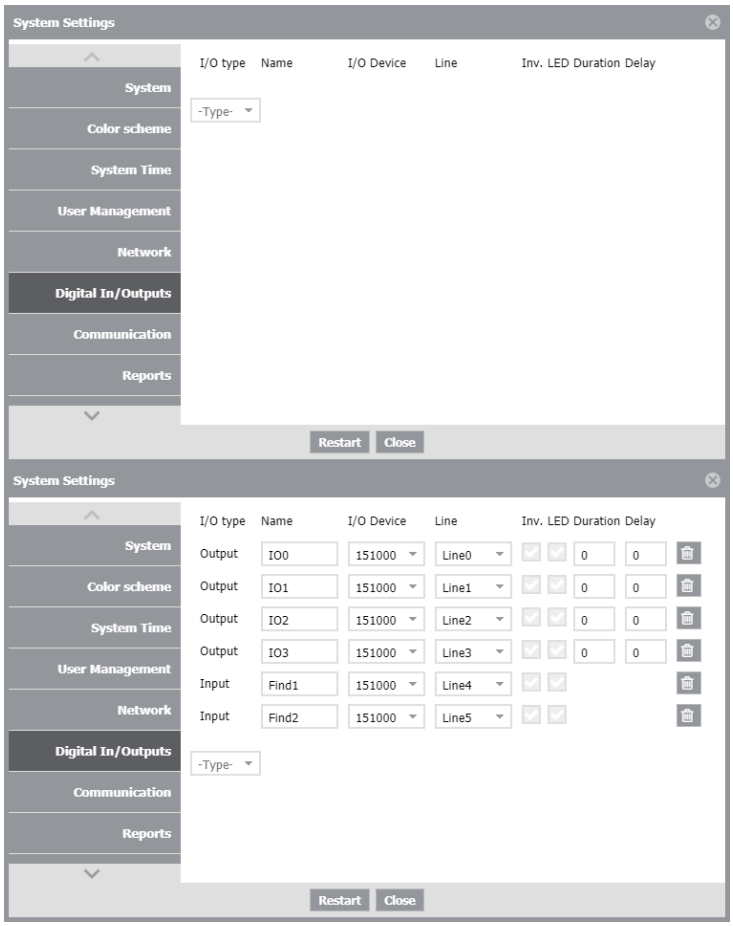
Configuration of the inspection program in the SMARTCAMERA

The inspection program consists of the following tools:



Here, two "**Find object**" tools should be turned to active or inactive according to two digital inputs ("Find1" and "Find2"). Results are send via the tool "**Set outputs**".

In the **System Settings** the I/O configurations are defined so that the camera can communicate via the digital inputs and outputs.



NOTE

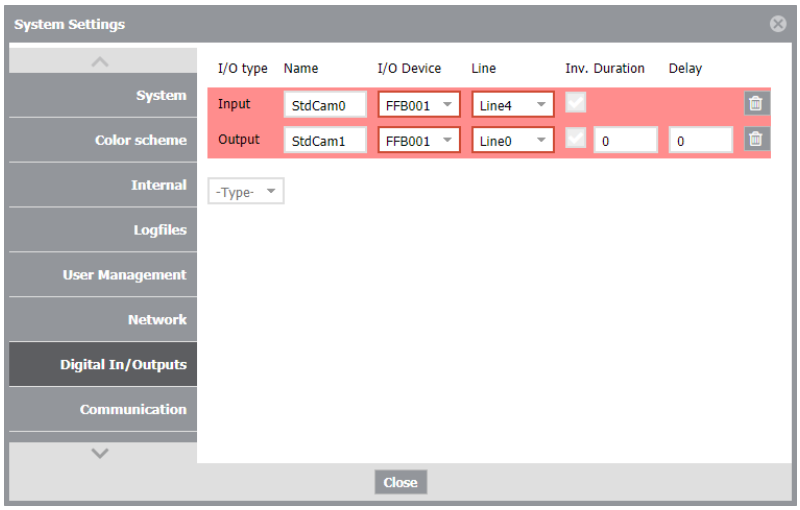
If the software was updated from a version < 2.5 to version 2.5, you will see the previously set digital inputs and outputs. To change these (e.g. to change an input to an output) or to add new ones, you have to delete the ones which are not used.

NOTE

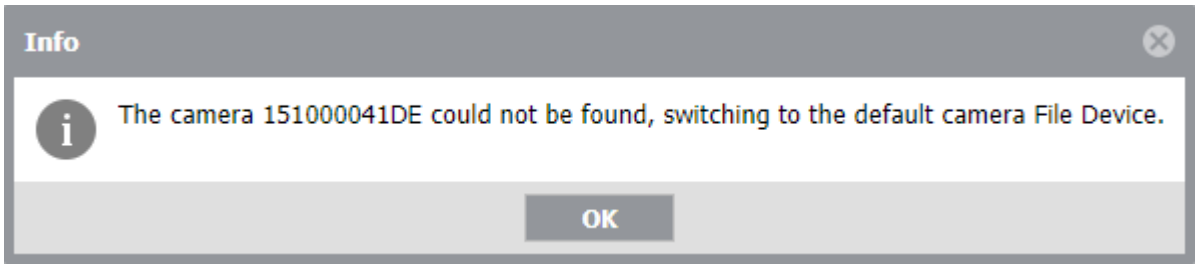
You can only use the number of I/O configurations which are provided by the used hardware (the selected camera in the tool "**Set up camera**"). Here the number of inputs and outputs is different from hardware to hardware. If you want to use more configurations as there are available by the hardware, you can only run the inspection program in the simulation mode (see below).

Importing the inspection program into the PC

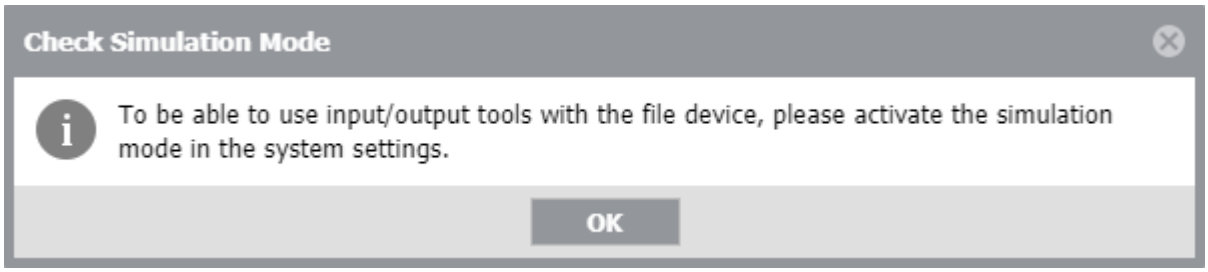
On the PC, which is used to optimize the inspection program, there are following I/O configurations available.



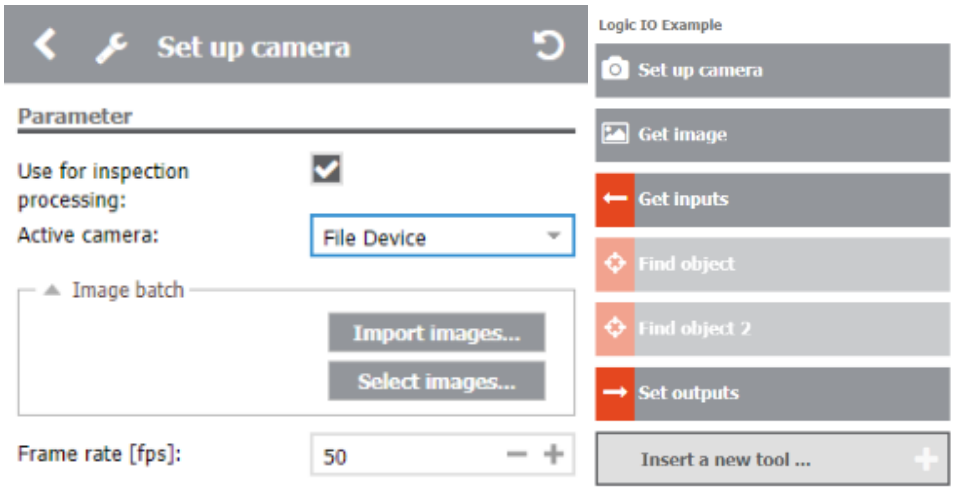
The exported inspection program is imported and opened in the PC.



Because the **SMART***CAMERA* is not available on the PC, the tool **" Set up camera**" changes to FileDevice automatically.



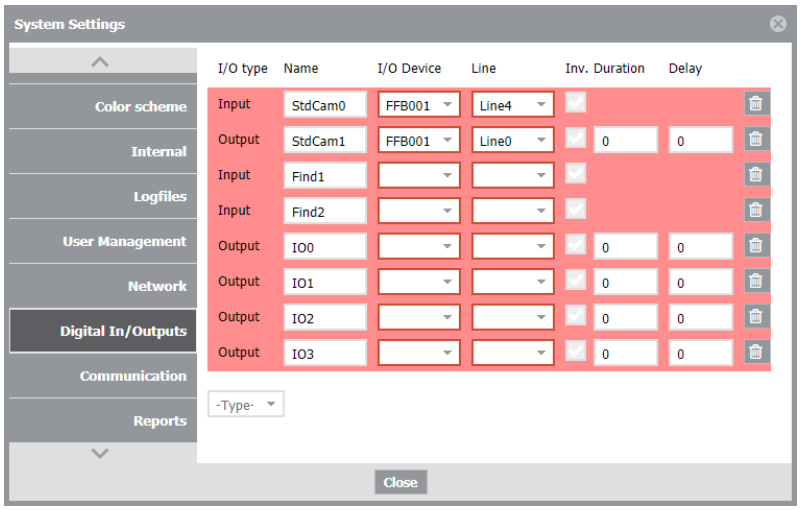
Because the FileDevice has no hardware based inputs and outputs, the used I/O configurations are invalid and the inspections program is not executable. A dialog points this out and suggests to change into the simulation mode. For a better understanding we will make this later.



In **System Settings**, the list of I/O configurations will be extended with the items of the inspection program automatically.

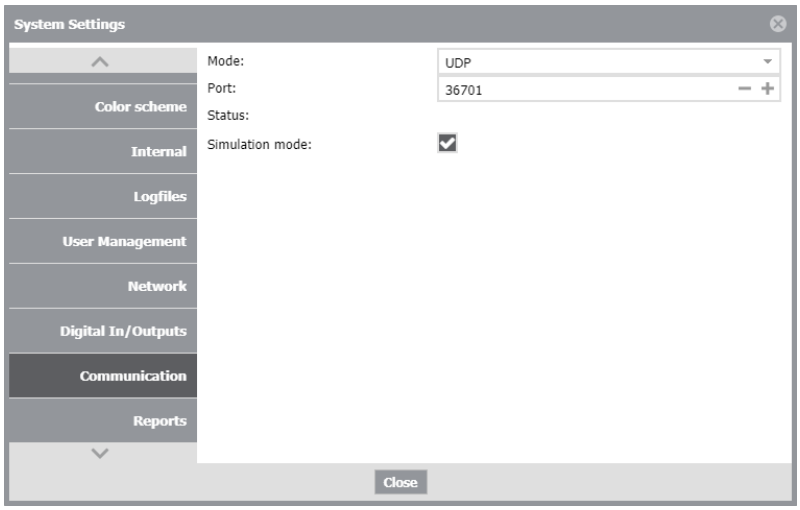
NOTE

The import process tries to take over existing I/O configurations. For this, name and I/O type have to match.

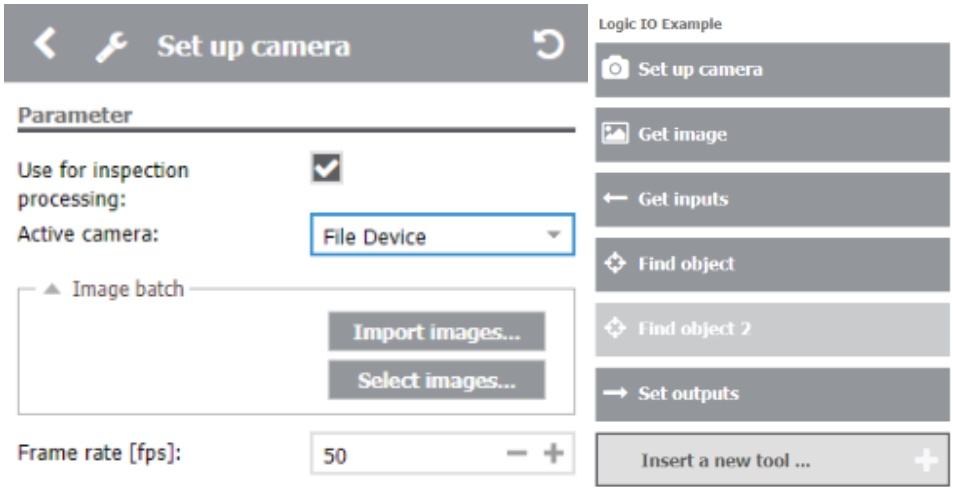


The imported I/O configurations are not connected to a hardware line. For this reason there are invalid. The configurations of the PC are invalid too, because the tool "**Set up camera**" has changed to FileDevice..

In **System Settings** in the section **Communication**, you have to change to the simulation mode so that the inspection program runs again. In this mode, the digital inputs and outputs of the tools "**Get inputs**" and "**Set outputs**" can be set by the user manually and there is no attention to hardware connection of the inputs and outputs.



The inspection program is now executable and can be optimized, for example, by training of new good parts and bad parts.



After training, you can export and import the inspection program to the **SMART***CAMERA*.

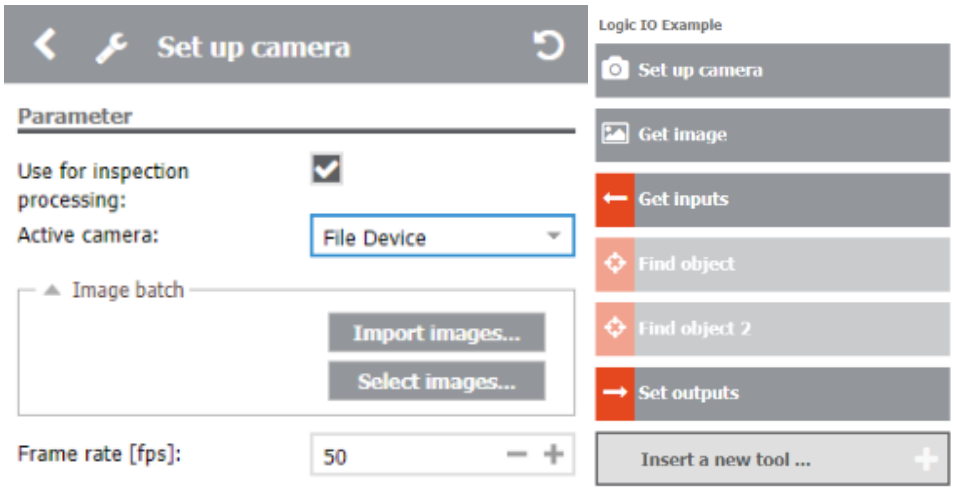
NOTE

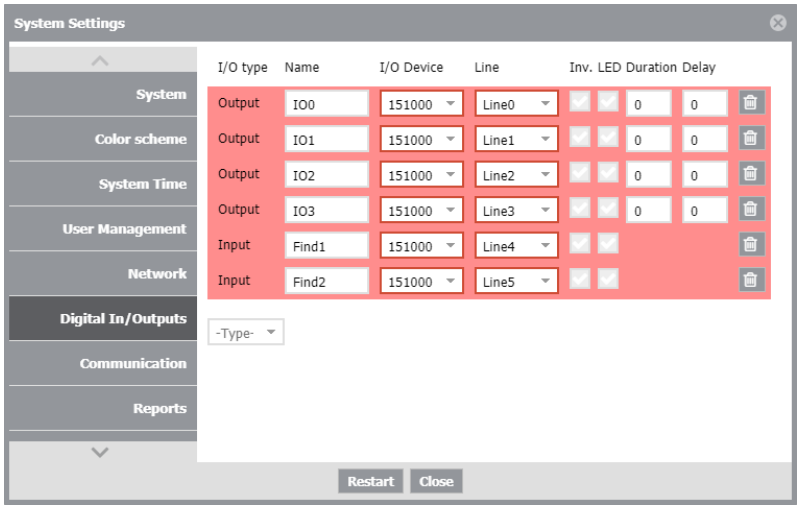
You have to delete not used settings of digital inputs and outputs on the PC, so that they are not exported.

Before importing an inspection program, please delete not used or better all digital input and output settings.

Operating of an optimized program on the SMARTCAMERA

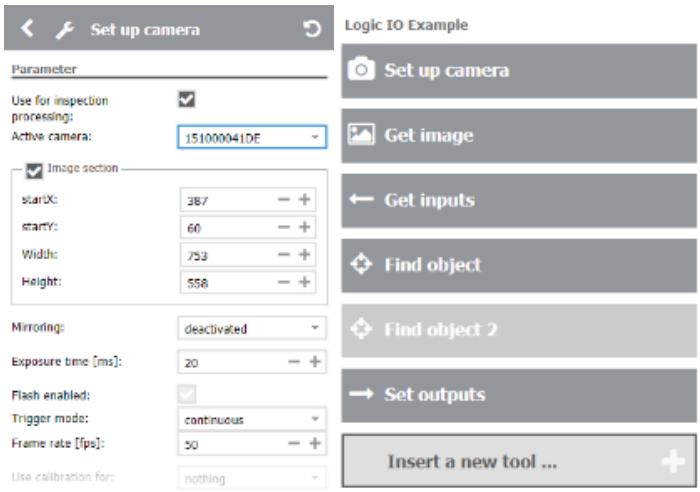
After the import of an optimized inspection program on the **SMART***CAMERA*, FileDevice is still set and for this reason the settings of the digital inputs and outputs are invalid.

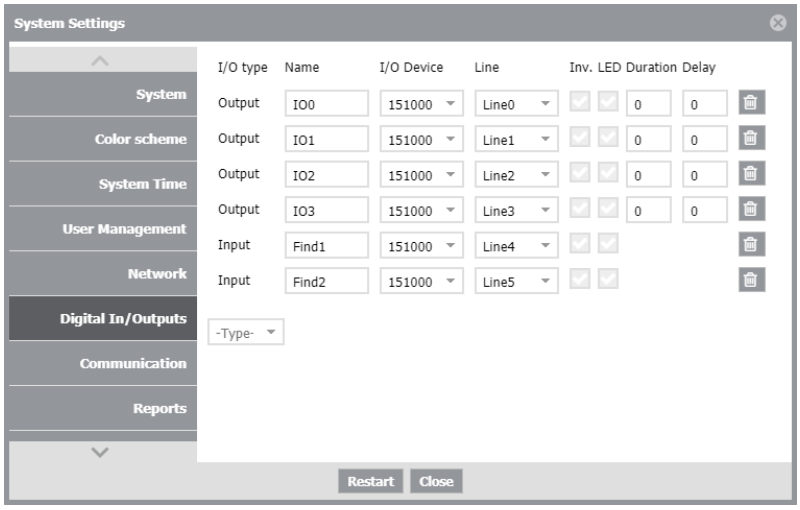




Here, you also get the dialog to change to the simulation mode. In order to use the **SMART***CAMERA*, you have to select the real **SMART***CAMERA* in the tool "**Set up camera**".

In "**System Settings**", the settings of the digital inputs and outputs will become valid again and so the tools in the inspection program.


Set up camera
Parameter
Use for inspection processing:
Active camera:
151000041DE
Image section
startX:
387
startY:
60
Width:
753
Height:
558
Mirroring:
deactivated
Exposure time [ms]:
20
Flash enabled:
Trigger mode:
continuous
Frame rate [fps]:
50
Use calibration for:
nothing
Logic IO Example
Set up camera
Get image
Get inputs
Find object
Find object 2
Set outputs
Insert a new tool ...


System Settings
System
Color scheme
System Time
User Management
Network
Digital In/Outputs
Communication
Reports
I/O type
Name
I/O Device
Line
Inv.
LED
Duration
Delay
Output
IO0
151000
Line0
0
0
Output
IO1
151000
Line1
0
0
Output
IO2
151000
Line2
0
0
Output
IO3
151000
Line3
0
0
Input
Find1
151000
Line4
Input
Find2
151000
Line5
-Type-
Restart
Close

9 Appendix

9.1 Troubleshooting Table

Error	Cause	Action
Accessing the shared folders is not possible.	Your Windows 10 installation does not allow a guest access to the shared folders (BVS Cockpit releases <= 2.3.0 and >= 2.2.2); sometimes also with error code **0x80070035**	Update BVS Cockpit to a version > 2.3.0, which allows accessing the shared folders with following credentials: • Login: *expert* • Password: *expert* If Windows does not prompt for the login and password please use this command in a terminal window, replacing the host name with the correct value: `cmdkey.exe /add:sc-16031893de /U:expert /P:expert`

innovating automation

Headquarters

Germany
Balluff GmbH
Schurwaldstrasse 9
73765 Neuhausen a.d.F.
Phone +49 7158 173-0
Fax +49 7158 5010
balluff@balluff.de

DACH Service Center

Germany
Balluff GmbH
Schurwaldstrasse 9
73765 Neuhausen a.d.F.
Phone +49 7158 173-370
service.de@balluff.de

Southern Europe Service Center

Italy
Balluff Automation S.R.L.
Corso Cuneo 15
10078 Venaria Reale (Torino)
Phone +39 0113150711
service.it@balluff.it

Eastern Europe Service Center

Poland
Balluff Sp. z o.o.
Ul. Graniczna 21A
54-516 Wrocław
Phone +48 71 382 09 02
service.pl@balluff.pl

Americas Service Center

USA
Balluff Inc.
8125 Holton Drive
Florence, KY 41042
Toll-free +1 800 543 8390
Fax +1 859 727 4823
service.us@balluff.com

Asia Pacific Service Center

Greater China
Balluff Automation (Shanghai) Co., Ltd.
No. 800 Chengshan Rd, 8F, Building A,
Yunding International Commercial Plaza
200125, Pudong, Shanghai
Phone +86 400 820 0016
Fax +86 400 920 2622
service.cn@balluff.com.cn

No. 928015-726 EN · 13.131491 · G21; subject to modification. Replaces D21.